



ВАРНЕНСКИ СВОБОДЕН УНИВЕРСИТЕТ

Черноризец Храбър

Архитектурен факултет

Катедра „Архитектура и урбанистика“

**Организационни модели за създаване на сгради в България и Бавария
и формиране на професионалните задачи на архитектите**

Дисертация за присъждане на
образователна и научна степен “доктор“

Докторска програма

“Архитектура на сгради, конструкции, съоръжения и детайли“

Том 2

Автор на дисертацията:	арх., дипл. инж. Стоян Георгиев Тодоров
Научен ръководител:	доц., д-р, арх. Жечка Илиева
Професионално направление:	„5.7. Архитектура, строителство и геодезия“
Научна специалност:	„Архитектура на сгради, съоръжения и детайли“
Рецензенти:	чл.-кор. на БАН, проф., д. а. н. арх. Атанас Ковачев доц., д-р, арх. Пламен Петров

юни 2026 г., гр. Варна

© Авторско право

Този труд, както и всички съдържащи се в него отделни материали и изображения, са защитени с авторско право. Всяко използване, което не е изрично разрешено от законодателството за авторското право, изисква предварителното съгласие на автора.

Това се отнася по-специално за възпроизвеждане, обработка, преводи, микрофилмиране, извличане на информация чрез бази данни и изкуствен интелект, както и за съхраняване и обработка в електронни системи.

Всички права, включително тези за частично препечатване, фотомеханично възпроизвеждане (включително микрокопиране), както и за използване чрез бази данни и изкуствен интелект или подобни системи, са запазени.

Съдържание

© Авторско право.....	2
ПРИЛОЖЕНИЯ.....	4
Приложение 1. Участници в инвестиционния процес в.....	5
строителството на сгради в България и техните функции.	5
Приложение 2. Участници в пазара на архитектурни и инженерни услуги в.....	5
строителството в България (към 31.12.2023 г.).	5
Приложение 4. Участници в инвестиционния процес в.....	5
строителството на сгради в Бавария и техните функции.	5
Приложение 3. Съпоставяне на Описанието на основните проектантски услуги, включени в отделните фази (чл. 19 ал. 2 от Методика за определяне на размера на възнагражденията за предоставяне на проектантски услуги в устройственото планиране и в инвестиционното проектиране) и Приложение 10 към Наредба за хонорарите за услуги на архитекти и инженери, издадена от правителството на Федерална република Германия.....	7
Приложение 5. Решение №1 на КАБ от 12.04.2024 г.	28
Приложение 6. Жалба до АССГ.	33
Приложение 7. Становище на ДНСК до АССГ.....	42
Приложение 8. Решение на ВАС по дело срещу ДНСК.....	47
Приложение 9. Фактори, влияещи върху вземането на решения от възложител в България относно оптималния модел на реализация на строително начинание.	52
Приложение 10. Фактори, влияещи върху вземането на решения от възложител в Бавария относно оптималния модел на реализация на строително начинание.	52
Приложение 11. Логическа верига на работния процес в България. Приложение 12. Логическа верига на работния процес в Бавария. Приложение 13. Работен алгоритъм в България.	52
Приложение 14. Работен алгоритъм в Бавария.	52
Приложение 15. Псевдокод на уеб-базиран софтуер, който ще води консултирането до едно оптимално организационно решение.	54
Приложение 16. Документация на β-версия на.....	63
уеб-базирано приложение ARCHModels.	63

ПРИЛОЖЕНИЯ

- Приложение 1.** Участници в инвестиционния процес в строителството на сгради в България и техните функции.
- Приложение 2.** Участници в пазара на архитектурни и инженерни услуги в строителството в България (към 31.12.2023 г.).
- Приложение 3.** Съпоставяне на Описанието на основните проектантски услуги, включени в отделните фази (чл. 19 ал. 2 от Методика за определяне на размера на възнагражденията за предоставяне на проектантски услуги в устройственото планиране и в инвестиционното проектиране) и Приложение 10 към Наредба за хонорарите за услуги на архитекти и инженери, издадена от правителството на Федерална република Германия.
- Приложение 4.** Участници в инвестиционния процес в строителството на сгради в Бавария и техните функции.
- Приложение 5.** Решение №1 на КАБ от 12.04.2024 г.
- Приложение 6.** Жалба до АССГ.
- Приложение 7.** Становище на ДНСК до АССГ.
- Приложение 8.** Решение на ВАС по дело срещу ДНСК.
- Приложение 9.** Фактори, влияещи върху вземането на решения от възложител в България относно оптималния модел на реализация на строително начинание.
- Приложение 10.** Фактори, влияещи върху вземането на решения от възложител в Бавария относно оптималния модел на реализация на строително начинание.
- Приложение 11.** Логическа верига на работния процес в България.
- Приложение 12.** Логическа верига на работния процес в Бавария.
- Приложение 13.** Работен алгоритъм в България.
- Приложение 14.** Работен алгоритъм в Бавария.
- Приложение 15.** Псевдокод на уеб-базиран софтуер, който ще води консултирането до едно оптимално организационно решение.
- Приложение 16.** Документация на β-версия на уеб-базирано приложение ARCHModels.

Приложение 1. Участници в инвестиционния процес в
строителството на сгради в България и техните функции.

Приложение 2. Участници в пазара на архитектурни и инженерни услуги в
строителството в България (към 31.12.2023 г.).

Приложение 4. Участници в инвестиционния процес в
строителството на сгради в Бавария и техните функции.

към

Дисертация за присъждане на
образователна и научна степен “доктор“

на тема

**Организационни модели за създаване на сгради в България и Бавария
и формиране на професионалните задачи на архитектите**

Приложение 3. Съпоставяне на Описанието на основните проектантски услуги, включени в отделните фази (чл. 19 ал. 2 от Методика за определяне на размера на възнагражденията за предоставяне на проектантски услуги в устройственото планиране и в инвестиционното проектиране) и

Приложение 10 към Наредба за хонорарите за услуги на архитекти и инженери, издадена от правителството на Федерална република Германия.

към

Дисертация за присъждане на
образователна и научна степен “доктор“

на тема

**Организационни модели за създаване на сгради в България и Бавария
и формиране на професионалните задачи на архитектите**

В синьо са оцветени, допълненията/измененията, които са предложени от Федералната камера на архитектите в Германия¹

МЕТОДИКА за определяне на размера на възнагражденията за предоставяне на проектантски услуги в устройственото планиране и в инвестиционното проектиране² Издадена от председателя на Камарата на архитектите в България, обн., ДВ, бр. 41 от 22.05.2007 г., в сила от 23.08.2007 г., изм. и доп., бр. 37 от 7.05.2019 г., в сила от 7.05.2019 г. Чл. 19 (2) Описанието на основните проектантски услуги, включени в отделните фази, е:	Наредба за хонорарите за услуги на архитекти и инженери (Наредба за хонорарите на архитекти и инженери) – в редакция от 2021 г. / <i>Verordnung über die Honorare für Architekten- und Ingenieurleistungen (Honorarordnung für Architekten und Ingenieure – HOAI) – in der Fassung von 2021</i> Приложение 10 (към § 34 ал 4, § 35 ал. 7) / Anlage 10 (zu § 34 Absatz 4, § 35 Absatz 7) Основни дейности в спектъра от услуги относно сгради и вътрешни пространства, Допълнителни дейности, Списъци на видове строителни обекти / Grundleistungen im Leistungsbild Gebäude und Innenräume, Besondere Leistungen, Objektlisten 10.1 Спектър от услуги относно сгради и вътрешни пространства / Leistungsbild Gebäude und Innenräume
Фаза 1 – Предварителни (прединвестиционни) проучвания Фазата в зависимост от обекта може да включва дейности и служи за изясняване на: 1. изясняване на инвестиционното строително намерение, изясняване на възлагането, посещение и оглед на площадката, обсъждане на необходимите процедури и проучвателни дейности, определяне на останалите специалности; 2. проучвания за определяне на подходящо местоположение на обекта и на условията за застрояване съобразно предвиденията на устройствените планове;	Фаза 1. Набавяне на изходни данни / LPh 1 Grundlagenermittlung. <i>Основни дейности / Grundleistungen</i> а) Изясняване на задачата въз основа на изискванията или прогнозирането на нуждите на възложителя / Klären der Aufgabenstellung auf Grundlage der Vorgaben oder der Bedarfsplanung des Auftraggebers б) Посещение на място / Ortsbesichtigung в) Консултиране относно цялостните нужди от дейности и проучвания вкл. изясняване на метод на проектиране и на изискванията на

¹ *BIM für Architekten Leistungsbild Vertrag Vergütung*: <https://bak.de/wp-content/uploads/2021/06/bim-bak-broschuere-web.pdf> [посетен на 13 юни 2024 г.]

² Текстът на Методиката е възпроизведен с оригиналната пунктуация.

3. инженерни проучвания, в т.ч. геоложки, геотехнически, хидрогеоложки, хидроложки и др., като съдържанието им се определя в зависимост от местоположението, вида, характера и спецификата на обекта; 4. проучвания и документиране за нуждите на инвестиционното проектиране на обекти – недвижими културни ценности; 5. геодезически проучвания; 6. технологични проучвания за избор на оферта, предлагаща оптимална технология – за обекти, при които технологичното решение е определящо; 7. технико-икономическа обосновка за определяне на икономическата целесъобразност и ефективност на инвестиционното строително намерение, в т.ч. за избор на подходящ негов вариант; 8. обосновка на социалната целесъобразност и ефективност на инвестиционната инициатива, в т.ч. откриване на работни места и осигуряване на обществени услуги; 9. проучвания за потенциала на териториите с особена и превантивна устройствена защита във връзка с режимите за опазване на териториите за природозащита и на обектите на културно-историческото наследство; 10. вариантни проучвания за рехабилитация на съществуващи и за изграждане на нови енергийни мощности; 11. специфични проучвания при реконструкция, пристрояване и надстрояване на съществуващи сгради, в т.ч. архитектурно и конструктивно заснемане на сградите и изследване на техническата инфраструктура; 12. микросейсмично изследване – за енергийни, хидротехнически, транспортни и други специфични обекти; 13. други специфични проучвания, налагащи се от вида и спецификата на отделните строежи, като климатични, водностановни, почвено-мелиоративни, аграрно-икономически и земеустройствени проучвания – при проекти на хидромелиоративни системи, проекти за противосвлачищни и брегоукрепителни съоръжения и др.;	възложителя относно информационния ресурс / Бератен zum gesamten Leistungs- und Untersuchungsbedarf einschließlich Klären der Planungsmethode und der Auftraggeber-Informations-Anforderungen d) Съдействие при изготвянето на план за изпълнението на BIM / Mitwirken beim Erarbeiten eines BIM-Abwicklungsplans e) Определяне на подкрепата, необходима за вземане на решения относно избор на други специализирани участници в проектирането / Formulieren der Entscheidungshilfen für die Auswahl anderer an der Planung fachlich Beteiligter f) Обобщаване, обяснение и документиране на резултатите / Zusammenfassen, Erläutern und Dokumentieren der Ergebnisse <i>Допълнителни дейности / Besondere Leistungen</i> – Прогнозиране на нуждите / Bedarfplanung – Установяване на нуждите / Bedarfsermittlung – Изготвяне на функционална схема / Aufstellen eines Funktionsprogramms – Изготвяне на пространствена схема / Aufstellen eines Raumprogramms – Анализ на местоположението / Standortanalyse – Съдействие при избора, закупуването и прехвърлянето на поземлен обект или строителен обект / Mitwirken bei Grundstücks- und Objektauswahl, -beschaffung und -übertragung – Осигуряване на документи, които имат значение за начинанието / Beschaffen von Unterlagen, die für das Vorhaben erheblich sind – Архитектурно заснемане / Bestandsaufnahme – Техническо изследване на средата / technische Substanzerkundung – Оперативно планиране / Betriebsplanung – Проверка на екологичната значимост / Prüfen der Umweltherheblichkeit – Проверка на екологичната съвместимост / Prüfen der Umweltverträglichkeit – Проучване за осъществимост / Machbarkeitsstudie – Икономическо изследване / Wirtschaftlichkeitsuntersuchung
--	--

14. прогнозен график за продължителност на строителството от откриване на строителна площадка и определяне на строителна линия и ниво до предаване на обекта от строителя на възложителя с акт образец 15; 15. определяне на прогнозна стойност на строителното намерение.	– Планиране на структурата на проекта / Projektstrukturplanung – Съставяне на изискванията на базата на системи за сертифициране / Zusammenstellen der Anforderungen aus Zertifizierungssystemen – Подкрепа при осъществяване на процедури, участие при възлагане на проектирането и на експертизи / Verfahrensbetreuung, Mitwirken bei der Vergabe von Planungs- und Gutachterleistungen – Изготвяне на дигитална платформа за сътрудничество (среда за общи данни) / Bereitstellen einer digitalen Kollaborationsplattform (Common Data Environment, CDE) – Управление на BIM / BIM-Management – Дигитално документиране на съществуващи сгради или на информации относно парцели / Digitale Erfassung von Bestandsgebäuden oder Grundstücksinformationen – Проверка на квалификацията относно BIM на другите специализирани участници в проектирането / Prüfung der BIM-Qualifikation von anderen an der Planung fachlich Beteiligten – Изясняване на изискванията относно обмена на данни с институции / Abklären der Anforderungen an den Datenaustausch mit Behörden
Фаза 2 – Обемно-устройствено проучване Фазата може да включва предварителни дейности, като: Съгласуване на предстоящата дейност с останалите специалисти на вижданията за крайния резултат, изисквания и конфликтни моменти. Изясняване и обясняване на взаимни връзки и зависимости (градоустройствени, обемно-пространствени, функционални икономически, енергийнозависими, социални и т.н.) Предварителни разговори за възможностите за одобряване на проекта, Предварителна оценка на стойността на обекта в сравнение със зададените финансови параметри Като основната дейност на архитект-проектантът: 1. определя очертанията на бъдещите обекти за застрояване, така че те да отговарят на зададените с действащия подробен устройствен план	Фаза 2. Предпроектно проучване и планиране / L.Ph 2 Vorplanung (Projekt- und Planungsvorbereitung) <i>Основни дейности / Grundleistungen</i> a) Анализиране на предпоставките, координиране на дейностите с проектантите по техническите специалности съобразно избрания метод за проектиране и съдействие при развитието на плана за изпълнението на BIM / Analysieren der Grundlagen, Abstimmen der Leistungen mit den fachlich an der Planung Beteiligten gemäß der gewählten Planungsmethode und Mitwirken beim Fortschreiben des BIM-Abwicklungsplans b) Съгласуване на набелязаните цели, посочване на противоречия между набелязани цели / Abstimmen der Zielvorstellungen, Hinweisen auf Zielkonflikte

и/или на нормативно допустимите мерки, разстояния и плътност на застрояване; 2. определя обема на застрояване в рамките на зададените с действащия подробен устройствен план и/или в рамките на нормативно допустимите височина – етажност, интензивност на застрояване; 3. представя обобщено функционално-пространствено решение за отделните етажи на сградата (сградите); 4. представя обобщено обменно-фасадно решение на сградата (сградите); 5. определя обобщено оформянето на околното пространство на сградата (сградите) в поземления имот или в групата поземлени имоти; 6. дава обобщено решение на инженерните мрежи в поземления имот (имоти), когато това има съществено значение за функционалното и архитектурно-обемното решение; 7. определя задължителните изисквания и параметри за опазването на недвижими културни ценности и тяхната среда, както и за интегриране на новото застрояване към тях; 8. обемно-устройственото проучване се представя със следните минимално необходими чертежи: 9. ситуационно решение върху комбинирана скица от кадастралната карта (кадастрален план) и от действащия подробен устройствен план или върху извадка от действащия ПУП в мащаб (М) 1:1000 или М 1:500; 10. схематични разпределения на етажите с обобщени групи помещения в М 1:500 или М 1:200; 11. схематични фасади-силуети с обобщено оформление на обемите в М 1:500 или М 1:200. Към обемно-устройственото проучване могат да се представят допълнително по преценка на проектанта или при договаряне с възложителя и следните чертежи: схематични разрези, схема на разположението на мрежите на техническата	с) Изготвяне на предпроектното проучване и планиране; разглеждане, представяне и оценяване на варианти по едни и същи изисквания, изработване на чертежи или на дигитални модели в мащаб според вида и големината на обекта / Erarbeiten der Vorplanung, Untersuchen, Darstellen und Bewerten von Varianten nach gleichen Anforderungen, Zeichnungen oder digitalen Modellen im Maßstab nach Art und Größe des Objekts д) Изясняване и обясняване на основните взаимоотношения, изисквания и условия (например: градоустройствени, проектантски, функционални, технически, икономически, екологични, строително-физични, енергийни, социални, публично-правни) / Klären und Erläutern der wesentlichen Zusammenhänge, Vorgaben und Bedingungen (zum Beispiel städtebauliche, gestalterische, funktionale, technische, wirtschaftliche, ökologische, bauphysikalische, energiewirtschaftliche, soziale, öffentlich-rechtliche) е) Предоставяне на резултатите от горепосочените работи като основа за дейностите на другите проектанти по техническите специалности, както и координиране и интегриране на техните дейности при изготвяне и използване на собствения дигитален модел и на дигиталните модели на другите специализирани участници в проектирането / Bereitstellen der Arbeitsergebnisse als Grundlage für die anderen an der Planung fachlich Beteiligten sowie Koordination und Integration von deren Leistungen unter Erstellen und Verwenden des eigenen digitalen Modells und der digitalen Modelle der anderen fachlich an der Planung Beteiligten ф) Предварителни преговори относно допустимостта на строителното начинание / Vorverhandlungen über die Genehmigungsfähigkeit г) Приблизителна преценка за разходите по DIN 276 въз основа на количествата, изведени от дигиталните модели; сравнение с финансовите рамкови условия / Kostenschätzung nach DIN 276 auf Basis der aus den digitalen Modellen abgeleiteten Mengen, Vergleich mit den finanziellen Rahmenbedingungen
---	---

инфраструктура, аксонометрични изгледи, перспективни изгледи, фотомонтажи, компютърна анимация, макет и др. Обемно-устройственото проучване се придружава от обяснителна записка, която пояснява предложените решения, тяхната целесъобразност и постигнатите технико-икономически показатели. Обемно-устройственото проучване може да се договаря за изработване във варианти. Фазата може да послужи за изготвяне на задание към останалите специалности, координирането и интегрирането им, изработване на план-график със съществените дейности, свързани с проектирането и строителството.	h) Изработване на график с основните операции в процеса на проектиране и строителство / Erstellen eines Terminplans mit den wesentlichen Vorgängen des Planungs- und Bauablaufs i) Обобщаване, обяснение и документиране на резултатите / Zusammenfassen, Erläutern und Dokumentieren der Ergebnisse <i>Допълнителни дейности / Besondere Leistungen</i> – Създаване на каталог за планиране и изпълнение на належащите цели / Aufstellen eines Katalogs für die Planung und Abwicklung der Programmziele – Проучване на алтернативни решения според различни изисквания, включително оценка на разходите / Untersuchen alternativer Lösungsansätze nach verschiedenen Anforderungen einschließlich Kostenbewertung – Ставяне на изискванията на договорената система за сертифициране / Beachten der Anforderungen des vereinbarten Zertifizierungssystems – Прилагане на системата за сертифициране / Durchführen des Zertifizierungssystems – Допълване на документацията на предприетното проучване и на планирането поради специални изисквания / Ergänzen der Vorplanungsunterlagen auf Grund besonderer Anforderungen – Съставяне на финансов план / Aufstellen eines Finanzierungsplanes – Съдействие при набавянето на заеми и субсидии / Mitwirken bei der Kredit- und Fördermittelbeschaffung – Провеждане на икономически проучвания / Durchführen von Wirtschaftlichkeitsuntersuchungen – Извършване на предварително запитване (запитване относно допустимостта на строителното начинание) / Durchführen der Voranfrage (Bauanfrage)
---	---

	– Изготвяне на специални помощни средства за представяне, които не са необходими за уточненията в процеса на създаване на предварителния идеен проект, например: / Anfertigen von besonderen Präsentationshilfen, die für die Klärung im Vorentwurfsprozess nicht notwendig sind, zum Beispiel: – Презентационни модели / Präsentationsmodelle – Изображения в перспектива / Perspektivische Darstellungen – Движещи се изображения / Animation // Bewegte Darstellung / Animation – Колажи от цветове и материали / Farb- und Materialcollagen – Дигитален модел на терена / digitales Geländemodell – Обработка на 3D или 4D модели на сгради (строително информационно моделиране – BIM) / 3-D oder 4-D Gebäudemodellbearbeitung (Building Information Modelling BIM) – Изготвяне на задълбочена приблизителна преценка за разходите въз основа на позициите относно отделните строителни специалности / Aufstellen einer vertieften Kostenschätzung nach Positionen einzelner Gewerke – Актуализиране на структурния план на проекта (Органиграма) / Fortschreiben des Projektstrukturplanes – Изготвяне на описания на помещения в сграда / Aufstellen von Raumbüchern – Разработване и изготвяне на специални доклади за съответствие на един проект със строителните норми относно превантивна и организационна противопожарна защита при строежи от специален вид и предназначение, при съществуващи сгради или при случаи на отклонения от нормативната уредба на строителството / Erarbeiten und Erstellen von besonderen bauordnungsrechtlichen Nachweisen für den vorbeugenden und organisatorischen Brandschutz bei baulichen Anlagen besonderer Art und Nutzung, Bestandsbauten oder im Falle von Abweichungen von der Bauordnung – Изготвяне на описания на помещения в сграда, базирани на дигитални модели / Aufstellen von modellbasierten Raumbüchern
--	---

	– Повишена степен на детайлност на дигиталния модел / Erhöhter Detaillierungsgrad des digitalen Modells – Подготовка на дигиталните модели на други специализирани участници в проектирането за координация и интеграция / Aufbereiten von digitalen Modellen anderer an der Planung fachlich Beteiligter zur Koordination und Integration – Изготвяне на дигитален модел съобразно особени изисквания / Erstellen eines digitalen Modells nach besonderen Anforderungen – Повишена цикличност на изготвянето на модели / Erhöhter Turnus der Modellbereitstellung – Изготвяне на в голяма степен интегрирани и лишени от колизии модели към междинни дати / Erstellen von weitgehend integrierten, kollisionsfreien Modellen zu Zwischenzeitpunkten – Проучване на алтернативни възможности за решения съобразно различни изисквания чрез използване на множество дигитални модели / Untersuchen von alternativen Lösungsmöglichkeiten nach verschiedenen Anforderungen unter Verwendung mehrerer digitaler Modelle
Фаза 3 – Идеен проект Фазата идеен проект се изготвя с отразяване на съществените изисквания, условия и предпоставки на основата на предприемаческите проучвания. Идеиният проект служи като предпоставка за следващите фази и изяснява изискванията се нормативно действия за одобряване на проекта с включване на проектни работи от останалите специалности. Част архитектурна представя решения за: – функционално-пространствената композиция на обекта (сградата) в съответствие с предвижданията на действащия ПУП или визата за проектиране; – архитектурно-художествения образ на обекта (сградата); – вписването на обекта в околната архитектурна и ландшафтна среда;	Фаза 3. Идеен проект (Архитектурно проектиране на сградата в съгласие с идеите на други специализирани участници в проектирането и планиране на сътрудничеството между архитект, конструктор и другите специализирани участници в проектирането) / LPh 3 Entwurfsplanung (System- und Integrationsplanung) <i>Основни дейности / Grundleistungen</i> а) Разработване на идейния проект при съобразяване със съществените взаимоотношения, изисквания и условия (например: градоустройствени, художествени, функционални, технически, икономически, екологични, строително-физични, енергийни, социални, публично-правни) на базата на предприемаческото проучване на планирането и като основа за по-нататъшните фази на изпълнението и

– създаването на устойчива, здравословна, безопасна, хигиенична и достъпна за всички архитектурна среда. Част архитектурна се представя в следните чертежи: 1. ситуационно решение, изработено върху геодезично заснемане, комбинирана скица от кадастралната карта (кадастралния план) и от действащия подробен устройствен план или върху извадка от действащия ПУП, в което се посочват точното местоположение на обекта, разстоянията до регулационните линии, между сградите в имота и до съседните сгради, постигнатите показатели на застрояване, площите за озеленяване, площите, осигуряващи необходимите места за паркиране на открито, както и подходът към уличната мрежа; 2. разпределения на всички етажи и план на покривните линии; 3. фасадни изображения, изясняващи външното оформление на обемите; 4. напречни и надлъжни вертикални разрези, изясняващи височините, нивата и вертикалната комуникация в сградата. По преценка на проектанта или в съответствие със заданията за проектиране (договора за проектиране) могат да се представят допълнително: 5. фрагменти на характерни и повтарящи се помещения с обзавеждане и елементи от фасадите; 6. перспективни и аксонометрични изгледи; 7. фотомонтажи, компютърна анимация, макети и др. Когато идейният проект служи за възлагане на строителство по реда на ЗУТ, разработката по част архитектурна съдържа допълнително: 8. количествена сметка за архитектурно-строителните работи; 9. спецификация на основните предвидени за влагане строителни продукти (материали, изделия, комплекти и системи) по част архитектурна с технически изисквания към тях в съответствие с действащи норми и стандарти; когато е предвидено със заданията (договора) за проектиране, към част архитектурна може да се приложи стойностна сметка на обекта.	за изискваните публично-правни одобрения с помощта на приноса на други специализирани участници в проектирането. Дигитален модел според вида и големината на обекта в необходимите обхват и ниво на детайлност, като се вземат предвид всички специфични по отделните специалности изисквания, например при сгради в степен на детайлност, която съответства на мащаб 1:100 / Erarbeiten der Entwurfsplanung, unter weiterer Berücksichtigung der wesentlichen Zusammenhänge, Vorgaben und Bedingungen (zum Beispiel städtebauliche, gestalterische, funktionale, technische, wirtschaftliche, ökologische, soziale, öffentlich-rechtliche) auf der Grundlage der Vorplanung und als Grundlage für die weiteren Leistungsphasen und die erforderlichen öffentlichen-rechtlichen Genehmigungen unter Verwendung der Beiträge anderer an der Planung fachlich Beteiligter. Digitales Modell nach Art und Größe des Objekts im erforderlichen Umfang und Detaillierungsgrad unter Berücksichtigung aller fachspezifischen Anforderungen, zum Beispiel bei Gebäuden in einer Detaillierung, die dem Maßstab 1:100 entspricht. b) Предоставяне на резултатите от горепосочените работи като основа за дейностите на другите проектанти по техническите специалности, както и координиране и интегриране на техните дейности при използване на собствения дигитален модел и на дигиталните модели на другите специализирани участници в проектирането / Bereitstellen der Arbeitsergebnisse als Grundlage für die anderen an der Planung fachlich Beteiligten sowie Koordination und Integration von deren Leistungen unter Verwendung des eigenen digitalen Modells und der digitalen Modelle der anderen fachlich an der Planung Beteiligten. c) Описание на строежа / Objektbeschreibung d) Преговори относно допустимостта на строителното начинание / Verhandlungen über die Genehmigungsfähigkeit e) Изчисление на разходите съгласно DIN 276 въз основа на количествата, изведени от дигиталните модели, и сравнение с предварителната преценка за разходите / Kostenberechnung nach DIN 276 auf der Basis der aus den digitalen Modellen abgeleiteten Mengen und Vergleich mit der Kostenschätzung
--	---

Фазата идеен проект може да послужи за изчисляване на приблизителна стойност на обекта и сравняване с предварителната оценка за стойността и продължаване на уточняване в графика за работа по обекта.	f) Актуализиране на графика / Fortschreiben des Terminplans g) Обобщаване, обяснение и документиране на резултатите / Zusammenfassen, Erläutern und Dokumentieren der Ergebnisse <i>Dопълнителни дейности / Besondere Leistungen</i> – Анализ на алтернативите/вариантите и тяхната преценка чрез изследване на разходите (оптимизация) / Analyse der Alternativen/Varianten und deren Wertung mit Kostenuntersuchung (Optimierung) – Изчисляване на рентабилността / Wirtschaftlichkeitsberechnung – Съставяне и актуализиране на задълбочено изчисление на разходите Aufstellen und Fortschreiben einer vertieften Kostenberechnung – Актуализиране на описания на помещения в сграда / Fortschreiben von Raumbüchern – Визуализиране на график в дигиталния модел / Visualisierung eines Terminplans im digitalen Modell – Изготвяне на изчисление на разходите, базирано на дигиталния модел / Aufstellen einer modellbasierten Kostenermittlung – Особени форми на проектиране и обработка на дигиталните модели на участващите в проектирането специалисти за комуникация и съгласуване / Besondere Präsentationsformen und Aufbereiten der digitalen Modelle der an der Planung fachlich Beteiligten zur Kommunikation und Abstimmung
Фаза 4 – Технически проект Част архитектурна на техническия проект се представя в следните чертежи: 1. ситуационно решение върху геодезично заснемане, комбинирана скица от кадастралната карта (кадастралния план) и от действащия подробен устройствен план или върху извадка от действащия ПУП, в което се посочват точното местоположение на обекта, разстоянията до	Фаза 4. Проект за разрешаване на строителството / LPh 4 Genehmigungsplanung <i>Основни дейности / Grundleistungen</i> a) Изработване и комплектоване на формулярите и на докладите за съответствие с цел публично-правни разрешения или съгласия, вкл. на исканията за изключения и освобождавания, както и необходимите

регулационните линии, между сградите в имота и до съседните сгради, постигнатите показатели на застрояване, площите за озеленяване, площите, осигуряващи необходимите места за паркиране на открито, както и подходът към уличната мрежа; 2. разпределения на всички етажи и план на покривните линии, изясняващи размерите и площите на всички помещения и на отворите в тях, предвидените материали или минималните изисквания към тях за обработката на стени, подове, тавани, стълбища и други части на сградата; 3. фасадни изображения, изясняващи външното оформление на обемите, употребените материали и тяхната обработка; 4. напречни и надлъжни вертикални разрези, изясняващи височините, нивата, вертикалната комуникация в сградата, наклоните на покривните равнини, изоляциите, подовите конструкции и настилки; 5. чертежи с подвижно и неподвижно архитектурно и технологично оборудване и обзавеждане; При договаряне с възложителя допълнително могат да се представят чертежи за: 6. фрагменти на характерни и повтарящи се помещения с обзавеждането; 7. фрагменти от елементи на фасадите; 8. характерни архитектурни детайли. В случаите, когато се предвижда изпълнение на строителството по одобрен технически проект, същият се допълва в процеса на строителството с необходимите детайли, достатъчни за изпълнението на СМР. Част архитектурна на техническия проект се придружава от обяснителна записка и количествена сметка за видовете архитектурно-строителни работи. Когато техническият проект служи за възлагане на строителство по реда на ЗОП съгласно ЗУТ, той съдържа спецификация на предвидените за влагане строителни продукти (материали, изделия,	преговори с административни органи, като се използват приносите на други специализирани участници в проектирането. Erarbeiten und Zusammenstellen der Vorlagen und Nachweise für öffentlich-rechtliche Genehmigungen oder Zustimmungen einschließlich der Anträge auf Ausnahmen und Befreiungen, sowie notwendiger Verhandlungen mit Behörden unter Verwendung der Beiträge anderer an der Planung fachlich Beteiligter b) Подаване на проекта за разрешаване на строителството, изведен от дигиталните модели / Einreichen der aus den digitalen Modellen abgeleiteten Vorlagen c) Допълване и преработване на чертежите, описанията и изчисленията / Ergänzen und Anpassen der Planungsunterlagen, Beschreibungen und Berechnungen <i>Допълнителни дейности / Besondere Leistungen</i> – Съдействие при осигуряване на съгласието на съседите / Mitwirken bei der Beschaffung der nachbarlichen Zustimmung – Доклади за съответствие, в частност от технически, конструктивен и строително-физически характер, за издаването на съгласия от административни органи в отделни случаи / Nachweise, insbesondere technischer, konstruktiver und bauphysikalischer Art, für die Erlangung behördlicher Zustimmungen im Einzelfall – Специализирана и организационна подкрепа на възложителя в производствата по обжалване, иски за производство и подобни процедури / Fachliche und organisatorische Unterstützung des Bauherrn im Widerspruchsverfahren, Klageverfahren oder ähnlichen Verfahren
---	---

комплекти и системи) по част архитектурна с технически изисквания към тях в съответствие с действащи норми и стандарти. Когато се изисква със заданието (договора) за проектиране, към част архитектурна се прилага стойностна сметка на обекта.	
Фаза 5 – Работен проект (Проект за изпълнение на строителството) Част архитектурна на работния проект прецизира и детайлизира решенията на идейния проект и/или на техническия проект и определя техническите характеристики на предвидените за влагане строителни продукти по част архитектурна (материали, изделия, комплекти и системи) и начини за изпълнението на обекта, а когато проектирането е еднофазно във фаза работен проект, дава цялостни архитектурни решения. Част архитектурна на работния проект представя освен решенията от ТП и решения за: – необходимите строителни продукти (материали, изделия, комплекти и системи) за изпълнение на СМР и начина на тяхната обработка, полагане и/или монтаж; – изпълнението на хидроизолациите, звукоизолациите и топлоизолациите на обекта; – осигуряването на естественото и изкуственото осветление в помещенията; – изпълнението на дограмата, настилките, облицовките и другите видове довършителни СМР в помещенията, покривите и фасадите; – акустичната обработка на стени и тавани, когато такава е необходима или се изисква със заданието за проектиране (договора за проектиране). Част архитектурна на работния проект се представя в следните чертежи: Освен чертежите от ТП: 1. допълнителни чертежи с подробни параметри и фрагменти от по-важни части на обекта, поясняващи проектните решения;	Фаза 5. Работен проект / LPh 5 Ausführungsplanung <i>Основни дейности / Grundleistungen</i> a) Разработване на Работния проект като дигитален модел с допълващи графични и текстови резултати, както и с цялата индивидуална информация, необходима за изпълнението (графична и текстова) на базата на Идейния проект и на Проекта за разрешаване на строителството до готово за изпълнение решение като основа за следващите фази на изпълнение / Erarbeiten der Ausführungsplanung als digitales Modell mit ergänzenden zeichnerischen und textlichen Arbeitsergebnissen sowie mit allen für die Ausführung notwendigen Einzelangaben (zeichnerisch und textlich) auf der Grundlage der Entwurfs- und Genehmigungsplanung bis zur ausführungsfähigen Lösung, als Grundlage für die weiteren Leistungsphasen b) Работни чертежи, чертежи на детайли и конструктивни чертежи като дигитален модел според вида и големината на обекта в необходимите обем и ниво на детайлност, като се вземат предвид всички специализирани изисквания, например за сгради в детайлност, която по правило съответства на мащаб 1:50. Плановите на детайли могат да се изготвят като допълнения от 2D-чертежи към дигиталния модел. / Ausführungs-, Detail- und Konstruktionszeichnungen als digitales Modell nach Art und Größe des Objekts im erforderlichen Umfang und Detaillierungsgrad unter Berücksichtigung aller fachspezifischen Anforderungen, zum Beispiel bei Gebäuden in einer Detaillierung, die im Regelfall dem Maßstab 1:50 entspricht. Detailplanungen können als 2D-zeichnerische Ergänzungen zum digitalen Modell erstellt werden.

2. архитектурни детайли в подходящ мащаб, изясняващи изпълнението на отделни СМР в интериора и екстериора на обекта, в т.ч. покриви, стени, тавани, подове, настилки, облицовки, изолации, дограми, козирки, стълби, парапети, комини, шахти и др.; Част архитектурна на работния проект се придружава от обяснителна записка, която освен описанията, посочени в записката на ТП, съдържа: 3. спецификация на предвидените за влагане строителни продукти (материали, изделия, комплекти и системи) по част архитектурна с технически изисквания към тях в съответствие с действащи норми и стандарти; Към част архитектурна на работния проект се прилага стойностна сметка на обекта, когато това е предвидено със заданието (договора) за проектиране. По искане на възложителя в част архитектурна на работния проект може да се изработи и решение за цвetoвo оформление на обекта (страдата). С решението за цвetoвo оформление се определят: цвetoвoо третиране и видът на фасадите, цокълните и покривните материали; местата и размерите на фасадните пана и реклами. Решението за цвetoвo оформление се представя в следните чертежи: 4. цвetoвo решение и материали на фасадите; 5. цвetoвo решение и материали на покривните линии; 6. характерни детайли на екстериора.	с) Предоставяне на резултатите от горепосочените работи като основа за дейностите на другите проектанти по техническите специалности, както и координиране и интегриране на техните дейности при използване на собствения дигитален модел и на дигиталните модели на другите специализирани участници в проектирането / Bereitstellen der Arbeitsergebnisse als Grundlage für die anderen an der Planung fachlich Beteiligten, sowie Koordination und Integration von deren Leistungen unter Verwendung des eigenen digitalen Modells und der digitalen Modelle der anderen fachlich an der Planung Beteiligten d) Актуализиране на графика / Fortschreiben des Terminplans e) Актуализиране на Работния проект въз основа на съобразена с отделните строителни специалности редакция по време на строителството / Fortschreiben der Ausführungsplanung auf Grund der gewerkeorientierten Bearbeitung während der Objektausführung f) Проверка на необходимите монтажни модели или планове на планираните от проектанта строителни конструкции и конструктивни елементи за съответствие с Работния проект / Überprüfen erforderlicher Montagemodelle oder -pläne der vom Objektplaner geplanten Baukonstruktionen und baukonstruktiven Einbauten auf Übereinstimmung mit der Ausführungsplanung <i>Допълнителни дейности / Besondere Leistungen</i> – Изготвяне на подробно Описание на обекта като основа за Описанието на дейностите заедно с Работна програма^х) / Aufstellen einer detaillierten Objektbeschreibung als Grundlage der Leistungsbeschreibung mit Leistungsprogramm^х) – Проверка на Работни проекти, изготвени от изпълняваща СМР фирма на база на Описанието на дейностите заедно с Работна програма, за съответствие с Идеияния проект^х / Prüfen der vom bauausführenden Unternehmen auf Grund der Leistungsbeschreibung mit Leistungsprogramm ausgearbeiteten Ausführungspläne auf Übereinstimmung mit der Entwurfsplanung^х
---	---

	– Актуализиране на подробни Описания на помещения в сграда / Fortschreiben von Raumbüchern in detaillierter Form – Съдействие при прилагане на Системата за означаване на технически съоръжения / Mitwirken beim Anlagenkennzeichnungssystem (AKS) – Проверка и приемане на проекти на трети страни, които не са проектанти по техническите специалности, за съответствие с работните проекти (напр. цехови чертежи от предприятия, монтажни и фундаментни планове за съоръжения със специфично предназначение или експлоатация), доколкото дейностите се отнасят до съоръжения, които не са включени в начислените разходи / Prüfen und Anerkennen von Plänen Dritter, nicht an der Planung fachlich Beteiligter auf Übereinstimmung mit den Ausführungsplänen (zum Beispiel Werkstattzeichnungen von Unternehmen, Aufstellungs- und Fundamentpläne nutzungsspezifischer oder betriebstechnischer Anlagen), soweit die Leistungen Anlagen betreffen, die in den anrechenbaren Kosten nicht erfasst sind – <u>Дигитално изготвяне на образци / Digitale Bemusterung</u> хТази особена дейност се превръща при Описане на дейностите заедно с Работна програма изцяло или отчасти в основна дейност. При този случай отпадат съответните основни дейности в тази фаза на изпълнение. / Diese besondere Leistung wird bei Leistungsbeschreibung mit Leistungsprogramm ganz oder teilweise Grundleistung. In diesem Fall entfallen die entsprechenden Grundleistungen dieser Leistungsphase.
Фаза 6 – Изготвяне на количествени сметки и оформянето им като тържни книжа В тази фаза архитект-проектантът извършва следните видове проектантски услуги: 1. обсъжда с възложителя изпълнението на проекта с предвидените материали; 2. окончателно уточнява изискванията към влаганите материали;	Фаза 6 Подготовка за възлагане на строителството / LPH 6 Vorbereitung der Vergabe <i>Основни дейности / Grundleistungen</i> а) Изготвяне на график за възлагане на строителството / Aufstellen eines Vergabeterminplans

3. изготвя спецификации на врати и прозорци, обзавеждане и оборудване; 4. изчислява количествата на материалите и строително-монтажните работи; 5. изработва подробни количества, описания и спецификации на основата на работното проектиране, включително количествата на всички останали специалности; 6. извежда стойността на обекта според стойността по отделните части; 7. оформя цялостна документация за провеждане на процедура по избор на изпълнител; 8. определя срок за възлагане.	b) Изготвяне на описания на видовете работи заедно с количествени сметки по раздели, изчисляване и обединяване на количествата на база на Работния проект, по-специално на дигиталния модел, при използване на приноса на другите участващи в проектирането специалисти и на техните дигитални модели / Aufstellen von Leistungsverzeichnissen mit Leistungseinheiten nach Leistungsbereichen, Ermitteln und Zusammenstellen von Mengen auf der Grundlage der Ausführungsplanung, insbesondere des digitalen Modells, unter Verwendung der Beiträge anderer an der Planung fachlich Beteiligter und deren digitaler Modelle c) Съгласуване и координиране на пресечните точки с описанията на видовете работи на участващите в проектирането специалисти / Abstimmen und Koordinieren der Schnittstellen zu den Leistungsbereichen der an der Planung fachlich Beteiligten d) Изчисляване на разходите на базата на остойностени от проектанта количествени сметки / Ermitteln der Kosten auf der Grundlage vom Planer bereiteter Leistungsverzeichnisse e) Контрол на разходите чрез сравнение на остойностените от проектанта количествени сметки с прогнозното изчисление на разходите / Kostenkontrolle durch Vergleich der vom Planer bereiteten Leistungsverzeichnisse mit der Kostenberechnung f) Комплектуване на тръжните книжа по всички раздели на работите / Zusammenstellen der Vergabeunterlagen für alle Leistungsbereiche <i>Допълнителни дейности / Besondere Leistungen</i> – Изготвяне на описанията на работите заедно с работна програма въз основа на детайлно описание на обекта* / Aufstellen der Leistungsverzeichnissen mit Leistungsprogramm auf der Grundlage der detaillierten Objektbeschreibung* – Изготвяне на алтернативни описания на обособени строителни работи / Aufstellen von alternativen Leistungsbeschreibungen für geschlossene Leistungsbereiche
---	--

	– Изготвяне на сравнителни прегледи на разходите, като се вземат предвид и приносите на другите участващи в проектирането специалисти / Aufstellen von vergleichenden Kostenübersichten unter Auswertung der Beiträge anderer an der Planung fachlich Beteiligter * При описание на работите заедно с работна програма тази допълнителна работа се превръща цялостно или частично в основна работа. В този случай отпадат съответните основни работи от тази фаза на изпълнение. / Diese Besondere Leistung wird bei einer Leistungsbeschreibung mit Leistungsprogramm ganz oder teilweise zur Grundleistung. In diesem Fall entfallen die entsprechenden Grundleistungen dieser Leistungsphase.
Фаза 7 – Съдействие при организация и провеждане на търг/търгове за избор на изпълнител/и В тази фаза архитект-проектантът извършва следните видове проектантски услуги: 1. съставя генерална количествено-стойностна сметка по всички специалности, ангажирани с проектирането; 2. обсъжда с възложителя начините на възлагане на строителството; 3. подготвя тържна документация; 4. подготвя общите условия и изисквания за провеждане на търг за избор на изпълнител/и; 5. обявява и провежда търг (отделни търгове); 6. изготвя сравнителна таблица по показатели на офертите, сравнява единичните цени и количества от офертите; 7. извършва проверка на променени позиции и описания на строителни дейности; 8. изготвя препоръчителна оценка; 9. съдействие при изготвяне на договора за строителство.	Фаза 7. Съдействие при възлагането на строителството / LPH 7 Mitwirkung bei der Vergabe <i>Основни дейности / Grundleistungen</i> a) Координиране на процедурите по договаряне с проектантите по инженерните специалности / Koordinieren der Vergaben der Fachplaner b) Получаване на офертите / Einholen von Angeboten c) Проверка и оценяване на офертите, включително изготвяне на сравнителна таблица по отделни позиции или по частични работи, проверка и оценяване на офертите относно допълнително добавени или променени видове работи на изпълняващите предприятия и на обособеността на цените / Prüfen und Werten der Angebote einschließlich Aufstellen eines Preisspiegels nach Einzelpositionen oder Teilleistungen, Prüfen und Werten der Angebote zusätzlicher und geänderter Leistungen der ausführenden Unternehmen und der Angemessenheit der Preise d) Водене на разговори с оферентите / Führen von Bietergesprächen e) Изготвяне на предложения за възлагане на строителството, документиране на процедурите по възлагането / Erstellen der Vergabevorschläge, Dokumentation des Vergabeverfahrens

	f) Комплектуване на договорите за всички раздели на строителството / Zusammenstellen der Vertragsunterlagen für alle Leistungsbereiche g) Сравняване на резултатите от търга с остойностите от проектантите количествени сметки или с прогнозното остойностяване на разходите / Vergleichen der Ausschreibungsergebnisse mit den vom Planer berechneten Leistungsverzeichnissen oder der Kostenberechnung h) Съдействие при сключване на договора / Mitwirken bei der Auftragserteilung <i>Допълнителни дейности / Besondere Leistungen</i> – Проверяване и оценяване на допълнителни оферти и тяхното влияние върху съгласуваните проекти / Prüfen und Werten von Nebenangeboten mit Auswirkungen auf die abgestimmte Planung – Съдействие при планиране на разплащанията / Mitwirken bei der Mittelabflussplanung – Подготовка и съдействие от страна на проектантите по инженерните специалности при допълнителни проверки по възражения от оференти / Fachliche Vorbereitung und Mitwirken bei Nachprüfungsverfahren – Съдействие при проверка на икономически аргументирани допълнителни оферти / Mitwirken bei der Prüfung von bauwirtschaftlich begründeten Nachtragsangeboten – Проверяване и оценяване на описанието на работите заедно с работната програма към офертите, включително сравнителна таблица с цените* / Prüfen und Werten der Angebote aus Leistungsbeschreibung mit Leistungsprogramm einschließlich Preisspiegel* – Изготвяне, проверяване и оценяване на сравнителните таблици с цените при специални изисквания / Aufstellen, Prüfen und Werten von Preisspiegeln nach besonderen Anforderungen * При описание на работите заедно с работна програма тази допълнителна работа се превръща цялостно или частично в основна работа. В този случай отпадат съответните основни работи от тази фаза
--	---

	на изпълнение. / *Diese Besondere Leistung wird bei Leistungsbeschreibung mit Leistungsprogramm ganz oder teilweise Grundleistung. In diesem Fall entfallen die entsprechenden Grundleistungen dieser Leistungsphase.	
Фаза 8 – Инвеститорски и авторски контрол, координация на строителния процес до въвеждане на обекта в експлоатация В тази фаза архитект-проектантът извършва следните видове проектантски услуги: 1. изпълнява цялостен инвеститорски и архитектурен контрол върху изпълнението на СМР и тяхното съответствие с изготвените проекти; 2. участва в изготвянето на календарен график за изпълнение от фирмата/фирмите- изпълнител/и и контролира неговото спазване; координира работата на инженерните специалности, изпълняващи авторски и инвеститорски контрол; 3. приема завършени етапи от СМР като качество и количество, писмено отразява евентуални пропуски и недостатъци на този етап с копие до възложителя; намира и прилага решения за незабавното им отстраняване, включително и детайли; 4. проверява стойностите по представените актове и авансови искове и тяхното съответствие с офертните цени, подписва и представя на възложителя съгласие за изплащането им, ако са изпълнени условията за това; 5. извършва корекция и съгласуване на архитектурни детайли в процеса на строителството, дава устни и писмени указания (в заповедната книга) на строителите при възникнали неясноти, несъответствия с проектите, възникнали по време на строителството, за които писмено се уведомява възложителят; 6. извършва контрол и евентуална корекция на елементите, доставени от изпълнителя/изпълнителите; 7. изготвя екзекутивни чертежи.	Фаза 8. Надзор на строежа (строителен надзор) и документация / Objektüberwachung (Bauüberwachung) und Dokumentation <i>Основни дейности / Grundleistungen</i> a) / Надзираване на изпълнението на обекта относно съответствие с публичноправното разрешение или съгласие, договорите със строителните предприятия, документацията за изпълнение (дигиталния модел), съответните разпоредби и общопризнатите правила на техниката / Überwachen der Ausführung des Objektes auf Übereinstimmung mit der öffentlich-rechtlichen Genehmigung oder Zustimmung, den Verträgen mit ausführenden Unternehmen, den Ausführungsunterlagen (dem digitalen Modell), den einschlägigen Vorschriften sowie mit den allgemein anerkannten Regeln der Technik b) Надзираване на изпълнението на конструкции с много ниски и ниски изисквания за проектиране относно съответствие с Доклада за съответствие на проекта със строителните норми относно конструктивната устойчивост / Überwachen der Ausführung von Tragwerken mit sehr geringen und geringen Planungsanforderungen auf Übereinstimmung mit dem Standardsicherheitsnachweis c) / Координиране на специализираните участници в надзора на строежа / Koordinieren der an der Objektüberwachung fachlich Beteiligten d) Създаване, актуализиране и надзираване на график (стъпбавидна диаграма) / Aufstellen, Fortschreiben und Überwachen eines Terminplans (Balkendiagramm) e) Документиране на строителния процес (напр. строителен дневник) / Dokumentation des Bauablaufs (zum Beispiel Bautagebuch)	

	f) Съвместни със строителните предприятия измервания или дигитални установявания на осъществени работи / Gemeinsames Aufmaß oder digitale Leistungsfeststellung mit den ausführenden Unternehmen g) Проверка на фактури, включително проверка на измерванията или на моделите за отчитане, на строителните предприятия / Rechnungsprüfung einschließlich Prüfen der Aufmaße oder Abrechnungsmodelle der bauausführenden Unternehmen h) Сравнение на резултатите от проверките на фактурите със сумите по поръчката, включително за допълнителните дейности. / Vergleich der Ergebnisse der Rechnungsprüfungen mit den Auftragssummen einschließlich Nachträgen i) Контрол на разходите чрез проверка на фактурирането на изпълнението на строителните предприятия в сравнение с договорните цени / Kostenkontrolle durch Überprüfen der Leistungsabrechnung der bauausführenden Unternehmen im Vergleich zu den Vertragspreisen j) Установяване на разходите, например съгласно DIN 276, ако е приложимо въз основа на изведените от дигиталния модел количества / Kostenfeststellung, zum Beispiel nach DIN 276, ggf. auf Basis der aus dem digitalen Modell abgeleiteten Mengen k) Организация на приемането на строителните работи с участието на други специалисти, участващи в проектирането и надзора на строежа, констатиране на дефекти, препоръки към възложителя относно приемането на обекта / Organisation der Abnahme der Bauleistungen unter Mitwirkung anderer an der Planung und Objektüberwachung fachlich Beteiligter, Feststellung von Mängeln, Abnahmeempfehlung für den Auftraggeber l) Заявление за приемане по реда на публичното право и участие в него / Antrag auf öffentlich-rechtliche Abnahmen und Teilnahme daran m) Систематично съставяне на документацията, на графичните изображения/дигитални модели и на резултатите от изчисленията на обекта / Systematische Zusammenstellung der Dokumentation, zeichnerischen Darstellungen/digitalen Modelle und rechnerischen Ergebnisse des Objekts
--	--

	n) Предаване на обекта / Übergabe des Objekts o) Списък на давностните срокове за искове относно дефекти / Auflisten der Verjährungsfristen für Mängelansprüche p) Надзираване на отстраняването на дефектите, установени при приемането на обекта / Überwachen der Beseitigung der bei der Abnahme festgestellten Mängel <i>Допълнителни дейности / Besondere Leistungen</i> – Създаване, надзираване и актуализиране на план за плащане / Aufstellen, Überwachen und Fortschreiben eines Zahlungsplanes – Създаване, надзираване и актуализиране на диференцирани планове за време, разходи или капацитет / Aufstellen, Überwachen und Fortschreiben von differenzierten Zeit-, Kosten- oder Kapazitätsplänen – Дейност като отговорен Ръководител на строителството, доколкото тази дейност съгласно съответното провинциално право надхвърля основните дейности по Фаза на изпълнение 8 / Tätigkeit als verantwortlicher Bauleiter, soweit diese Tätigkeit nach jeweiligem Landesrecht über die Grundleistungen der LPH 8 hinausgeht – Обхващане на напредъка на строителството в дигиталния модел / Erfassung des Baufortschritts im digitalen Modell – Базирано на дигиталния модел управление на дефектите / Modellbasiertes Mängelmanagement – Изготвяне на модел на логистиката на строителството / Erstellen eines Baulogistikmodells – Изготвяне на екзекутивен модел / Erstellen eines as-built-Modells – Изготвяне на модел за фасилити мениджмънт / Erstellen eines Facility-Management-Modells
Фаза 9 – Комплектоване на архивна документация, установяване на дефекти по обекта и съдействие за отстраняването им в рамките на гаранционните срокове	Фаза 9. Съдействие при гаранционното обслужване на сгради / LPH 9 Objektbetreuung <i>Основни дейности / Grundleistungen</i>

В тази фаза архитектът може да извършва следните дейности: 1. регистрира и извършва технически измервания на деформации и повреди; 2. разследва и установява причините за повредите; 3. описва и документира повредите; 4. организира изпълнение на мероприятия за защита и наблюдаване на обекта в полза на ползвателя му; 5. изготвя специализирана оценка за възможни претенции на установените недостатъци в рамките на гаранционните срокове в продължение на не повече от петгодишна давност от датата на приемането.	a) Специализирана оценка на дефектите, установени в рамките на давностните срокове за гаранционни рекламации, но не повече от пет години след приемане на строителната дейност, включително необходимите обхождания./ Fachliche Bewertung der innerhalb der Verjährungsfristen für Gewährleistungsansprüche festgestellten Mängel, längstens jedoch bis zum Ablauf von fünf Jahren seit Abnahme der Leistung, einschließlich notwendiger Begehungen b) Обход на строежа с цел установяване на дефекти преди изтичането на давностния срок за предявяване на искове за дефекти срещу предприятието изпълнител / Objektbegehung zur Mängelfeststellung vor Ablauf der Verjährungsfristen für Mängelansprüche gegenüber den ausführenden Unternehmen c) Съдействие при освобождаването на гаранционни депозити / Mitwirken bei der Freigabe von Sicherheitsleistungen <i>Допълнителни дейности / Besondere Leistungen</i> – Надзор на развитието и поддръжката на озеленяването / Überwachung der Entwicklungs- und Unterhaltungspflege – Надзираване на дейностите по поддръжка / Überwachen von Wartungsleistungen – Надзираване на отстраняването на дефекти в рамките на съответния давностен срок / Überwachen der Mängelbeseitigung innerhalb der Verjährungsfrist
--	---

Приложение 5. Решение №1 на КАБ от 12.04.2024 г.

към

Дисертация за присъждане на
образователна и научна степен “доктор“

на тема

**Организационни модели за създаване на сгради в България и Бавария
и формиране на професионалните задачи на архитектите**

**РЕШЕНИЕ****№ 1 от 12.04.2024 г.****по чл. 14а от ЗКАИИП във връзка с чл. 15 от ЗППК**

Председателят на Управителния съвет на Камарата на архитектите (КАБ), като разгледа подаденото заявление от арх. Стоян Георгиев Тодоров (заявител) с вх. № КР-050А/29.01.2024 г. (заявлението) чрез Комисията по регистъра, с което на основание чл. 14а от Закона за камарите на архитектите и инженерите в инвестиционното проектиране (ЗКАИИП) във връзка с чл. 15 от Закона за признаване на професионални квалификации (ЗППК) заявителят желае да бъде вписан служебно в регистъра на Камарата на архитектите с пълна проектантска правоспособност по целия обхват на инвестиционните проекти (всички проектни части) в строителството и да получи удостоверение за вписване като временно (за срок от една година) предоставящ услуги в областта на архитектурата и на инвестиционното проектиране в целия обхват на инвестиционните проекти в строителството на територията на страната, позовавайки се на разпоредби от немското законодателство, в качеството си на орган по признаване на професионални квалификации за професията „архитект“ съгласно чл. 4, ал. 1, т. 3 от ЗППК, за да се произнесе взе предвид следното:

1. Заявителят арх. Стоян Георгиев Тодоров излага, че е гражданин на Федерална Република Германия, установен в гр. Мюнхен, и архитект, предоставящ от 2009 г. проектантски услуги в Германия в областта на архитектурата и на инвестиционното проектиране в целия обхват (всички проектни части) на инвестиционните проекти в строителството съгласно: (1) чл. 3, ал. 1, ал. 6 и ал. 7 от Закона за Баварската камара на архитектите и Баварската камара на инженерите в сектор „Строителство“ от 9 май 2007 г.; (2) чл. 51, ал. 1 и ал. 2, чл. 61, ал. 2, т. 1 от Баварския строителен закон от 14 август 2007 г.; (3) т. 10.1 от Приложение 10 – Основни дейности в спектъра от услуги относно сгради и вътрешни пространства, опълнителни дейности, Списъци на видовете строителни обекти, включително деветте Фази на изпълнение към германската Наредба за хонорарите за услугите на архитектите и инженерите; (4) § 5 от Регламент за изпълнение на разпоредбите относно енергийния сектор от 22 януари 2002 г.

2. По служебен път се установи, че заявителят е преминал през процедура по признаване на професионалната квалификация по архитектура чрез Министерството на образованието на Република България през 2006 г. С удостоверение № 31-33/09.10.2006 г. министърът на образованието на Република България е признал висшето образование на Стоян Георгиев Тодоров за образователно-квалификационна степен магистър по архитектура в Технически университет гр. Мюнхен, Германия.

3. По служебен път се установи, че заявителят е кандидатствал за придобиване на пълна проектантска правоспособност (ППП) през 2007 г. пред Управителния съвет (УС) на КАБ чрез Регионална колегия (РК) Стара Загора. Въз основа на заявлението си от 2007 г. арх. Стоян Георгиев Тодоров е придобил PPP, вписан е в регистъра на архитектите с PPP с регистрационен № 03385 РК Стара Загора и му е издадено удостоверение за PPP и личен печат, с решение на



УС на КАБ от 02.08.2007 г. по т.5.1 от протокол № 35/ 02.08.2007 г. по предложение на Комисията по регистъра към УС на КАБ по т. 2.3 от протокол № 96/18.07.2007 г.

4. От датата на вписването на заявителя в регистъра на архитектите с ППП (02.08.2007 г.) до датата на подаване на настоящото заявление (29.01.2024 г.), арх. Стоян Георгиев Тодоров е редовен член на КАБ, вписан в регистъра на архитектите с пълна проектантска правоспособност към КАБ, включително за 2024 г. Проектантската правоспособност на арх. Стоян Георгиев Тодоров към датата на подаване на заявлението и през целия период от 02.08.2007 г. до 29.01.2024 г. не е променяна на някое от основанията по чл. 14, ал. 2 от ЗКАИИП, и същият не е бил отписан от регистъра на архитектите с ППП нито по негова молба, нито въз основа на решение на УС на КАБ.

5. Въз основа на изложените факти в т. 2, 3 и 4 от настоящото решение се налага изводът, че заявление с вх. № КР-050А/29.01.2024 г. на арх. Стоян Георгиев Тодоров за вписване в регистъра на архитектите с ППП и за издаване на удостоверение се явява второ поред, налице е влязъл в сила акт със същия предмет и страни – Решение на Управителния съвет на КАБ от 02.08.2007 г. по т.5.1 от протокол № 35/ 02.08.2007 г. по заявление на арх. Стоян Георгиев Тодоров, поради което заявлението е недопустимо на основание чл. 27, ал. 2, т. 1 от Административнопроцесуалния кодекс (АПК).

6. Като председател на УС на КАБ и орган по признаване по закон на професионалната квалификация на професията „архитект“, извън установената по-горе недопустимост на искането, обръщам внимание и на негова недопустимост и поради липса на компетентност.

6.1. В частта от заявлението, в която се иска освен проектантска правоспособност за професионалната квалификация „архитект“ и вписване в регистъра на КАБ и издаване на удостоверение за вписване от КАБ *„по целия обхват на инвестиционните проекти (всички проектни части) в строителството“ и „предоставящ услуги в областта на архитектурата и на инвестиционното проектиране в целия обхват на инвестиционните проекти в строителството на територията на страната“*, според националното законодателство е извън компетенциите на председателят на УС на КАБ, тъй като е извън обхвата на професионалната квалификация за професията „архитект“.

6.2. Съгласно чл. 5, ал. 1 от ЗППК, граждани на държави членки, които притежават доказателства за професионални квалификации, придобити в друга държава членка, могат да упражняват регулирана професия в Република България в съответствие с българските закони.

6.3. Съгласно чл. 3 от ЗППК, списъкът на регулираните професии в Република България се приема с решение на Министерския съвет. С Решение на Министерски съвет № 619/20.07.2009 г. са утвърдени регулираните професии „архитект“ и „инженер в инвестиционното проектиране“. Редът за упражняване на регулираните професии, чрез които се предоставят проектантски услуги в областта на инвестиционното проектиране и устройственото планиране, „архитект“ и „инженер в инвестиционното проектиране“, е уреден в ЗКАИИП, като обхватът на двете професии не съвпада.

6.4. Професионалната квалификация „архитект“ на заявителя не му дава право в Република България да предоставя проектантски услуги по всички проектни части на инвестиционния проект, които в своята цялост и пълен обхват включват освен изработване на инвестиционни проекти в областта на архитектурата, така също и на проектни части, които



попадат в обхвата на професионалната квалификация на регулираната професия „инженер в инвестиционното проектиране“, като: част „Конструктивна“, „Електрическа“, „Водоснабдяване и канализация“, „Отопление, вентилация и климатизация“ и др.

6.5. Съгласно чл. 230, ал. 5 от Закона за устройство на територията (ЗУТ), чужденци и граждани на държави - членки на Европейския съюз, или на други държави – страни по Споразумението за Европейското икономическо пространство, на които е призната професионалната квалификация по реда на ЗППК, могат да извършват дейностите по чл. 229, ал. 1 от ЗУТ (проучвателни, проектантски, контролни, строителни, надзорни) в обхвата на квалификацията си при условията на ЗКАИИП.

6.6. Съгласно чл. 7, ал. 7 от ЗКАИИП, проектантите с ППП, вписани в регистъра на съответната камара, в съответствие с придобитата професионална квалификация могат да предоставят проектантски услуги в областта на устройственото планиране и инвестиционното проектиране без ограничения по вид и размер, да договарят участие в инженеринг на строежи и да упражняват контрол по изпълнението на проектите им. Съгласно чл. 7, ал. 8 от ЗКАИИП, при подписването на проектите проектантите с ППП използват собствен печат, съдържащ името и регистрационния им номер, а за инженерите - и частта на проектната документация съобразно професионалната квалификация по специалност от регулираната професия „инженер в инвестиционното проектиране“.

6.7. Професионалните квалификации, които попадат в обхвата на регулираната професия „инженер в инвестиционното проектиране“ подлежат на признаване от председателя на Камарата на инженерите в инвестиционното проектиране съгласно чл. 4, ал. 1, т. 4 от ЗППК, а не от председателя на УС на КАБ.

6.8. Съгласно чл. 16, ал. 1 от ЗППК, гражданите на държава членка, които временно предоставят услуги в областта на инвестиционното проектиране, се вписват служебно в регистъра на Камарата на инженерите в инвестиционното проектиране и получават удостоверение за вписване.

6.9. В тази връзка, заявлението в частта, в която се иска вписване в регистъра на КАБ и издаване на удостоверение от КАБ „по целия обхват на инвестиционните проекти (всички проектни части) в строителството“ и „предоставящ услуги в областта на архитектурата и на инвестиционното проектиране в целия обхват на инвестиционните проекти в строителството на територията на страната“ на арх. Стоян Георгиев Тодоров, освен недопустимо на основание чл. 27, ал. 2, т. 6 от АПК е недопустимо и поради това, че компетентен да се произнесе по искането в частта, излизаща извън обхвата на професионалните квалификации за професията „архитект“, е председателят на УС на Камарата на инженерите в инвестиционното проектиране.

7. Предвид изложената фактическа и правна обстановка, след проверка на представените от заявителя документи, служебно събраната информация и прилагане на относимите норми на ЗППК, ЗКАИИП, ЗУТ и АПК се налага изводът, че заявление с вх. № КР-050А/ 29.01.2024 г. на арх. Стоян Георгиев Тодоров е недопустимо поради наличие на влязъл в сила административен акт със същия предмет и страни – Решение на УС на КАБ от 02.08.2007 г. по т. 5.1 от протокол № 35/ 02.08.2007 г. по заявление на арх. Стоян Георгиев Тодоров за вписване в регистъра на КАБ на архитектите с ППП и за издаване на удостоверение за ППП, и



поради това, че не отговаря на другите специални изисквания, установени със закон - чл. 7, ал. 7 и 8 от ЗКАИИП във вр. с чл. 5, ал. 1, чл. 4, ал. 1, т. 4 и чл. 16 от ЗППК.

Воден от горното, на основание чл. 27, ал. 2, т. 1 и 6 от АПК

Р Е Ш И Х :

Прекратявам производството по заявление с вх. № КР-050А/29.01.2024 г. на арх. Стоян Георгиев Тодоров в частта от искането за вписване в регистъра за пълна проектантска правоспособност на лицата, упражняващи професията „архитект“.

Копие от Решението и от административната преписка, на основание чл. 31, ал. 2 от Административнопроцесуалния кодекс, да се изпрати на председателя на УС на Камарата на инженерите в инвестиционното проектиране в Република България за произнасяне в частта от искането на заявителя в неговите компетенции.

Решението да се съобщи на заявителя по реда на Административнопроцесуалния кодекс.

Решението подлежи на обжалване в 14-дневен срок от съобщаването му по реда на чл. 197 от АПК.

Дата: 12.04.2024

арх. Владимир Милков
Председател на Управителния съвет на
Камара на архитектите в България



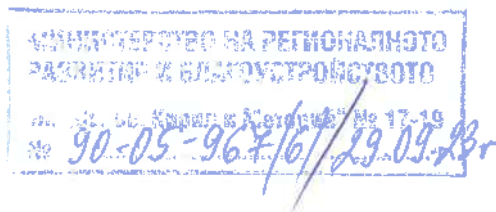
Приложение 6. Жалба до АССГ.

към

Дисертация за присъждане на
образователна и научна степен “доктор“

на тема

**Организационни модели за създаване на сгради в България и Бавария
и формиране на професионалните задачи на архитектите**



ЧРЕЗ
МИНИСТЪРА НА РЕГИОНАЛНОТО
РАЗВИТИЕ И БЛАГОУСТРОЙСТВОТО

ДО
АДМИНИСТРАТИВЕН СЪД
СОФИЯ-ГРАД

ЖАЛБА

от: арх. Станимир Христов Савов с ЕГН 7407111066,
с постоянен адрес: гр. Варна, ж.к. "Възраждане";
бл. 53; вх. 1, ап. 11, чрез адвокат Силвия
Русанова от Софийска адвокатска колегия,
вписана с личен номер 1000488610 в единния
адвокатски регистър, с адрес на упражняване на
дейността: гр. София, бул. „Витоша“ № 10, ет. 2,
ап. 8, електронна поща:
sylvierussanova@gmail.com, телефон за контакт:
0889 227838

срещу: Акт изх. № 90-05-967-(4)/12.09.2023 г. на
министъра на регионалното развитие и
благоустройството относно правоспособността
на проектантите от професия „инженер в
инвестиционното проектиране“ и регулираната
професия „архитект“ да изработват част
„Енергийна ефективност“ и част „Пожарна
безопасност“ на инвестиционните проекти

УВАЖАЕМИ Г-Н/Г-ЖО АДМИНИСТРАТИВЕН СЪДИЯ,

В срока по чл. 179 от Административнопроцесуалния кодекс /АПК/, на основание чл. 146, т. 1, т. 2, т. 3, т. 4 и т. 5 от АПК, оспорвам Акт изх. № 90-05-967-(4)/12.09.2023 г. на министъра на регионалното развитие и благоустройството относно правоспособността на проектантите от професия „инженер в инвестиционното проектиране“ и регулираната професия „архитект“ да изработват част „Енергийна ефективност“ и част „Пожарна безопасност“ на инвестиционните проекти.

Считам, че процесният акт е издаден при липса на компетентност на административния орган, при неспазване на установената форма, при съществено нарушение на административнопроизводствените правила, същият противоречи на материалноправни разпоредби и не е съобразен с целта на закона, поради което са налице релевантите от мен отменителни основания по чл. 146, т. 1 – т. 5 от АПК.

I. Допустимост на жалбата:

I.1. Обжалваемост на акта

Актът, предмет на настоящата жалба, представлява общ административен акт по смисъла на чл. 65 от АПК. Съгласно посочената разпоредба общи са административните

актове с еднократно правно действие, с които се създават права или задължения или непосредствено се засягат права, свободи или законни интереси на неопределен брой лица, както и отказите да се издадат такива актове.

I.2. Правен интерес

Правото ми на оспорване на общия административен акт се обуславя от качеството ми на проектант с пълна проектантска правоспособност, вписан в Регистъра на Камара на архитектите в България под № 02716, упражняващ регулираната професия „архитект“. Като лице с пълна проектантска правоспособност съм носител и на правото да изработвам, а съответно съм и изработвал през различни периоди от време, т. нар. от министъра част „Енергийна ефективност“ и част „Пожарна безопасност“. Съгласно Удостоверение за проектантска правоспособност изх. номер 230101U7957, издадено на 01.01.2023 г. от Камара на архитектите в България, обхватът на проектантската ми правоспособност, в съответствие с придобитата проектантска квалификация включва проектантски услуги в областта на устройственото планиране и инвестиционното проектиране без ограничения по вид и размер, договаряне участие в инженеринг на строежи и да упражняване на контрол по изпълнението на проектите. (чл.7, ал.7, изр.1 от ЗКАИИП). В удостоверението е посочено, че към тези проекти архитектите с пълна проектантска правоспособност могат да предоставят услуги и за всички видове устройствени планове, "Генерален план", "Интериор и дизайн", "Благоустройство", "Пожарна безопасност", "План за безопасност и здраве", "Енергийна ефективност", "План за управление на строителните отпадъци" и други в съответствие с придобитата им професионална квалификация.

I.3 Срок

Съгласно разпоредбата на чл. 179 от АПК общите административни актове могат да се оспорят в едномесечен срок от съобщението за издаването им или в 14-дневен срок от отделните съобщения до лицата, участвали в производството пред административния орган. Доколкото Камара на архитектите в България, чийто член съм и аз, е посочена като участник в производството, и същата е получила акта на министъра на 15.09.2023 г., а съответно на същата дата го е изпратила на електронна поща и на Регионална колегия – град Варна, подаването на жалбата в 14-дневния срок обуславя извода за спазването на този преклузивен срок.

II. Основателност на жалбата

Считам, че оспореният акт е издаден при липса на компетентност на органа, който го е издал, същият противоречи на материалноправни разпоредби, при издаването му са допуснати особено съществени нарушения на административнопроизводствените правила и не е спазена установената форма, както и не е съответен на целта на закона. Тези нарушения са толкова съществени, че обуславят нищожността на акта, респективно неговата незаконосъобразност.

Съображенията ми за това са следните:

С акт изх. № 90-05-967-(4)/12.09.2023 г. министърът на регионалното развитие и благоустройството е формулирал няколко принципни съображения, с които е приел, че липсва съвпадение по минимални образователни изисквания между професионалната квалификация „Архитект“ и тези по части „Енергийна ефективност“ и „Пожарна безопасност“, като в този смисъл е посочил, че наличието на общи познания в дадена област не дават правото за упражняване на регулирана професия при наличието на

минимални държавни образователни изисквания по отношение на резултатите от обучението. В изложението на административния орган се съдържа и извод, че удостоверения за проектантска правоспособност за части „Енергийна ефективност“ и „Пожарна безопасност“, като специалности на регулираната професия „инженер в инвестиционното проектиране“, трябва да се издават от Камара на инженерите в инвестиционното проектиране (КИИП). В частта „Архитектура“ на инвестиционния проект министърът е посочил, че същата може да се разработва единствено от лица с висше образование по специалността „Архитектура“ и образователно-квалификационна степен „магистър“ с професионална квалификация „архитект“, като местно компетентният на национално ниво орган, който издава удостоверения за проектантска правоспособност на архитектите, е Камара на архитектите в България.

Въз основа на извършения в акта анализ на нормативната уредба министърът е достигнал до обобщаващия извод, че всеки архитект, желаещ да придобие проектантски права за разработването на т. нар. част „Енергийна ефективност“ или част „Пожарна безопасност“, е необходимо да премине и завърши успешно съответното обучение с минимален хорариум от учебния план, като впоследствие удостоверението му за проектантска правоспособност ще бъде издадено от КИИП.

За да достигне до тези съждения министърът се е позовал както на норми от европейското законодателство, така и на вътрешни правни норми, съответно са цитирани разпоредбите на Директива 2005/36/ЕО на Европейския парламент и на Съвета от 7 септември 2005 година относно признаването на професионалните квалификации (Директива 2005/36/ЕО), Закона за камарите на архитектите и инженерите в инвестиционното проектиране (ЗКАИИП), Законът за устройство на територията (ЗУТ) и др.

Считам, че акта на министъра е нищожен, тъй като същият е издаден от некомпетентен по материя орган, както и при липсата на нормативно основание за това.

С разпоредбата на чл. 230, ал. 1 от ЗУТ е предвидено, че устройствени планове и инвестиционни проекти по същия закон се изработват от проектанти - физически лица, които притежават съответната техническа и проектантска правоспособност, като условията и редът за признаване на пълна проектантска правоспособност се определят със закон. Този закон е ЗКАИИП, който регламентира условията и реда за придобиване, признаване и изгубване на проектантска правоспособност за упражняване на професиите "архитект" и "инженер" в устройственото планиране и инвестиционното проектиране, както и устройството и организацията на дейността на професионалните им организации.

Легалната дефиниция на понятието „проектант“ се съдържа в § 1, т. 1 от Допълнителните разпоредби на ЗКАИИП, според чието установяване това е физическо лице, притежаващо диплома от акредитирано висше училище с професионална квалификация „архитект“, „ландшафтен архитект“, „урбанист“, „строителен инженер“ или „инженер“ с образователно-квалификационна степен „бакалавър“ или „магистър“, вписано в съответния регистър на Камарата на архитектите, съответно на Камарата на инженерите в инвестиционното проектиране, както и лице с призната проектантска правоспособност при условията и по реда на чл. 10 и 11, което има право да изработва концепции и схеми за пространствено развитие, устройствени планове и инвестиционни проекти в съответствие с придобитата правоспособност.

ЗКАИИП е определил камарите на архитектите и на инженерите в инвестиционното проектиране като юридически лица със съответните органи, чиито функции са нормативно регламентирани.

На основание чл. 6, т. 4 от ЗКАИИП удостоверенията за съответната проектантска правоспособност се издават от Камарата на архитектите в България и Камарата на инженерите в инвестиционното проектиране, което води до извод, че преценката относно наличието на предпоставките за издаване на това удостоверение не принадлежи на министъра на регионалното развитие и благоустройството.

При преценката за материалната компетентност на министъра следва да се има предвид, че никъде в закона не се съдържат разписани правомощия за този административен орган, съобразно които същият да е овластен да се произнася не само по въпроси относно проектантска правоспособност, а така също и по отношение на всякакви други въпроси. В подкрепа на това наше разбиране е и съдържанието на акта, от което е видно, че в него не се съдържа правното основание, послужило за неговото издаване. В ЗКАИИП министърът дори не е посочен и като горестоящ административен орган на органите на двете камари, в който смисъл единственият правилен и законосъобразен извод е, че с издаването на оспорения от мен акт министърът е излязъл извън рамките на своята компетентност. В конкретния случай считам, че постановеният акт изцяло е лишен от законова опора, поради което е налице материална незаконосъобразност, тъй като пълната липса на материална компетентност прави административния акт нищожен.

Освен липсата на материална компетентност, водеща до нищожност на акта, считам, че при неговото издаване е нарушено и друго изискване за неговата законосъобразност - не е спазена неговата форма, представляващо основание за отмяна по чл. 146, т. 2 от АПК.

По своя правен характер писмото на министъра представлява общ административен акт, като в този смисъл по отношение на него важат изискванията за форма на акта, съдържащи се в чл. 59, ал. 2 от АПК. Разпоредбата на чл. 59, ал. 2 от АПК регламентира изискванията към формата на акта, които следва да бъдат спазени при издаването на акта. Изискването за издаване на акта е същият да съдържа фактически и правни основания за издаването му, като в настоящия случай твърдим, че същото е съществено нарушено. Само при наличието на фактическо и правно основание в акта адресатът може да упражни ефективно своята защита, а съответно съда да извърши преценка за законосъобразността на акта. С оглед на това значение съдебната практика трайно приема, че липсата на фактически и правни основания е основание за отмяна на акта.

Наред с горното в настоящия случай според мен съществено са нарушени и административнопроизводствените правила при издаване на акта. Не ми е известно обстоятелството по какъв начин е положено началото на административното производство, но според разпоредбата на чл. 66, ал. 1 от АПК откриването на производството по издаване на общия административен акт се оповестява публично чрез средствата за масово осведомяване, чрез изпращането на проекта до организации на заинтересованите лица или по друг подходящ начин, а съгласно чл. 66, ал. 2 от с. к. уведомяването включва и основните съображения за издаване на акта. В настоящия случай считам, че не е изпълнен фактическият състав на посочените разпоредби, което е

довело и до лишаване от възможност на заинтересованите лица да вземат участие при издаване на акта.

Макар и описаните по-горе пороци, допуснати при издаване на акта, да обосновават предпоставките за прогласяването му като нищожен, респ. отмяната му като незаконосъобразен, следва да се посочи и друг съществен порок на акта, допуснат при неговото издаване, а именно нарушението на материалния закон.

Предварителното уточнение, което следва да направим е, че на ниво право на Европейския съюз професията „инженер в инвестиционното проектиране“ не е регламентирана като регулирана професия, поради което посоченото обстоятелство неправилно се интерпретира в акта на министъра.

Директива 2005/36/ЕО на Европейския парламент и на Съвета от 7 септември 2005 година относно признаването на професионалните квалификации установява реда за признаване от държава членка, която обвързва достъпа до регламентирана професия или нейното упражняване на своята територия с притежаване на определени професионални квалификации, с цел достъп до тази професия и нейното упражняване, на професионални квалификации, придобити в една или повече други държави членки, и който ред позволява на титуляра на споменатите в директивата квалификации да упражнява там същата професия.

В член 46 от Директива 2005/36/ЕО се съдържат изискванията към обучението на архитектите, като наред с разширеното описание на тези изисквания, в разпоредбата е посочено, че обучението по архитектура осигурява придобиването на знания, умения и компетентности за създаване на идейни архитектурни проекти, които отговарят едновременно на естетическите и техническите изисквания, а така също и придобиването на подходящи знания за физическите проблеми и технологии и за функцията на сградите, така че да се изградят вътрешните условия за удобство и климатична защита, в рамките на устойчивото развитие. В директивата е посочено, че за целите на достъпа и упражняването на професионална дейност като архитект всяка държава-членка признава на удостоверенията за професионални квалификации същите последици на своята територия, каквито имат удостоверенията за професионални квалификации, които самата тя издава. В този контекст следва да се приеме, че архитект, упражняващ правата на територията на една държава членка, упражнявайки професионалната дейност на територията на страната ни, ще бъде ограничен относно обхвата на своята дейност. Системата за автоматично признаване на професионалните квалификации, предвидена по отношение на професията „архитект“ в членове 21, 46 и 49 от Директива 2005/36, не дава никакво право на преценка на държавите членки. Забраната да се поставят каквито и да било допълнителни изисквания важи в пълна степен за Директива 2005/36, доколкото тя засилва автоматичния характер на признаването на удостоверенията за професионална квалификация за професията „архитект“ по отношение на Директива 85/384 относно взаимното признаване на дипломи, удостоверения и други официални документи за професионална квалификация в областта на архитектурата и за мерките за улесняване на ефективното упражняване на правото на установяване и свободното предоставяне на услуги. По отношение на целта на Директива 2005/36 от членове 1 и 4 става ясно, че основният предмет на взаимното признаване е да осигури на притежателите на професионална квалификация достъпа в приемащата държава членка до регламентираната професия, за която са придобили квалификация в държавата членка по произход, и

възможността да упражняват тази професия в приемащата държава членка при същите условия, които се прилагат за нейните граждани (решение *Ordre des architectes*, C-365/13, EU:C: 2014:280, т. 19).

Със Закона за признаване на професионални квалификации наред с регламентацията на условията и реда за признаване на професионални квалификации, придобити в други държави членки и в трети държави, с цел достъп и упражняване на регулирани професии в Република България, условията и реда за частичен достъп до упражняване на регулирана професия и признаването на стаж за усвояване на професията в друга държава членка, са уредени и правилата за извършване на оценка на пропорционалността преди въвеждането на нови или изменението на действащи разпоредби в закон, подзакон или административен акт, с които се ограничава упражняването или достъпът до регулирани професии. В § 22 от Допълнителната разпоредба към Закона за изменение и допълнение на Закона за признаване на професионални квалификации (ДВ, бр. 21 от 2021 г.) е посочено, че законът въвежда изискванията на Директива (ЕС) 2018/958 на Европейския парламент и на Съвета от 28 юни 2018 г. относно проверката за пропорционалност преди приемането на ново регулиране на професии (ОВ, L 173/25 от 9 юли 2018 г.) и изисквания на Директива 2005/36/ЕО на Европейския парламент и на Съвета от 7 септември 2005 г. относно признаването на професионалните квалификации. От съображение трето на преамбюла на Директива (ЕС) 2018/958 се установява, че принципът на пропорционалност като един от основните принципи на правото на ЕС следва да гарантира, че националните мерки с цел да не направят или да не възпрепятстват упражняването на основни свободи, гарантирани от ДФЕС, следва да изпълняват четири условия: да бъдат прилагани по недискриминационен начин, да бъдат обосновани от цели от обществен интерес, да бъдат подходящи за постигане на преследваната с тях цел и да не надхвърлят необходимото за постигането ѝ.

Предвид невъзможността, създадена от акта на министъра, за упражняването на регулирана професия, за административния орган е било задължително да изследва наличието на кумулативните предпоставки, съдържащи се в Директива (ЕС) 2018/958, което е безспорно, че не е сторено. Една от четирите свободи, представляващи основата на функционирането на вътрешния пазар на Европейския съюз е свободата на предоставяне на услуги, която с направеното от министъра неправилно тълкуване на приложими разпоредби съществено е накърнена.

При анализ на разпоредбите на Директива 2005/36 се установява, че всеки архитект носи отговорност както за естетическите, така и за техническите изисквания на строежите, доколкото и обучението във връзка с проектирането на сгради дава достатъчно знания за физическите проблеми и технологии, така че архитектът да изгради балансирана вътрешна среда за удобство и климатична защита. Изискванията на директивата по отношение на обучението на архитектите са транспонирани и в приетата с ПМС № 11 от 21.01.2016 г. (обн., ДВ, бр. 7 от 26.01.2016 г.) Наредба за държавните изисквания за придобиване на висше образование по специалността „Архитектура“ на образователно-квалификационна степен "магистър" с професионална квалификация „архитект“. Съдържанието на обучението за придобиване на висше образование по специалността "Архитектура" на образователно-квалификационната степен "магистър" осигурява знания, практически умения и компетентности, необходими за получаване на професионална квалификация

"архитект", в това число и такива, свързани със знания за физическите и технологичните проблеми, както и знания за функцията на сградите и съоръженията – с оглед осигуряването им с вътрешни условия за комфорт на обитаване, здравословна жизнена среда и за защита от климатични въздействия и др.

Безспорно е, че т. нар. от министъра проектна част „Енергийна ефективност“ изследва основно топлотехническите характеристики на сградите, с цел климатична защита на използваните за това видове топлоизолационни материали и топлоизолационни системи, което води до единствения законосъобразен извод, че всеки архитект, съобразно придобитите от него знания, умения и компетентности може да докаже енергийната ефективност на проектираната от него сграда.

Съгласно разпоредбата на чл. 139, ал. 2 от ЗУТ възложителят в зависимост от спецификата на обекта задължително възлага за изработване тези части на инвестиционния проект, въз основа на които може да се направи оценка за съответствие с изискванията на чл. 169, ал. 1 и 3 и да се изпълни строежът. Според нормата на чл. 169, ал. 1, т. 6 от ЗУТ строежите се проектират, изпълняват и поддържат в съответствие с основните изисквания към строежите, определени в приложение I на Регламент (ЕС) № 305/2011 на Европейския парламент и на Съвета от 9 март 2011 г. за определяне на хармонизирани условия за предлагането на пазара на строителни продукти и за отмяна на Директива 89/106/ЕИО на Съвета (ОВ, L 88/5 от 4 април 2011 за икономия на енергия и топлосъхранение.

При тълкуването на разпоредбата на чл. 169, ал. 1, т. 6 от ЗУТ в контекста на Регламент (ЕС) № 305/2011 на ЕП, ведно с Приложение I, определящо основните изисквания към строежите, се установява, че европейското законодателство, което предвижда строежите като цяло и като отделни части да са годни за предвидената за тях употреба, съдържа норми за икономия на енергия и топлосъхранение, като според т. 6 от приложението строежите, както и техните отоплителни, охладителни, осветителни и вентилационни инсталации трябва да са проектирани и изпълнени по такъв начин, че необходимото при експлоатацията количество енергия да бъде малко, като се отчитат обитателите и местните климатични условия. В т. 6 се съдържа и изискване строежите да бъдат също така енергийно ефективни, като консумират възможно най-малко енергия по време на тяхното изграждане и демонтиране.

При позоваването на посочения Регламент, ведно с приложението към него, е необходимо да се има предвид, че предметният им обхват, определен с разпоредбата на член 1 от Регламента, е относим към строителни материали и компоненти. Според съдържанието на член 1 регламентът определя условията за пускането или предоставянето на пазара на строителни продукти посредством установяването на хармонизирани правила за това как да се изразят експлоатационните показатели на строителните продукти по отношение на техните съществени характеристики и за използването на маркировката „СЕ“ за тези продукти.

Безспорно е за нас обстоятелството, че избора на строителни продукти и системи е основна дейност от професионалния профил на архитекта, на база неговото мултидисциплинарно образование. По същество с акта на министъра се създава задължение, с което архитектите се подлагат на изискване за преминаване на съответен курс на обучение и съответно на специална проверка за техните професионални способности, което е недопустимо.

ри тълкуване на използваното в процесния акт понятие част „Енергийна ефективност“ е необходимо да се изясни неговото съдържание. В българското законодателство понятието „енергийна ефективност в сгради“ се съдържа в разпоредбата на § 1, т. 4 от Допълнителните разпоредби на Закона за енергийната ефективност и според посочената разпоредба това е осигуряването и поддържането на нормативните параметри на микроклимата в сградите, топлосъхранението им и икономията на енергийни ресурси за нуждите на сградите с минимални финансови разходи. По смисъла на § 1, т. 5 от ДР на закона енергийната ефективност представлява съотношението между изходното количество произведена стока, услуга или енергия и вложеното количество енергия. Законът цели повишаване на енергийната ефективност като основен фактор за повишаване конкурентноспособността на икономиката, сигурността на енергийните доставки и опазването на околната среда чрез използване на система от дейности и мерки за повишаване на енергийната ефективност при крайните потребители. Всъщност, последното представлява и дейност за реализиране на знанията, уменията и компетентностите на архитектите.

С издаването на акта министърът превратно е упражнил своите права, което е довело и до несъответствие с целта на закона.

Анализът на европейската и на вътрешната ни правна уредба формира извода, че мерките от вида на съдържащите се в акта на министъра имат две цели – от една страна – ограничаване на правата на архитектите, а от друга – предоставяне на права на определена категория правни субекти, т. нар. „инженери в инвестиционното проектиране“ по част „Енергийна ефективност“ и част „Пожарна безопасност“, което неминуемо рефлектира и върху потребителите на проектантски услуги, предвид необоснованото увеличаване на разходите при изработване на инвестиционния проект. Целта на акта, която е насочена към осуетяване на възможността за изработване от архитектите на посочените части „Енергийна ефективност“ и „Пожарна безопасност“, води до извод, че административният орган не е упражнил правомощията си по разумен начин, добросъвестно и справедливо.

По изложените съображения, моля да прогласите нищожността на акта на министъра на регионалното развитие и благоустройството, алтернативно – да го отмените като незаконосъобразен, за което допълнителни съображения ще изложя след запознаване с административната преписка.

Моля, да ми присъдите сторените в производството разноски.

Представям и моля да приемете удостоверение за пълна проектантска правоспособност и получен в РК – Варна имейл от председателя на КАБ, като в зависимост от становището на ответника ще ангажираме други доказателства и ще направим съответните доказателствени искания.

Приложение: съгласно текста, ведно с адвокатско пълномощно и документ за платена от трето лице в полза на жалбоподателя държавна такса, както и препис от жалбата и от писмените доказателства за ответната страна.

С уважение:


/адв. Силвия Русанова/

Приложение 7. Становище на ДНСК до АССГ.

към

Дисертация за присъждане на
образователна и научна степен “доктор“

на тема

**Организационни модели за създаване на сгради в България и Бавария
и формиране на професионалните задачи на архитектите**



РЕПУБЛИКА БЪЛГАРИЯ

Министерство на регионалното развитие и благоустройството
Дирекция за национален строителен контрол

Изх. № РК-229-01-080
15.04.2024

ДО

АДМИНИСТРАТИВЕН СЪД-СОФИЯ ГРАД

КОПИЕ

АРХ. ДИПЛ. ИНЖ. СТОЯН ГЕОРГИЕВ ИВАНОВ

С ПОСТОЯНЕН АДРЕС: ФЕДЕРАЛНА РЕПУБЛИКА ГЕРМАНИЯ

ГР. МЮНХЕН, УЛ. „ХОХБРЮКЕНЩРАСЕ“ 3,

80331 МЮНХЕН

e-mail: stoyan.todorov@st-architekten.com

АДВОКАТ СИЛВИЯ РУСАНОВА

С АДРЕС ЗА КОРЕСПОНДЕНЦИЯ,

ПОСОЧЕН ОТ ЖАЛБОПОДАТЕЛЯ:

ГР. СОФИЯ, БУЛ. „ВИТОША“ № 10, ЕТ. 2, АП. 8

*Относно: Постъпила жалба в Дирекция за национален строителен контрол /ДНСК/ с
вх. РК-229-04-378 / 08.04.2024 година от Стоян Георгиев Тодоров*

УВАЖАЕМИ/А ГОСПОДИН/ГОСПОЖО ПРЕДСЕДАТЕЛ,

Приложено изпращаме Ви по компетентност, постъпила в ДНСК и заведена с вх. №РК-229-04-378/08.04.2024 г., жалба от арх. дипл. инж. Стоян Георгиев Тодоров срещу Писмо изх. № РК-229-00-851/14.02.2024 г. на началника на ДНСК, с който е отказано издаване на удостоверение за оценяване на съответствието на инвестиционните проекти и/или упражняване на строителен надзор от физическо лице.

Приложения от жалбоподателя:

1. Жалба с вх. № РК-229-04-378/08.04.2024 г. в оригинал с придружена вносна бележка /копие/ за внесена сметка на АССГ за държавна такса в размер на 10 /десет/ лева.
2. На следващо място е посочено, че се изпраща и препис от подадената жалба за административния орган, като следва да се отбележи, че препис от жалбата не е представен, така както е посочено в самата жалба;

Приложения от ДНСК:

1. Писмо изх. № РК-229-00-851 / 14.02.2024 г.-заверено копие;
2. Писмо вх. № РК-229-01-350/31.01.2024 г.-копие;
3. Приложено заявление от 28.01.2024 г.-копие;
4. Списък по чл. 10, ал. 3 от Наредба № РД-02-20-25 от 03.12.2012 г. за условията и реда за издаване на удостоверение за вписване в регистъра на консултантите за оценяване на съответствието на инвестиционните проекти и/или упражняване на строителен надзор-копие;
5. Потвърждение съгласно Директива 2005/36/ЕО относно признаването на професионални квалификации-копие;
6. Писмо от баварската камара на архитектите-копие;
7. Декларация от Баварската камара на архитектите, включваща удостоверение, придружаващо свидетелство за квалификации-копие;
8. Закон за Баварската камара на архитектите и за Баварската камара на инженерите в сектор „Строителство“, Закон за камарите в строителството/ЗКС/-копие;
9. Баварски строителен закон /БСЗ/-копие;
10. Регламент за изпълнение на разпоредбите относно енергийния сектор /РИРИС/-копие;
11. Проект на договор № 1 от 29.01.2024 г., Приложение № 2-копие;
12. Пълномощно-заверено копие.

Становище на ДНСК по подадената жалба :

Считаме, че подадената жалба от арх. дипл. инж. Стоян Георгиев Тодоров с постоянен адрес: Федерална Република Германия, гр. Мюнхен, ул. „Хохбрюкенщрасе“ 3, с адрес за кореспонденция: гр. София, бул. „Витоша“ № 10, ет. 2, ап. 8-адвокат Силвия Русанова, срещу писмо на Началника на ДНСК с изх. № РК-229-00-851/14.02.2024 г. е неоснователна и недоказана.

В Централно управление на ДНСК е постъпило Ваше заявление с вх. № РК- 229-01-350/31.01.2024г., с което в качеството Ви на гражданин на Федерална република Германия и

на архитект, предоставящ от 2009г. проектантски услуги в Германия в областта на архитектурата и на инвестиционното проектиране и осъществяващ дейности като консултант идентични с тези по чл. 166, ал. 1, т. 1 от Закон за устройство на територията /ЗУТ/ и желаещ да бъде вписан в регистъра на ДНСК като физическо лице, предоставящо консултантски услуги по смисъла на чл. 166, ал. 1, т. от ЗУТ, Ви уведомяваме:

Съгласно действащият ЗУТ в Република България, дейностите по оценяване на съответствието на инвестиционните проекти и/или упражняване на строителен надзор се осъществява от юридически лица и/или еднолични търговци, които са търговци по смисъла на Търговския закон или са регистрирани в търговски или друг публичен регистър в държава - членка на Европейския съюз, или в друга държава - страна по Споразумението за Европейското икономическо пространство и разполагат с екип от специалисти, наети по трудов или друг договор с доказан професионален опит и технически компетентности, необходими за осъществяване на дейностите за оценяване на съответствието на инвестиционните проекти и/или упражняване на строителен надзор.

С чл. 166, ал. 7 от ЗУТ е дадена възможността дейностите като консултант да могат да се извършват и от чуждестранни юридически лица, представили копие от документ, удостоверяващ правото да извършват такава дейност, издаден от компетентен орган на държава - членка на Европейския съюз, или на друга държава – страна по споразумението за европейското икономическо пространство.

Съгласно Наредба № РД-02-20-25/2012 г., Глава втора. Изисквания и критерии за вписване в регистъра на консултанта, кандидатстващ за издаване на удостоверение, се дава възможност на лицата по чл. 166, ал. 7 от ЗУТ да кандидатстват по чл. 9 или по чл. 10 от същата наредба. Юридическите лица, кандидатстващи по чл. 9 от Наредбата, е необходимо да удостоверят с документи правото им да извършват дейностите по чл. 166, ал. 1, т. 1 от ЗУТ., издаден от компетентен орган на държава-членка на Европейския съюз или на друга държава-страна по Споразумението за Европейското икономическо пространство. С чл. 10 от Наредбата се дава възможност на юридическите лица, които имат право да упражняват дейностите си за оценяване на съответствието на инвестиционните проекти и/или упражняване на строителен надзор по законодателството на друга държава-членка на Европейския съюз, или друга държава-страна по Споразумението за Европейското икономическо пространство, в които няма режим, еквивалентен на режима по чл. 166, ал. 1, т. 1 от ЗУТ, да могат да упражняват тези дейности на територията на Република България временно и еднократно за определен строеж въз основа на удостоверение, издадено от началника на ДНСК или оправомощено от него длъжностно лице.

В тази връзка следва да се отбележи, че липсват законови основания за издаване на удостоверение за оценяване на съответствието на инвестиционните проекти и/или упражняване на строителен надзор от физическо лице.

По изложените по-горе подробни поводи и съображения, моля да постановите решение, с което да отхвърлите жалбата от арх. дипл. инж. Стоян Георгиев Тодоров с постоянен адрес: Федерална Република Германия, гр. Мюнхен, ул. „Хохбрюкенщрасе“ 3 срещу Писмо изх. № РК-229-00-851/14.02.2024 г. на началника на ДНСК, с което е отказано издаване на удостоверение за оценяване на съответствието на инвестиционните проекти и/или упражняване на строителен надзор от физическо лице.

В изпълнение на задълженията си по чл. 152, ал. ал. 3 от АПК прилагаме списък на страните, както следва:

1. **Жалбоподател:** арх. дипл. инж. Стоян Георгиев Тодоров с постоянен адрес: Федерална Република Германия, гр. Мюнхен, ул. „Хохбрюкенщрасе“ 3, с адрес за кореспонденция: гр. София, бул. „Витоша“ № 10, ет. 2, ап. 8-адвокат Силвия Русанова;
2. **Отвeтник:** Началник на Дирекция за национален строителен контрол със седалище и адрес на управление: гр. София, бул. „Христо Ботев“ № 47.

С уважение,

.....

Анна Сибинова

Процесуален представител на ДНСК



Приложение 8. Решение на ВАС по дело срещу ДНСК.

към

Дисертация за присъждане на
образователна и научна степен “доктор“

на тема

**Организационни модели за създаване на сгради в България и Бавария
и формиране на професионалните задачи на архитектите**

РЕШЕНИЕ
№ 2191
София, 05.03.2025 г.
В ИМЕТО НА НАРОДА

Върховният административен съд на Република България - Второ отделение, в съдебно заседание на деветнадесети февруари две хиляди двадесет и пета година в състав:

ПРЕДСЕДАТЕЛ: ИЛИЯНА ДОЙЧЕВА

ЧЛЕНОВЕ: СЛАВИНА ВЛАДОВА
СВЕТЛОЗАР РАЧЕВ

при секретар Илияна Иванова
на прокурора Момчил Таралански
от съдията Славина Влагова
по административно дело № 12127/2024 г.

и с участието
изслуша докладваното

Производство по реда на чл. 208 и следващите от Административно процесуалния кодекс (АПК) във връзка със Закона за устройство на територията (ЗУТ).

Образувано е по касационна жалба на С. Г. Т. с постоянен адрес гр. Мюнхен, чрез процесуалния му представител адв. Русанова, против решение № 22882 от 08.11.2024г., постановено по адм.д. № 4055/2024г. по описа на Административен съд София – град /АССГ/, с което е отхвърлена жалбата му против писмо изх. № РК – 229 – 00 – 851 от 14.02.2024г. /неправилно в оспореното решение посочено с № РК – 220 – 00 – 851/ на началника на Дирекция за национален строителен контрол /ДНСК/, с което на основание чл. 166 ал. 7 във връзка с член 166 ал. 1 т. 1 ЗУТ е отказано издаването на удостоверение за оценяване на съответствието на инвестиционните проекти и/или упражняване на строителен надзор от физическо лице на жалбоподателя. С решението жалбоподателят е осъден да заплати разноските по делото.

В касационната жалба се твърди неправилност на оспореното съдебно решение, като постановено при неправилното приложение на закона, при допуснати съществени нарушения на съдопроизводствените правила и поради необоснованост – отменителни основания по чл. 209 т. 3 АПК. Навежда съображения, че съдът не се е произнесъл по всички наведени в жалбата възражения, не е обсъдил събраните в хода доказателства и не е изложил съответни на тях мотиви, като посочва, че съдържащите се такива са бланкетни и без да достигне до собствени изводи. Излага съображения, че е изпълнил изискванията на чл. 166 ал. 7 във връзка с чл. 166 ал. 1 т. 1 ЗУТ да бъде вписан в съответния регистър и да му бъде издадено удостоверение да извършва съответни дейности в инвестиционното проектиране като физическо лице, тъй като е вписан в регистъра на архитектите на Федерална република Германия от 2009г. и предоставя такива услуги. Навежда съображения, че е незаконосъобразен извод на съда, че съгласно нормативната уредба тези дейности могат да се осъществяват само от юридическо лице или едноличен търговец и разполагат с екип от специалисти, наети по трудов или друг договор с доказан професионален опит, съгласно Наредба № РД – 02 – 20 – 25 от 2012г. Посочва, че именно възможността на предоставя такива услуги като физическо лице и в друга държава са израз на свободното предоставяне на услуги. Поради това твърди, че оспореният акт е издаден в противоречи както с българското право, така и с европейското, противно на изводите на първоинстанционния съд. Иска оспореното решение да бъде отменено и вместо него да бъде постановено друго по съществото на спора, с което се отменя оспореният административен акт Претендира разноски.

Ответникът Началникът на ДНСК, чрез процесуалния си представител юрк. Сибинова, изразява становище за неоснователност на касационната жалба. Претендира разноси в размер на юрисконсултско възнаграждение.

Представителят на Върховна касационна прокуратура дава мотивирано заключение за основателност на касационната жалба и неправилност на оспореното съдебно решение.

Касационната жалба е подадена в срока по чл. 211 ал. 1 АПК, от надлежна страна, адресат на оспореното решение и срещу съдебен акт, подлежащ на касационен контрол, поради което е процесуално допустима.

Разгледана по същество е неоснователна.

За да постанови оспореното решение първоинстанционният съд е приел, че оспореният отказ е издаден от компетентен орган в рамките на предоставените му правомощия, при спазване на изискванията за форма и без допуснати съществени нарушения на административно производствените правила. Достигнал е до извод и за материална законосъобразност на постановеният отказ, тъй като по действащия ЗУТ дейностите по оценяване на съответствието на инвестиционните проекти и/или упражняването на строителен надзор се осъществява само от юридически лица и/или еднолични търговци, т.е. субекти, които са търговци по съответния Търговски закон или са регистрирани в търговски или друг публичен регистър на държава – членка на Европейския съюз /ЕС/ или в друга държава, страна по Споразумението за Европейско икономическо пространство и разполага с екип от специалисти, наети по трудов или друг вид договор, с доказан професионален опит и технически компетентности, необходими за осъществяване на дейностите по оценяване на съответствието на инвестиционните проекти и/или упражняването на строителен надзор. Приел е, че съгласно чл. 166 ал. 7 ЗУТ е дадена възможност и на чуждестранни юридически лица, представили съответен документ, да осъществяват тези дейности като консултант, като съгласно Наредба № РД – 02 – 20 – 25/2012г. критериите за вписване в регистъра на лицата по чл. 166 ал. 7 ЗУТ и за издаване на удостоверение за извършване на такава дейност, са също кандидатстващия да е юридическо лице, включително в случаите на еднократно изпълнение на такива действия по отношение на конкретен строеж. Поради това е приел, че не са били налице предпоставките за издаване на исканото от жалбоподателя удостоверение за право да извършва оценяване на съответствието на инвестиционни проекти и/или да упражнява строителен надзор като физическо лице, съответно направил извод за законосъобразност на постановения отказ и е отхвърлил оспорването.

Решението е правилно.

Съгласно чл. 166 ал. 1 т. 1 ЗУТ „Консултантът въз основа на писмен договор с възложителя: 1. извършва оценяване на съответствието на инвестиционните проекти и/или упражнява строителен надзор“, а ал. 7 на посочената разпоредба предвижда, че „Дейностите като консултант по ал. 1 могат да се извършват и от лица, представили копие от документ, удостоверяващ правото да извършват такава дейност, издаден от компетентен орган на държава - членка на Европейския съюз, или на друга държава - страна по Споразумението за Европейското икономическо пространство“. С чл. 166 ал. 7 е допуснато без ограничения и лице, което представи документ, удостоверяващ правото да извършва съответните технически дейности, а именно оценяване на съответствието на инвестиционните проекти и/или да упражнява строителен контрол, който е издаден от държава членка на ЕС или на ЕИП, да извършва дейността на консултант на територията на РБългария. Това е общата разпоредба допускаща лица, които са получили съответна правоспособност от държава членка на ЕС да извършват

съответните дейности, като обаче тези лица на общо основание следва да отговарят на критериите, заложи в българския закон, относно предпоставките за издаване то на удостоверение. Дейността по чл. 166, ал. 1, т. 1 ЗУТ е със значим обществен интерес, като от нея може да зависи и живота и здравето на хората, и за да гарантира качествено й изпълнение, законодателят е поставил определени изисквания и към самите физически лица, чрез които може да се извършва дейността – чл. 167, ал. 1, т. 2 ЗУТ и към консултанта, който следва да е юридическите лица, които следва да я извършват по чл. 167 ал. 1 ЗУТ, като разумът на това изискване е консултантът да може да осигури всеотранната оценка на съответствието на инвестиционните проекти, които включват различни части, изискващи компетентност в различни технически области и специалности. На посочените изисквания, които са разписани в чл. 167 ал. 1 ЗУТ, при наличието на които може да бъде издадено удостоверение за консултант, жалбоподателят безспорно не отговаря на тях, поради и което е правилен извода на съда, че в тази връзка постановеният отказ е законосъобразен..

Неоснователно е възражението в касационната жалба, че оспореният отказ е в противоречие с европейското право и в частност с чл. 56 от Договора за функциониране на Европейския съюз /ДФЕС/, който забранява ограничения на свободното предоставяне на услуги в рамките на Съюза. Това е така, тъй като чл. 57 ал. 2 от ДФЕС предвижда, че „Без да се засягат разпоредбите на главата относно правото на установяване, лицата които предлагат услуги имат право, за да ги осъществяват, временно да упражняват своята дейност в държавата – членка, където се предоставя услугата, при същите условия, каквито тази държава прилага спрямо своите собствени граждани“. Предвид цитираното осигуреното от ДФЕС равнопоставено третиране при свободното предоставяне на услуги не означава абсолютно свободното предоставяне на такива, особено които касаят регулирана професия, като инвестиционното проектиране. В този смисъл ДФЕС изисква с оглед равнопоставеността предвидените в законодателството условия за предоставянето на съответната услуга да са еднакви както спрямо собствените на държавата граждани, така и спрямо гражданите на държавата – членка на ЕС или на такива на държава, която е страна по Споразумението на ЕИП. В случая относимите към спора разпоредби на чл. 166 ал. 7 във връзка с чл. 166 ал. 1 т. 1 ЗУТ, както и на чл. 167 ал. 1 ЗУТ предвиждат еднакви условия относно изискванията за издаването на удостоверение за извършва оценяване на съответствието на инвестиционните проекти и/или упражнява строителен надзор, като в чл. 167 ал. 1 ЗУТ дори по отношение на лицата на държава – членка на ЕС или на такава по споразумението за ЕИП е допуснато разширение на изискването същите да бъдат вписване в търговски или друг публичен регистър на съответната държава – членка или такава по споразумението.

По изложените съображения оспореното съдебно решение като правилно следва да бъде оставено в сила.

Предвид изхода на спора е неоснователно искането на касатора за присъждане на разноски, а искането на ответника по касация като основателно съгласно чл. 143 ал. 3 АПК следва да бъде уважено в размер на 100 лева на основание чл. 37 от Закона за правната помощ във връзка с чл. 24 от Наредбата за заплащане на правната помощ.

Воден от горното и на основание чл. 221, ал. 2 АПК, Върховният административен съд, състав на второ отделение,

РЕШИ:

ОСТАВЯ В СИЛА решение № 22882 от 08.11.2024г., постановено по адм.д. № 4055/2024г. по описа на Административен съд София – град.

ОСЪЖДА С. Г. Т. да заплати на Дирекция за национален строителен контрол сума в размер на 100 /сто/ лева, разноси за настоящата инстанция.

РЕШЕНИЕТО е окончателно.

Вярно с оригинала,

ПРЕДСЕДАТЕЛ: /п/ ИЛИЯНА ДОЙЧЕВА

секретар:

ЧЛЕНОВЕ: /п/ СЛАВИНА ВЛАДОВА

/п/ СВЕТОЗАР РАЧЕВ

Приложение 9. Фактори, влияещи върху вземането на решения от възложител в България относно оптималния модел на реализация на строително начинание.

Приложение 10. Фактори, влияещи върху вземането на решения от възложител в Бавария относно оптималния модел на реализация на строително начинание.

Приложение 11. Логическа верига на работния процес в България.

Приложение 12. Логическа верига на работния процес в Бавария.

Приложение 13. Работен алгоритъм в България.

Приложение 14. Работен алгоритъм в Бавария.

към

Дисертация за присъждане на
образователна и научна степен “доктор“

на тема

**Организационни модели за създаване на сгради в България и Бавария
и формиране на професионалните задачи на архитектите**

Приложение 15. Псевдокод на уеб-базиран софтуер, който ще води консултирането до едно оптимално организационно решение.

към

Дисертация за присъждане на
образователна и научна степен “доктор“

на тема

**Организационни модели за създаване на сгради в България и Бавария
и формиране на професионалните задачи на архитектите**

Псевдо код

branches["decision_tree"] = обект с question: "Какъв е Вашият правен статут?", options: unregistered и registered

branches["public_pressure_branch"] = обект с question: "Упражнява ли се обществен натиск по отношение на начина на изпълнение на строителното начинание?", options: yes и no

branches["special_circumstances_branch"] = обект с question: "Спада ли обектът към някое от следните обстоятелства: обекти, спадащи към чл. 147 от ЗУТ и/или изключение съгласно чл. 14 (2) от Закона за Камара на строителите в България?", options: yes и no

branches["rare_knowhow_branch"] = обект с question: "Предвижда ли се използване на рядко и специфично ноу-хау за проектиране и строителство?", options: yes и no

branches["contest_branch"] = обект с question: "Желаете ли да организирате конкурс за проекта?", options: yes и no

branches["quality_control_interest_branch"] = обект с question: "Имате ли интерес да ангажирате отделни изпълнители с цел контрол на качеството?", options: yes и no

branches["investor_control_branch"] = обект с question: "Готови ли сте да заплатите на доверено лице за инвеститорски контрол върху осъществяването на строителното начинание?", options: yes и no

branches["equipment_suppliers_branch"] = обект с question: "Искате ли да ангажирате доставчици на машини, съоръжения и технологично оборудване?", options: yes и no

branches["trust_participants_branch"] = обект с question: "На какво се основава Вашето доверие към участниците в процеса?", options: capacity и other_motives

branches["trust_justice_branch"] = обект с question: "Имате ли доверие в правораздаването?", options: yes и no

branches["economic_interest_branch"] = обект с question: "Факторът икономически интерес от съществено значение ли е за Вас?", options: yes и no

branches["preferred_timeline_branch"] = обект с question: "Какъв е предпочитаният от Вас срок за реализация на обекта?", options: shortest и quality_adjusted

branches["management_capacity_branch"] = обект с question: "Искате ли да приложите свой капацитет по управление на строителното начинание?", options: yes и no

branches["lead_designer_branch"] = обект с question: "Искате ли да ангажирате проектант по водеща проектна част?", options: yes и no

Дефинирай функция process_branch(current_branch):

Нека node = branches[current_branch] ако current_branch е string, иначе node = current_branch

Покажи въпрос: node.question

Вземи user_choice от потребителя (ключ от node.options)

Нека option = node.options[user_choice]

Покажи description: option.description

Покажи модели: option.models (списък като ["BG-1", "BG-2", ...])

ако option има end_model

Покажи краен модел: option.end_model

Върни (спри процеса)

иначе

ако option.next е string (референция като "public_pressure_branch")

ако option.next == "same_as_unregistered_above"

process_branch(branches["decision_tree"].options.unregistered.next) // Специален

случай: копира unregistered next

иначе

process_branch(option.next)

иначе ако option.next е обект

process_branch(option.next) // Рекурсия за вложени next

// Детайлни дефиниции за options в всеки branch (1:1 от JSON)

// За decision_tree:

options.unregistered.description = "физическо или юридическо лице, невписано в регистъра на КСБ с лиценз, валиден и за конкретния обект и използващо тази регистрация за строителното начинание"

options.unregistered.models = ["BG-1", "BG-2", "BG-3", "BG-4"]

options.unregistered.next = обект с question: "Каква е основната цел на строителното начинание?", options: rental_sale и own_use

// За rental_sale:

rental_sale.description = "наем/продам"

rental_sale.models = ["BG-1", "BG-2", "BG-3", "BG-4"]

rental_sale.next = обект с question: "Кой е решаващият фактор при реализацията на тази цел?", options: market_situation и client_requirements

// market_situation:

market_situation.description = "текуща ситуация на пазара на имоти и тенденции"

market_situation.models = ["BG-1", "BG-2", "BG-3", "BG-4"]

market_situation.next = "public_pressure_branch"

// client_requirements:

client_requirements.description = "изисквания на потенциални наематели/купувачи"

client_requirements.models = ["BG-1", "BG-2", "BG-3", "BG-4"]

client_requirements.next = обект с question: "Какви са изискванията на потенциалните наематели/купувачи?", options: high и low

high.description = "високи"

high.models = ["BG-1", "BG-2", "BG-3"]

high.next = "public_pressure_branch"

low.description = "ниски"

low.models = ["BG-1", "BG-2", "BG-3", "BG-4"]

low.next = "public_pressure_branch"

// За own_use:

own_use.description = "собствен бизнес/лично потребление"

own_use.models = ["BG-1", "BG-2", "BG-3", "BG-4"]

own_use.next = обект с question: "Кой е решаващият фактор при реализацията на тази цел?", options: own_quality

own_quality.description = "собствени изисквания за качество"

own_quality.models = ["BG-1", "BG-2", "BG-3", "BG-4"]

own_quality.next = обект с question: "Какви са собствените Ви изисквания за качество?", options: high и high_investment

high.description = "високи"

high.models = ["BG-1", "BG-2"]

high.next = "public_pressure_branch"

high_investment.description = "високи (обектът е дългосрочна инвестиция)"

high_investment.models = ["BG-1", "BG-2", "BG-3"]

high_investment.next = "public_pressure_branch"

```
options.registered.description = "физическо или юридическо лице, вписано в регистъра на КСБ с лиценз, валиден и за конкретния обект и използващо тази регистрация за строителното начинание"
```

```
options.registered.models = ["BG-1", "BG-2", "BG-3", "BG-4"]
```

```
options.registered.next = "same_as_unregistered_above"
```

```
// За public_pressure_branch:
```

```
yes.description = "да"
```

```
yes.models = ["BG-1", "BG-2", "BG-3", "BG-4"]
```

```
yes.next = "special_circumstances_branch"
```

```
no.description = "не"
```

```
no.models = ["BG-1", "BG-2", "BG-3", "BG-4"]
```

```
no.next = "special_circumstances_branch"
```

```
// За special_circumstances_branch:
```

```
yes.description = "да"
```

```
yes.models = ["BG-1", "BG-2", "BG-3", "BG-4"]
```

```
yes.next = "rare_knowhow_branch"
```

```
no.description = "не"
```

```
no.models = ["BG-1", "BG-2", "BG-3", "BG-4"]
```

```
no.next = "rare_knowhow_branch"
```

```
// За rare_knowhow_branch:
```

```
yes.description = "да"
```

```
yes.models = ["BG-4"]
```

```
yes.end_model = "BG-4"
```

```
no.description = "не"
```

```
no.models = ["BG-1", "BG-2", "BG-3", "BG-4"]
```

```
no.next = "contest_branch"
```

```
// За contest_branch:
```

```
yes.description = "да"
```

```
yes.models = ["BG-3"]
```

```
yes.end_model = "BG-3"
```

```

no.description = "не"
no.models = ["BG-1", "BG-2", "BG-3", "BG-4"]
no.next = "quality_control_interest_branch"

// За quality_control_interest_branch:
yes.description = "да"
yes.models = ["BG-1", "BG-2", "BG-3", "BG-4"]
yes.next = обект с question: "Кой от двамата участници в процеса на строителство
предпочитате да ангажирате?", options: technical_manager и registered_builder
    technical_manager.description = "технически ръководител"
    technical_manager.models = ["BG-1", "BG-2", "BG-3"]
    technical_manager.next = "investor_control_branch"
    registered_builder.description = "строител, вписан в регистъра на КСБ"
    registered_builder.models = ["BG-1"]
    registered_builder.next = "investor_control_branch"
no.description = "не"
no.models = ["BG-1", "BG-2", "BG-3", "BG-4"]
no.next = "investor_control_branch"

// За investor_control_branch:
yes.description = "Готов съм."
yes.models = ["BG-2", "BG-3", "BG-4"]
yes.next = обект с question: "Какъв инвеститорски контрол бихте възложили?", options:
max_contractors, max_with_single_executor_project_build и max_with_full_single_executor
    max_contractors.description = "контрол върху възможно най-много контрагенти в
процеса"
    max_contractors.models = ["BG-2"]
    max_contractors.end_model = "BG-2"
    max_with_single_executor_project_build.description = "контрол върху възможно най-
много контрагенти в процеса, имайки предвид, че работен проект и строителство се
възлагат на един изпълнител"
    max_with_single_executor_project_build.models = ["BG-2", "BG-3"]
    max_with_single_executor_project_build.end_model = "BG-1"

```

```
max_with_full_single_executor.description = "контрол върху възможно най-много  
контрагенти в процеса, имайки предвид, че цялостно проектиране и строителство се  
възлагат на един изпълнител"
```

```
max_with_full_single_executor.models = ["BG-2", "BG-3"]
```

```
max_with_full_single_executor.end_model = "BG-1"
```

```
no.description = "Не съм готов и съм наясно, че липсата на такъв контрол може да се  
отрази на качеството и на стойността на обекта."
```

```
no.models = ["BG-1", "BG-2", "BG-3", "BG-4"]
```

```
no.next = "equipment_suppliers_branch"
```

```
// За equipment_suppliers_branch:
```

```
yes.description = "да"
```

```
yes.models = ["BG-2"]
```

```
yes.end_model = "BG-2"
```

```
no.description = "не"
```

```
no.models = ["BG-3"]
```

```
no.end_model = "BG-3"
```

```
// За trust_participants_branch:
```

```
capacity.description = "на техния капацитет (например: образование, експертиза, лиценз,  
опит)"
```

```
capacity.models = ["BG-1", "BG-2", "BG-3", "BG-4"]
```

```
capacity.next = "trust_justice_branch"
```

```
other_motives.description = "мотиви, несвързани с техния капацитет (например: ниска  
цена, кратък срок и т.н.)"
```

```
other_motives.models = ["BG-2", "BG-3", "BG-4"]
```

```
other_motives.next = "trust_justice_branch"
```

```
// За trust_justice_branch:
```

```
yes.description = "да"
```

```
yes.models = ["BG-1", "BG-2"]
```

```
yes.next = "economic_interest_branch"
```

```
no.description = "не"
```

```
no.models = ["BG-1", "BG-2", "BG-3", "BG-4"]
```

```

no.next = "economic_interest_branch"

// За economic_interest_branch:
yes.description = "да"
yes.models = ["BG-1", "BG-2"]
yes.next = "preferred_timeline_branch"
no.description = "не"
no.models = ["BG-1", "BG-2", "BG-3", "BG-4"]
no.next = "preferred_timeline_branch"

// За preferred_timeline_branch:
shortest.description = "възможно най-краткият"
shortest.models = ["BG-2"]
shortest.end_model = "BG-2"
quality_adjusted.description = "съобразен с изискуемото качество"
quality_adjusted.models = ["BG-3"]
quality_adjusted.end_model = "BG-3"

// За management_capacity_branch:
yes.description = "Да, бих приложил организационен и/или административен капацитет
и/или експертиза в строителството и/или опит от минали сравними проекти."
yes.models = ["BG-1", "BG-2", "BG-3"]
yes.next = "trust_participants_branch"
no.description = "Не притежавам/не искам да приложа такъв капацитет."
no.models = ["BG-4"]
no.end_model = "BG-4"

// За lead_designer_branch:
yes.description = "да"
yes.models = ["BG-1", "BG-2", "BG-3"]
yes.next = "quality_control_interest_branch"
no.description = "не"
no.models = ["BG-1"]
no.next = "quality_control_interest_branch"

```

```
// Стартирай процеса  
process_branch("decision_tree")
```

Приложение 16. Документация на β -версия на
уеб-базирано приложение ARCHModels.

към

Дисертация за присъждане на
образователна и научна степен “доктор“

на тема

**Организационни модели за създаване на сгради в България и Бавария
и формиране на професионалните задачи на архитектите**

Optima - Production Документация

****Версия на документа:**** 1.0
****Дата на създаване:**** 2026-02-02
****Проект:**** Optima v1.0.0
****URL:**** <https://models.iqvolution.ws>
****Лиценз:**** Proprietary

Съдържание

1. [Резюме на Проекта] (#1-резюме-на-проекта)
2. [Предпоставки и Зависимости] (#2-предпоставки-и-зависимости)
3. [Инструкции за Сглобяване и Разгръщане] (#3-инструкции-за-сглобяване-и-разгръщане)
4. [Конфигурация и Настройки] (#4-конфигурация-и-настройки)
5. [Наблюдаемост и Логове] (#5-наблюдаемост-и-логове)
6. [Поддръжка и Операции] (#6-поддръжка-и-операции)
7. [Сигурност] (#7-сигурност)
8. [Резервни Копия и Възстановяване] (#8-резервни-копия-и-възстановяване)
9. [Масшабиране] (#9-масшабиране)
10. [Приложения] (#10-приложения)

1. Резюме на Проекта

1.1. Цел

Optima е интерактивна уеб система за подбор на строителни модели. Приложението използва decision tree (дърво за вземане на решения), за да помогне на потребителите да определят кой от наличните строителни модели е най-подходящ за техните нужди.

****Решаван проблем:**** Автоматизиране на консултантския процес за избор на подходящ строителен модел чрез структуриран въпросник, който елиминира неподходящите варианти.

1.2. Основни функционалности

Функционалност	Описание
Двоен регионален режим	Поддържа модели за България (BG-1 до BG-5) и Бавария/Германия (DE-1 до DE-5)
Интерактивен въпросник	Wizard интерфейс с прогресивно елиминиране на неподходящи модели
Потребителска автентикация	Регистрация, вход, потвърждение на имейл, възстановяване на парола
История на анкети	Съхранение на потребителските отговори и резултати
PDF експорт	Генериране на PDF отчет за резултатите от анкетата
Decision trace	Проследяване на логиката за достигане до конкретен модел
Responsive дизайн	PWA-ready с TailwindCSS

1.3. Целева аудитория

	Аудитория		Роля	
	-----		-----	
	Крайни потребители Клиенти, търсещи подходящ строителен модел			
	Консултанти Специалисти, използващи системата за препоръки			
	Администратори Технически екип за поддръжка и актуализация на данните			

2. Предпоставки и Зависимости

2.1. Системни изисквания

	Ресурс		Минимум		Препоръчително	
	-----		-----		-----	
	CPU		1 vCPU		2+ vCPU	
	RAM		1 GB		2+ GB	
	Дисково пространство 2 GB 5+ GB (с логове)					
	OS Linux (CentOS 8+, Ubuntu 20.04+) CentOS 8 / AlmaLinux 8					

2.2. Софтуерни зависимости

Задължителен софтуер

	Софтуер		Версия		Предназначение	
	-----		-----		-----	
	PHP		8.2+		Runtime за Laravel	
	Composer		2.x		PHP dependency manager	
	Node.js		18+ LTS		Frontend build	
	NPM		9+		JavaScript package manager	
	MySQL		8.0+		Релационна база данни	
	Apache/Nginx		Latest		Web сървър	

PHP разширения (задължителни)

```

php-cli, php-fpm, php-mysql, php-mbstring, php-xml, php-curl,  
php-zip, php-gd, php-bcmath, php-json, php-tokenizer, php-fileinfo  
```

Опционален софтуер

	Софтуер		Версия		Предназначение	
	-----		-----		-----	
	Redis		6+		Кеширане и сесии (опционално)	
	Supervisor		Latest		Управление на queue workers	

2.3. Външни услуги

	Услуга		Доставчик		Предназначение		Критичност	
	-----		-----		-----		-----	
	SMTP		SuperHosting.bg		Изпращане на имейли		Средна	
	DNS		Доставчик на домейна		DNS резолюция		Висока	
	SSL сертификат Let's Encrypt или доставчик HTTPS криптиране							
	Висока							

Забележка: Няма външни платежни системи или API интеграции.

```

---

## 3. Инструкции за Сглобяване и Разгръщане

### 3.1. Сглобяване (Build)

#### Клониране на кода

```bash
cd /home/igfound
git clone <REPOSITORY_URL> models.igvolution.ws
cd models.igvolution.ws/backend
```

#### Инсталиране на PHP зависимости

```bash
Production (без dev зависимости)
composer install --no-dev --optimize-autoloader

Development
composer install
```

#### Инсталиране на JavaScript зависимости

```bash
npm ci
```

#### Build на frontend активи

```bash
Production build
npm run build

Development с hot reload
npm run dev
```

#### Генериране на APP_KEY (само за първоначална настройка)

```bash
php artisan key:generate
```

### 3.2. Конфигурация на Production

#### Environment файл

Копирайте и конфигурирайте `.env` файла:

```bash
cp .env.example .env
nano .env
```

#### Критични environment променливи

```

```

```ini
Приложение
APP_NAME="Optima"
APP_ENV=production
APP_KEY=base64:ГЕНЕРИРАН_КЛЮЧ
APP_DEBUG=false # КРИТИЧНО: Винаги false в
production!
APP_URL=https://models.iqvolution.ws

База данни
DB_CONNECTION=mysql
DB_HOST=localhost
DB_PORT=3306
DB_DATABASE=iqfound_arch_models
DB_USERNAME=<DATABASE_USER>
DB_PASSWORD=<SECURE_PASSWORD>

Сесии
SESSION_DRIVER=database
SESSION_LIFETIME=120

Кеширане
CACHE_STORE=file # Или 'redis' за production

Опашки
QUEUE_CONNECTION=sync # Или 'redis/database' за async

Логове
LOG_CHANNEL=stack
LOG_LEVEL=error # warning или error за production

Имейл
MAIL_MAILER=smtp
MAIL_HOST=<SMTP_HOST>
MAIL_PORT=587 # STARTTLS
MAIL_USERNAME=<SMTP_USER>
MAIL_PASSWORD=<SMTP_PASSWORD>
MAIL_ENCRYPTION=tls
MAIL_FROM_ADDRESS=noreply@example.com
MAIL_FROM_NAME="${APP_NAME}"
```

#### Управление на секрети

**Текущо състояние:** Секретите се съхраняват в `.env` файл.

**Препоръки за production:**
- Не commit-вайте `.env` файл в хранилището
- Използвайте `.env.example` само с placeholder стойности
- Обмислете HashiCorp Vault или cloud secret manager за критични среди

### 3.3. Разгръщане (Deployment)

#### Предварителна проверка

```bash
Проверка на PHP версията
php -v # Трябва да е 8.2+

```

```

Проверка на разширения
php -m | grep -E "(mysql|mbstring|xml|curl|gd) "
```

#### Стъпки за разгръщане

```bash
1. Влизане в директорията
cd /home/iqfound/models.iqvolution.ws/backend

2. Режим на поддръжка (опционално)
php artisan down

3. Актуализиране на кода
git pull origin main

4. Инсталиране на зависимости
composer install --no-dev --optimize-autoloader
npm ci && npm run build

5. Миграции
php artisan migrate --force

6. Импортиране на decision tree (ако има промени)
php artisan rules:import storage/app/data/decision_tree.json
php artisan rules:import storage/app/data/decision_tree_de.json

7. Кеширане на конфигурация
php artisan config:cache
php artisan route:cache
php artisan view:cache

8. Оптимизация на autoloader
composer dump-autoload --optimize

9. Рестартиране на сервизите
sudo systemctl restart php-fpm
или
sudo systemctl restart apache2

10. Край на режима на поддръжка
php artisan up
```

#### Конфигурация на Web сървър



```

Apache (.htaccess в public/):

```apache
<IfModule mod_rewrite.c>
    RewriteEngine On
    RewriteRule ^(.*)$ public/$1 [L]
</IfModule>
```

Nginx (пример за конфигурация):

```nginx
server {

```


```

```

listen 443 ssl http2;
server_name models.iqvolution.ws;
root /home/iqfound/models.iqvolution.ws/backend/public;

index index.php;

SSL конфигурация
ssl_certificate /path/to/certificate.crt;
ssl_certificate_key /path/to/private.key;

location / {
 try_files $uri $uri/ /index.php?$query_string;
}

location ~ /\.php$ {
 fastcgi_pass unix:/var/run/php/php8.2-fpm.sock;
 fastcgi_param SCRIPT_FILENAME
$realpath_root$fastcgi_script_name;
 include fastcgi_params;
}

location ~ /\.(!well-known).* {
 deny all;
}
}
...

Права на достъп до файловата система

```bash
# Собственост
chown -R www-data:www-data /home/iqfound/models.iqvolution.ws/backend

# Права за storage и bootstrap/cache
chmod -R 775 storage bootstrap/cache

# Права за .env
chmod 600 .env
```

```

#### #### CI/CD Pipeline

**\*\*[ТОЗИ РАЗДЕЛ ИЗИСКВА ПОПЪЛВАНЕ ОТ РАЗРАБОТЧИКА]\*\***

Понастоящем няма автоматизиран CI/CD pipeline. Препоръчително е да се настрои:

Ръчно настроен pipeline на build и deployment процеса.

#### #### Стратегия за разгръщане

**\*\*Текуща:\*\*** Rolling update (ръчно)

**\*\*Препоръчителни стратегии:\*\***

- **\*\*Blue-Green Deployment\*\*** - За zero-downtime deployments
- **\*\*Canary Release\*\*** - За постепенно разгръщане

```

4. Конфигурация и Настройки

4.1. База данни

Структура на таблиците

Таблица	Предназначение
`users`	Потребителски акаунти
`password_reset_tokens`	Токени за възстановяване на парола
`sessions`	Данни за сесии
`survey_responses`	Отговори на потребителите от анкетите
`cache`	Кеш таблица (fallback)
`jobs`	Опашка от задачи (ако се използва)

Изпълняване на миграции

```bash
# Проверка на статуса
php artisan migrate:status

# Изпълняване на нови миграции
php artisan migrate --force

# Rollback на последната миграция (ВНИМАНИЕ!)
php artisan migrate:rollback
```

Seed данни

```bash
# Изпълняване на всички seeders
php artisan db:seed

# Само конкретен seeder
php artisan db:seed --class=RealUserSeeder
```

4.2. Кеширане

Текуща конфигурация

```ini
CACHE_STORE=file
```

Опции за кеширане

Driver	Предназначение	Production Ready
`file`	Файлов кеш (текущ)	⚠ Ограничено
`redis`	In-memory кеш	✔ Препоръчително
`database`	DB-based кеш	✔ Приемливо
`memcached`	In-memory кеш	✔ Приемливо

Redis конфигурация (опционално)

```

```

```ini
CACHE_STORE=redis
REDIS_HOST=127.0.0.1
REDIS_PORT=6379
REDIS_PASSWORD=null
```

Команди за кеш

```bash
# Изчистване на всички кешове
php artisan cache:clear

# Кеширане на конфигурация
php artisan config:cache

# Кеширане на маршрути
php artisan route:cache

# Кеширане на изгледи
php artisan view:cache

# Пълно изчистване
php artisan optimize:clear
```

4.3. Фонови задачи

Текуща конфигурация

```ini
QUEUE_CONNECTION=sync
```

Sync режим: Задачите се изпълняват синхронно (блокиращо).

Асинхронна опашка (за production)

Redis queue:

```ini
QUEUE_CONNECTION=redis
```

Database queue:

```ini
QUEUE_CONNECTION=database
```

Supervisor конфигурация

Файл: `/etc/supervisor/conf.d/archmodels-worker.conf`

```ini
[program:archmodels-worker]
process_name=%(program_name)s_%(process_num)02d

```

```

command=php /home/iqfound/models.iqvolution.ws/backend/artisan
queue:work --sleep=3 --tries=3
autostart=true
autorestart=true
stopasgroup=true
killasgroup=true
user=www-data
numprocs=2
redirect_stderr=true
stdout_logfile=/var/log/archmodels-worker.log
```

```bash
sudo supervisorctl reread
sudo supervisorctl update
sudo supervisorctl start archmodels-worker:*
```

4.4. Decision Tree данни

Местоположение на файловете

Файл	Път	Предназначение
BG модели	`storage/app/data/decision_tree.json`	Български модели (BG-1 до BG-5)
DE модели	`storage/app/data/decision_tree_de.json`	Баварски модели (DE-1 до DE-5)

Импортиране на данни

```bash
# Импортиране на BG дърво
php artisan rules:import storage/app/data/decision_tree.json

# Импортиране на DE дърво
php artisan rules:import storage/app/data/decision_tree_de.json

# Анализ на данните
php artisan rules:analyze
```

5. Наблюдаемост и Логове

5.1. Логове

Конфигурация

```ini
LOG_CHANNEL=stack
LOG_STACK=single
LOG_LEVEL=error
```

Местоположение на лог файловете

| Лог | Път |

```

```
|-----|-----|
| Laravel log | `storage/logs/laravel.log` |
| PHP errors | Според php.ini конфигурация |
| Apache/Nginx | `/var/log/httpd/` или `/var/log/nginx/` |

Формат на логовете

Тип: Plain text (Monolog)

Пример:

...

[2026-02-02 10:30:45] production.ERROR: SQLSTATE[HY000] [2002]
Connection refused {"exception":"[object] (PDOException(code: 2002):
...)"}
```

#### Препоръчителни нива

```
Среда	Ниво
Development	`debug`
Staging	`info`
Production	`error` или `warning`
```

#### Дневна ротация на логове

```
```php
// config/logging.php
'daily' => [
    'driver' => 'daily',
    'path' => storage_path('logs/laravel.log'),
    'level' => env('LOG_LEVEL', 'error'),
    'days' => 14,
],
```
```

Променете `LOG\_CHANNEL=daily` в `.env` за автоматична ротация.

### 5.2. Метрики и Health Check

#### Health Check Endpoints

**\*\*[ВНИМАНИЕ: Няма имплементиран health check endpoint]\*\***

**\*\*Препоръчителна имплементация:\*\***

```
```php
// routes/web.php
Route::get('/health', function () {
    try {
        DB::connection()->getPdo();
        return response()->json([
            'status' => 'healthy',
            'database' => 'connected',
            'timestamp' => now()->toIso8601String()
        ]);
    } catch (\Exception $e) {
        return response()->json([

```

```

        'status' => 'unhealthy',
        'database' => 'disconnected',
        'error' => $e->getMessage()
    ], 503);
}
});
```

```

#### Ключови метрики за наблюдение

| Метрика              | Описание                | Аларма при       |
|----------------------|-------------------------|------------------|
| HTTP 5xx errors      | Сървърни грешки         | > 5% от заявките |
| Response time        | Време за отговор        | > 2 секунди      |
| Database connections | Активни връзки          | > 80% от max     |
| Disk usage           | Използвано пространство | > 85%            |
| Memory usage         | Използвана памет        | > 90%            |

### ### 5.3. Аларми

**\*\*[ТОЗИ РАЗДЕЛ ИЗИСКВА ПОПЪЛВАНЕ ОТ РАЗРАБОТЧИКА]\*\***

Понастоящем няма конфигурирана система за аларми. Препоръчително е да се настрои:

| Условие                      | Критичност | Действие                        |
|------------------------------|------------|---------------------------------|
| Недостъпност > 1 минута      | Критично   | Уведоми on-call инженер         |
| 5xx error rate > 5% за 5 мин | Високо     | Уведоми екипа                   |
| DB connection failed         | Критично   | Незабавно уведомяване           |
| Disk > 90%                   | Средно     | Уведоми за планирано почистване |
| Log file size > 500MB        | Ниско      | Планирана ротация               |

#### Инструменти за мониторинг

Препоръчителни решения:

- **\*\*UptimeRobot / Pingdom\*\*** - Uptime мониторинг
- **\*\*Sentry\*\*** - Error tracking
- **\*\*Laravel Telescope\*\*** - Debug и мониторинг (само development)
- **\*\*Grafana + Prometheus\*\*** - Метрики и визуализация

### ### 5.4. Distributed Tracing

**\*\*[НЕ Е ИМПЛЕМЕНТИРАНО]\*\***

За микросервизни архитектури се препоръчва:

- **\*\*Jaeger\*\***
- **\*\*Zipkin\*\***
- **\*\*AWS X-Ray\*\***

Текущата монолитна архитектура не изисква distributed tracing.

---

## ## 6. Поддръжка и Операции

### ### 6.1. Често срещани задачи

#### Рестартиране на услугите

```

```bash
# PHP-FPM
sudo systemctl restart php-fpm

# Apache
sudo systemctl restart httpd
# или
sudo systemctl restart apache2

# Nginx
sudo systemctl restart nginx

# Queue workers (ако се използва Supervisor)
sudo supervisorctl restart archmodels-worker:*
```

Проверка на статуса

```bash
# PHP-FPM
sudo systemctl status php-fpm

# Web сървър
sudo systemctl status httpd

# MySQL
sudo systemctl status mysqld

# Проверка на Laravel
cd /home/iqfound/models.iqvolution.ws/backend
php artisan about
```

Режим на поддръжка

```bash
# Включване на maintenance mode
php artisan down --message="Системата е в режим на поддръжка" --
retry=60

# Изключване
php artisan up

# Разрешаване на достъп от конкретен IP
php artisan down --allow=192.168.1.100
```

Изчистване на кешове

```bash
# Всички кешове
php artisan optimize:clear

# Само application кеш
php artisan cache:clear

# Само конфигурационен кеш
php artisan config:clear

```

```

# Само маршрути
php artisan route:clear
```

Прилагане на миграции

```bash
# Проверка за нови миграции
php artisan migrate:status

# Прилагане
php artisan migrate --force

# Rollback (ВНИМАНИЕ!)
php artisan migrate:rollback --step=1
```

Актуализиране на decision tree

```bash
# Редактирайте JSON файла
nano storage/app/data/decision_tree.json

# Валидирайте
php artisan rules:analyze

# Презаредете (ако се кешира)
php artisan cache:clear
```

6.2. Мониторинг на Performance

Места за търсене на bottlenecks

Компонент	Как да проверите
База данни	`SHOW PROCESSLIST;` в MySQL
Бавни заявки	Активирайте slow query log в MySQL
PHP memory	`php -i   grep memory_limit`
Disk I/O	`iostat -x 1 5`
Web сървър	Access/error логове

Оптимизация на Laravel

```bash
# Проверка на конфигурацията
php artisan config:show database

# Проверка на маршрутите
php artisan route:list

# Оптимизация
php artisan optimize
```

MySQL оптимизация

```sql
```

```

```

-- Показване на бавни заявки
SET GLOBAL slow_query_log = 'ON';
SET GLOBAL long_query_time = 2;
SET GLOBAL slow_query_log_file = '/var/log/mysql/slow-queries.log';

-- Проверка на индекси
SHOW INDEX FROM survey_responses;

-- Анализ на таблица
ANALYZE TABLE survey_responses;
```



### 6.3. Процедури при инцидент



#### Ескалационна матрица



| Ниво          | Критерии                          | Отговорен        | Време за реакция  |
|---------------|-----------------------------------|------------------|-------------------|
| P1 - Критично | Системата е недостъпна            | On-call инженер  | < 15 минути       |
| P2 - Високо   | Основна функционалност не работи  | DevOps екип      | < 1 час           |
| P3 - Средно   | Вторична функционалност засегната | Development екип | < 4 часа          |
| P4 - Ниско    | Козметични проблеми               | Планирано        | Следваща итерация |



#### Runbook за често срещани проблеми



**Проблем: 500 Internal Server Error**



```

```bash
# 1. Проверете логовете
tail -100 storage/logs/laravel.log

# 2. Проверете PHP-FPM
sudo systemctl status php-fpm
sudo journalctl -u php-fpm -n 50

# 3. Проверете права
ls -la storage/
chmod -R 775 storage bootstrap/cache

# 4. Изчистете кеш
php artisan cache:clear
php artisan config:clear
```

```



**Проблем: Database connection refused**



```

```bash
# 1. Проверете MySQL статус
sudo systemctl status mysqld

# 2. Рестартирайте MySQL
sudo systemctl restart mysqld

# 3. Проверете .env конфигурация
cat .env | grep DB_

```


```

```

4. Тествайте връзката
mysql -u iqfound_arch_models_user_admin -p iqfound_arch_models
\`\`\`

Проблем: Имейлите не се изпращат

\`\`\`bash
1. Проверете SMTP конфигурация
cat .env | grep MAIL_

2. Тествайте SMTP връзката
telnet <MAIL_HOST> 587

3. Проверете queue (ако е async)
php artisan queue:work --once

4. Проверете failed jobs
php artisan queue:failed
\`\`\`

Контакти

[ТОЗИ РАЗДЕЛ ИЗИСКВА ПОПЪЛВАНЕ ОТ РАЗРАБОТЧИКА]

Роля	Име	Контакт
Primary On-call	[ИМЕ]	[ТЕЛЕФОН/EMAIL]
Secondary On-call	[ИМЕ]	[ТЕЛЕФОН/EMAIL]
DevOps Lead	[ИМЕ]	[ТЕЛЕФОН/EMAIL]
Project Manager	[ИМЕ]	[ТЕЛЕФОН/EMAIL]

7. Сигурност

7.1. Аутентикация и Авторизация

Механизъм

Аспект	Имплементация
Аутентикация	Session-based (Laravel Breeze)
Парола хеширане	Bcrypt с 12 рунда
Сесии	Database-backed
CSRF защита	Активна (middleware)
Email потвърждение	Задължително

Маршрути с аутентикация

Маршрут	Изискване
`/wizard`, `/wizard-de`	Аутентикация + Email потвърждение
`/api/tree`, `/api/evaluate`	Аутентикация + Email потвърждение
`/dashboard`, `/profile`	Аутентикация + Email потвърждение
`/`, `/login`, `/register`	Публични

Rate Limiting

| Endpoint | Лимит |

```

```

|-----|-----|
| Email verification | 6 заявки/минута |
| Login attempts | [ДА СЕ КОНФИГУРИРА] |
| API endpoints | [ДА СЕ КОНФИГУРИРА] |

7.2. Най-добри практики

Критични действия за production

- [] **APP_DEBUG=false** - Задължително!
- [] **Ротирайте всички exposed пароли** - DB и SMTP пароли са видими в .env
- [] **HTTPS/TLS** - Използвайте само HTTPS
- [] **Secure headers** - Добавете security headers

Ротация на секрети

```bash
# 1. Генериране на нов APP_KEY
php artisan key:generate

# 2. Промяна на DB парола
# В MySQL:
ALTER USER 'iqfound_arch_models_user_admin'@'localhost'
IDENTIFIED BY 'НОВА_СИЛНА_ПАРОЛА';

# 3. Актуализиране на .env
nano .env

# 4. Изчистване на кешове
php artisan cache:clear
php artisan config:clear
```

HTTPS/TLS конфигурация

Apache (mod_ssl):

```apache
```apache
<VirtualHost *:443>
 ServerName models.iqvolution.ws
 DocumentRoot /home/iqfound/models.iqvolution.ws/backend/public

 SSLEngine on
 SSLCertificateFile /path/to/certificate.crt
 SSLCertificateKeyFile /path/to/private.key
 SSLCertificateChainFile /path/to/chain.crt

 # Modern TLS settings
 SSLProtocol all -SSLv3 -TLSv1 -TLSv1.1
 SSLCipherSuite ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256
 SSLHonorCipherOrder off
</VirtualHost>

Redirect HTTP to HTTPS
<VirtualHost *:80>
 ServerName models.iqvolution.ws
 Redirect permanent / https://models.iqvolution.ws/

```

```
</VirtualHost>
```

```
```
```

Security Headers

Добавете в `.htaccess` или сървърната конфигурация:

```
```apache
```

```
Header always set X-Frame-Options "SAMEORIGIN"
Header always set X-Content-Type-Options "nosniff"
Header always set X-XSS-Protection "1; mode=block"
Header always set Referrer-Policy "strict-origin-when-cross-origin"
Header always set Content-Security-Policy "default-src 'self'; script-
src 'self' 'unsafe-inline'; style-src 'self' 'unsafe-inline'"
Header always set Strict-Transport-Security "max-age=31536000;
includeSubDomains"
```
```

Firewall правила

```
```bash
```

```
Позволете само необходимите портове
sudo firewall-cmd --permanent --add-service=http
sudo firewall-cmd --permanent --add-service=https
sudo firewall-cmd --permanent --add-service=ssh
sudo firewall-cmd --reload
```

```
Блокирайте директен достъп до MySQL отвън
sudo firewall-cmd --permanent --remove-service=mysql
```
```

Сканиране за уязвимости

```
```bash
```

```
PHP зависимости
composer audit
```

```
JavaScript зависимости
npm audit
```

```
За автоматично коригиране (внимателно!)
npm audit fix
```
```

7.3. Одит логове

****[ЧАСТИЧНО ИМПЛЕМЕНТИРАНО]****

Текущо се логват:

- Грешки при аутентикация (в `laravel.log`)
- Изключения и грешки

****Препоръчително да се добави:****

- Успешни и неуспешни login опити
- Промени в профила
- CRUD операции по анкети
- Административни действия

****Пример за audit logging:****

```

```php
// app/Listeners/LogSuccessfulLogin.php
Log::info('User logged in', [
 'user_id' => $event->user->id,
 'email' => $event->user->email,
 'ip' => request()->ip(),
 'user_agent' => request()->userAgent()
]);
```

---

## 8. Резервни Копия и Възстановяване

### 8.1. Какво се backup-ва

Компонент	Критичност	Метод
**MySQL база данни**	Критично	mysqldump
**Decision tree JSON файлове**	Високо	File copy
**Uploaded files**	Средно	rsync
**.env файл**	Критично	Secure copy
**SSL сертификати**	Високо	File copy

### 8.2. Честота и ретенция

Тип backup	Честота	Ретенция
Пълен DB backup	Ежедневно, 02:00	30 дни
Инкрементален backup	На всеки 6 часа	7 дни
Файлов backup	Ежеседмично	4 седмици
Off-site backup	Ежеседмично	90 дни

### 8.3. Backup скриптове

#### Database backup

```bash
#!/bin/bash
/opt/scripts/backup-db.sh

DATE=$(date +%Y%m%d_%H%M%S)
BACKUP_DIR="/backups/mysql"
DB_NAME="iqfound_arch_models"
DB_USER="backup_user"
DB_PASS="BACKUP_PASSWORD"

mkdir -p $BACKUP_DIR

Създаване на backup
mysqldump -u $DB_USER -p$DB_PASS $DB_NAME | gzip >
$BACKUP_DIR/${DB_NAME}_${DATE}.sql.gz

Изтриване на стари backups (над 30 дни)
find $BACKUP_DIR -name "*.sql.gz" -mtime +30 -delete

Logging

```

```

echo "$(date): Backup completed: ${DB_NAME}_${DATE}.sql.gz" >>
/var/log/backup.log
```

#### Crontab entry

```cron
Database backup - ежедневно в 02:00
0 2 * * * /opt/scripts/backup-db.sh

Files backup - ежеседмично в неделя 03:00
0 3 * * 0 rsync -avz /home/iqfound/models.iqvolution.ws/
/backups/files/ --exclude='node_modules' --exclude='vendor'
```

#### 8.4. Процедура за възстановяване (Disaster Recovery)

#### Сценарий: Пълна загуба на сървър

**Време за възстановяване (RTO):** 2-4 часа  

**Точка на възстановяване (RPO):** До 6 часа (според backup честотата)

#### Стъпки за възстановяване

```bash
1. Подготовка на нов сървър
Инсталирайте: PHP 8.2+, MySQL 8.0+, Composer, Node.js, Nginx/Apache

2. Възстановяване на кода
git clone <REPOSITORY_URL> /home/iqfound/models.iqvolution.ws
cd /home/iqfound/models.iqvolution.ws/backend

3. Възстановяване на .env
Копирайте backup на .env файла или създайте нов

4. Инсталиране на зависимости
composer install --no-dev --optimize-autoloader
npm ci && npm run build

5. Възстановяване на база данни
gunzip < /backups/mysql/iqfound_arch_models_LATEST.sql.gz | mysql -u
root -p iqfound_arch_models

6. Възстановяване на файлове
rsync -avz /backups/files/ /home/iqfound/models.iqvolution.ws/

7. Настройка на права
chown -R www-data:www-data /home/iqfound/models.iqvolution.ws/backend
chmod -R 775 storage bootstrap/cache

8. Генериране на кешове
php artisan config:cache
php artisan route:cache
php artisan view:cache

9. Тестване
php artisan migrate:status
curl -I https://models.iqvolution.ws

```

```

10. DNS актуализация (ако е необходимо)
Насочете домейна към новия IP адрес
```

#### Тестване на backup

**Честота:** Минимум веднъж месечно

```bash
Възстановяване в тестова среда
mysql -u root -p test_restore <
/backups/mysql/iqfound_arch_models_LATEST.sql.gz

Проверка на данните
mysql -u root -p -e "SELECT COUNT(*) FROM test_restore.users;"
mysql -u root -p -e "SELECT COUNT(*) FROM
test_restore.survey_responses;"
```

---

## 9. Мащабиране

### 9.1. Хоризонтално vs. Вертикално мащабиране

#### Препоръка за текущата архитектура

**Вертикално мащабиране** е препоръчително за текущия етап:
- Монолитна Laravel + Vue.js архитектура
- Session-based аутентикация
- Файлов кеш

Мащабиране	Кога да се използва
**Вертикално**	До ~10,000 дневни потребители
**Хоризонтално**	Над 10,000 дневни потребители или за HA

### 9.2. Вертикално мащабиране

#### Препоръчителни ъпгрейди

Етап	CPU	RAM	Дисково пространство
Старт	1 vCPU	1 GB	10 GB SSD
Растеж	2 vCPU	4 GB	20 GB SSD
Голям трафик	4 vCPU	8 GB	50 GB SSD

### 9.3. Хоризонтално мащабиране

#### Предварителни условия

Преди хоризонтално мащабиране:

1. **Мигрирайте сесиите към Redis/Database**
   ```ini
 SESSION_DRIVER=redis
   ```

2. **Мигрирайте кеша към Redis**

```

```

` ``ini
CACHE_STORE=redis
` ``

```

3. ****Използвайте споделено файлово хранилище****
 - NFS или S3 за `storage/app`

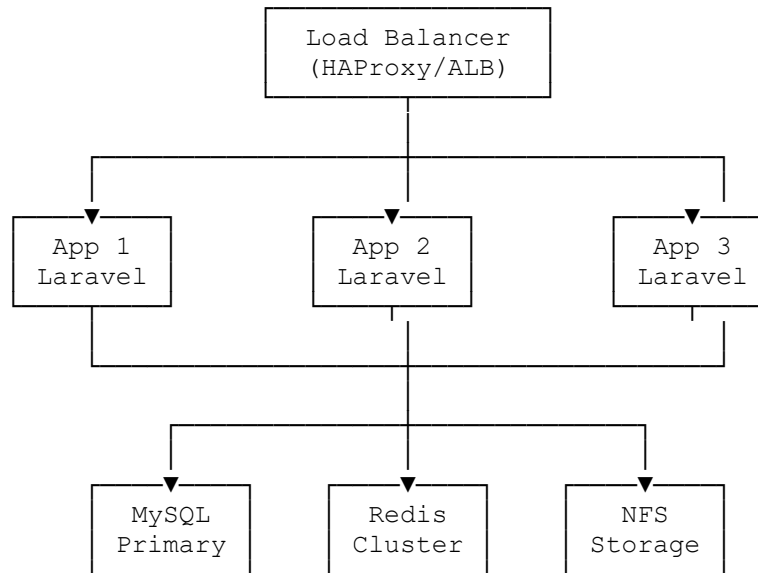
4. ****Настройте load balancer****
 - HAProxy, Nginx, AWS ALB

Архитектура при хоризонтално мащабиране

```

` ``

```



```

` ``

```

9.4. Автоматично мащабиране

****[НЕ Е КОНФИГУРИРАНО]****

За cloud среди (AWS, GCP, Azure):

| Метрика | Праг за scale-up | Праг за scale-down |
|---------------|------------------|--------------------|
| CPU | > 70% за 5 мин | < 30% за 15 мин |
| Memory | > 80% за 5 мин | < 40% за 15 мин |
| Request queue | > 100 pending | < 10 pending |

```

---

```

10. Приложения

10.1. Списък на API endpoints

| Метод | Endpoint | Описание | Auth |
|-------|-----------------|--------------------------------|------------|
| GET | `/` | Начална страница (SPA) | Не |
| GET | `/me` | Информация за текущ потребител | Опционално |
| GET | `/refresh-csrf` | Обновяване на CSRF токен | Опционално |
| POST | `/register` | Регистрация | Не |

| | | | |
|--------|-------------------------------|------------------------|----|
| POST | /login | Вход | Не |
| POST | /logout | Изход | Да |
| POST | /forgot-password | Забравена парола | Не |
| POST | /reset-password | Нулиране на парола | Не |
| GET | /api/tree | BG decision tree | Да |
| POST | /api/evaluate | Оценка на BG отговори | Да |
| GET | /api/tree-de | DE decision tree | Да |
| POST | /api/evaluate-de | Оценка на DE отговори | Да |
| GET | /survey-responses | История на анкети | Да |
| POST | /survey-responses | Запазване на анкета | Да |
| GET | /survey-responses/{id}/export | PDF експорт | Да |
| GET | /profile | Профил страница | Да |
| PATCH | /profile | Актуализация на профил | Да |
| DELETE | /profile | Изтриване на профил | Да |

10.2. Artisan команди

| Команда | Описание |
|---------------------------------|------------------------------|
| php artisan rules:import <file> | Импортиране на decision tree |
| php artisan rules:analyze | Анализ на decision tree |
| php artisan user:create | Създаване на потребител |
| php artisan migrate | Изпълнение на миграции |
| php artisan db:seed | Seed на данни |
| php artisan cache:clear | Изчистване на кеш |
| php artisan config:cache | Кеширане на конфигурация |
| php artisan route:cache | Кеширане на маршрути |
| php artisan view:cache | Кеширане на изгледи |
| php artisan down | Maintenance mode |
| php artisan up | Край на maintenance mode |

10.3. Структура на проекта

...

```

/home/igfound/models.iqvolution.ws/
├── backend/                                     # Laravel приложение
│   ├── app/
│   │   ├── Http/Controllers/                 # Контролери
│   │   ├── Models/                          # Eloquent модели
│   │   ├── Services/                        # Business logic
│   │   ├── Console/Commands/                # Artisan команди
│   │   └── Notifications/                   # Email нотификации
│   ├── database/
│   │   ├── migrations/                      # DB миграции
│   │   └── seeders/                         # Seed данни
│   ├── resources/
│   │   ├── js/                             # Vue.js SPA
│   │   ├── css/                           # TailwindCSS
│   │   └── views/                          # Blade templates
│   ├── routes/                             # Web и API маршрути
│   ├── config/                             # Конфигурация
│   ├── storage/
│   │   ├── app/data/                       # Decision tree JSON
│   │   └── logs/                           # Логове
│   ├── public/                             # Web root
│   ├── .env                                # Environment
│   ├── composer.json                       # PHP dependencies
│   ├── package.json                       # JS dependencies
└── docs/                                   # Документация

```

```

└─ README.md                                     # Обзор на проекта
\ \ \

### 10.4. Чек-листа за deployment

#### Преди deployment

- [ ] Код е прегледан и тестван
- [ ] Всички тестове минават (`php artisan test`)
- [ ] `APP_DEBUG=false` в production `.env`
- [ ] Пароли не са hardcoded в кода
- [ ] Dependencies са актуални (`composer audit`, `npm audit`)

#### По време на deployment

- [ ] Включен maintenance mode (`php artisan down`)
- [ ] Database backup преди миграции
- [ ] Миграции приложени (`php artisan migrate --force`)
- [ ] Кешове изчистени и регенерирани
- [ ] Assets билднати (`npm run build`)

#### След deployment

- [ ] Maintenance mode изключен (`php artisan up`)
- [ ] Health check endpoint отговаря
- [ ] Критични потоци тествани (login, wizard, export)
- [ ] Логове проверени за грешки
- [ ] Мониторинг потвърден

---

## Ревизии на документа

Версия	Дата	Автор	Промени
1.0	2026-02-02	Стоян Тодоров	Първоначална версия

---

```

ARCHModels

> Интерактивна система за избор на строителен модел за България и Бавария

```
[![PHP](https://img.shields.io/badge/PHP-8.2-777BB4?logo=php)](https://php.net)
[![Laravel](https://img.shields.io/badge/Laravel-12-FF2D20?logo=laravel)](https://laravel.com)
[![Vue.js](https://img.shields.io/badge/Vue.js-3-4FC08D?logo=vue.js)](https://vuejs.org)
[![License](https://img.shields.io/badge/License-Proprietary-red)]()
```

За проекта

ARCHModels е интерактивна уеб система, която помага на възложители в строителството да изберат подходящ строителен модел чрез структурирано дърво на решения. Системата обслужва два регулаторни режима:

- ****България****: BG-1 до BG-5 модели съгласно българското законодателство
- ****Бавария (Германия)****: DE-1 до DE-5 модели съгласно баварското законодателство

Целева аудитория

- ****Възложители****: Публични и частни организации, които планират строителни проекти
- ****Консултанти****: Инженери, архитекти, юристи в строителния сектор
- ****Образователни институции****: За обучение по избор на строителни модели

Дърво на решенията

...

Начало → Избор на държава (BG/DE)

↓

Серия въпроси → Филтриране на модели

↓

Препоръка → Модел BG-X или DE-X + обяснение

...\

Съдържание на документацията

| Документ | Описание |
|--|---|
| ARCHITECTURE.md | Техническа архитектура и компонентна структура |
| DEPLOYMENT.md | Инструкции за деплой в production (cPanel/shared hosting) |
| CONFIGURATION.md | Конфигурационни параметри и environment variables |
| API_DOCUMENTATION.md | Описание на Inertia маршрути и комуникация |
| OPERATIONS.md | Мониторинг, логове и troubleshooting |
| SECURITY.md | Практики за сигурност и управление на секрети |

| [BACKUP_DISASTER_RECOVERY.md] (BACKUP_DISASTER_RECOVERY.md) |
Стратегии за backup и disaster recovery |
| [SCALING.md] (SCALING.md) | Препоръки за мащабиране на приложението |

Бърз старт за разработка

Предварителни изисквания

- PHP 8.2 или по-нова версия
- Composer 2.x
- Node.js 18+ и npm
- MySQL 8.0+
- Git

Локална инсталация

```
```bash
1. Клониране на репозиторията
git clone https://github.com/your-org/archmodels.git
cd archmodels

2. Настройка на backend
cd backend
cp .env.example .env
composer install

3. Генериране на application key
php artisan key:generate

4. Конфигурация на базата данни
Редактирайте .env и въведете вашите MySQL credentials:
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=archmodels
DB_USERNAME=your_username
DB_PASSWORD=your_password

5. Миграция на базата данни и seed данни
php artisan migrate --seed

6. Инсталация на frontend зависимости
npm install

7. Компилиране на assets и стартиране на dev server
npm run dev

8. В отделен терминал стартирайте Laravel dev server
php artisan serve
```
```

Приложението ще бъде достъпно на `http://localhost:8000`

Импорт на данни за дървото на решения

```
```bash
Импортиране на данните за България и Бавария
php artisan rules:import
```

```

Проверка на генерираните файлове
ls -la storage/app/data/
Очаквани файлове:
- decision_tree.json (България)
- decision_tree_de.json (Бавария)
```

## Структура на проекта
```
archmodels/
├── backend/ # Laravel приложение
│ ├── app/
│ │ ├── Http/
│ │ │ ├── Controllers/ # Inertia контролери
│ │ │ └── Middleware/ # Middleware за auth и locale
│ │ ├── Models/ # Eloquent модели
│ │ │ ├── Question.php
│ │ │ ├── Answer.php
│ │ │ ├── DecisionNode.php
│ │ │ └── User.php
│ │ ├── Services/ # Бизнес логика
│ │ │ └── DecisionTreeEngine.php
│ ├── database/
│ │ ├── migrations/ # Database schema
│ │ └── seeders/ # Начални данни за BG/DE
│ ├── resources/
│ │ ├── js/
│ │ │ ├── Pages/ # Vue страници (Inertia)
│ │ │ │ ├── Home.vue
│ │ │ │ ├── Wizard.vue
│ │ │ │ └── Result.vue
│ │ │ └── Auth/ # Laravel Breeze компоненти
│ │ ├── Components/ # Vue компоненти
│ │ ├── app.js
│ │ ├── css/
│ │ │ └── app.css # TailwindCSS
│ ├── routes/
│ │ ├── web.php # Inertia маршрути
│ │ └── api.php # API endpoints (при необходимост)
│ ├── storage/
│ │ └── app/
│ │ └── data/ # JSON данни за дървото
│ ├── tests/ # PHPUnit тестове
│ ├── docs/ # Допълнителна документация
│ │ ├── tekst_bg_models.md # Описание на BG модели
│ │ ├── tekst_de_models.md # Описание на DE модели
│ │ └── USER-GUIDE-DUAL-SYSTEM.md
│ ├── .env.example # Примерна конфигурация
│ ├── composer.json # PHP зависимости
│ ├── package.json # JavaScript зависимости
│ └── README.md # Този файл
```

## Технологичен стек

### Backend
- **Framework**: Laravel 12
- **PHP Version**: 8.2

```

- ****ORM****: Eloquent
- ****Authentication****: Laravel Breeze
- ****Database****: MySQL 8.0+
- ****Session/Cache Driver****: File (може да се upgrade към Redis)

Frontend

- ****Framework****: Vue 3 (Composition API)
- ****State Management****: Pinia (при необходимост)
- ****Routing****: Vue Router + Inertia.js
- ****Styling****: TailwindCSS 3
- ****Build Tool****: Vite

Интеграция

- ****SSR Bridge****: Inertia.js (връзка между Laravel и Vue без API)
- ****Локализация****: Laravel translations (български език)

Основни функционалности

1. Избор на държава

- Модален прозорец при стартиране
- България (BG) или Бавария (DE)
- Различни цветови схеми за всяка държава

2. Интерактивен въпросник

- Поредица от въпроси базирани на избраната държава
- Динамично филтриране на допустими модели
- Визуален индикатор на напредъка

3. Препоръка на модел

- Крайна препоръка (BG-1 до BG-5 или DE-1 до DE-5)
- Обяснение на избора (trase на решенията)
- Детайлно описание на препоръчания модел

4. Автентикация

- Регистрация и вход (Laravel Breeze)
- Запазване на предишни сесии (за регистрирани потребители)
- История на препоръките

Работа в development режим

Горещо презареждане (Hot Module Replacement)

```
```bash
npm run dev
```
```

Компилиране за production

```
```bash
npm run build
```
```

Изпълнение на тестове

```
```bash
Backend тестове
php artisan test

Frontend тестове (ако са конфигурирани)
npm run test
```
```

```
### Изчистване на кеш
```bash
php artisan cache:clear
php artisan config:clear
php artisan route:clear
php artisan view:clear
```

## Известни проблеми и ограничения

- **Споделен хостинг**: При cPanel/shared hosting може да има ограничения за изпълнение на Artisan команди
- **File cache**: При липса на Redis, file driver може да бъде по-бавен при високо натоварване
- **Локализация**: Понастоящем поддържа само български език (немски и английски в бъдещи версии)

## Помощ и поддръжка

За въпроси, проблеми или предложения:

- **Issues**: [GitHub Issues] (https://github.com/your-org/archmodels/issues)
- **Email**: support@archmodels.example
- **Документация**: Вижте документите по-горе в секцията "Съдържание на документацията"

## Лиценз

Този проект е с proprietary лиценз. Всички права запазени.

---

**Версия**: 1.0.0
**Последна актуализация**: 2026-01-31
```

Архитектура на ARCHModels

Този документ описва техническата архитектура на ARCHModels - интерактивната система за избор на строителен модел за България и Бавария.

Съдържание

- [Високо-нивова архитектура] (#високо-нивова-архитектура)
- [Компонентна архитектура] (#компонентна-архитектура)
- [Структура на базата данни] (#структура-на-базата-данни)
- [Поток на данните] (#поток-на-данните)
- [Decision Tree Engine] (#decision-tree-engine)
- [Frontend архитектура] (#frontend-архитектура)
- [Backend архитектура] (#backend-архитектура)

Високо-нивова архитектура

ARCHModels следва монолитна Single Page Application (SPA) архитектура с използване на Inertia.js като мост между Laravel и Vue.

```
``mermaid
graph TB
    subgraph "Client Layer"
        Browser[Browser<br/>Vue 3 SPA]
        Router[Vue Router]
    end

    subgraph "Web Server Layer"
        Nginx[Nginx/Apache<br/>Web Server]
        SSL[SSL/TLS<br/>Encryption]
    end

    subgraph "Application Layer"
        Laravel[Laravel 12<br/>Application]
        Inertia[Inertia.js<br/>SSR Bridge]
        Breeze[Laravel Breeze<br/>Authentication]
        Engine[Decision Tree<br/>Engine]
    end

    subgraph "Data Layer"
        MySQL[(MySQL 8.0<br/>Database)]
        FileCache[File Cache<br/>Session/Config]
        JSONData[JSON Files<br/>Decision Trees]
    end

    Browser <-->|HTTPS| SSL
    SSL <--> Nginx
    Nginx <--> Laravel
    Laravel <--> Inertia
    Inertia <--> Router
    Laravel <--> Breeze
    Laravel <--> Engine
    Laravel <--> MySQL
    Laravel <--> FileCache
    Engine <--> JSONData

    style Browser fill:#4FC08D
```

```

        style Laravel fill:#FF2D20
        style MySQL fill:#4479A1
        style Inertia fill:#9553E9
    ...

## Компонентна архитектура

### 1. Decision Tree Engine

Централният компонент на системата, отговорен за логиката на дървото на решения.

**Местоположение**:`backend/app/Services/DecisionTreeEngine.php`

**Отговорности**:  

- Зареждане на JSON структурата за дървото на решения (BG или DE)  

- Поддържане на текущ набор от допустими модели  

- Филтриране на модели базирани на отговори  

- Определяне на следващия най-информативен въпрос  

- Генериране на финална препоръка

**Ключови методи**:  

```php
class DecisionTreeEngine
{
 // Зарежда дървото за дадена държава
 public function loadTree(string $country): void

 // Инициализира нова сесия
 public function startSession(string $country): array

 // Обработка отговор и връща следващ въпрос
 public function processAnswer(int $questionId, int $answerId):
array

 // Връща финална препоръка
 public function getFinalRecommendation(): array

 // Генерира trace на взетите решения
 public function getDecisionTrace(): array
}
...

2. Модели на данни

Question Model
```php
// backend/app/Models/Question.php
class Question extends Model
{
    protected $fillable = ['country', 'text', 'order', 'category'];

    public function answers(): HasMany
    {
        return $this->hasMany(Answer::class);
    }
}
...

```

```

#### Answer Model
```php
// backend/app/Models/Answer.php
class Answer extends Model
{
 protected $fillable = ['question_id', 'text', 'model_filter'];

 // model_filter: JSON array от допустими модели след този отговор
 // Пример: ["BG-1", "BG-2", "BG-3"] или ["DE-1", "DE-5"]

 public function question(): BelongsTo
 {
 return $this->belongsTo(Question::class);
 }
}
...

DecisionNode Model
```php
// backend/app/Models/DecisionNode.php
class DecisionNode extends Model
{
    // Съхранява резултати от минали сесии
    protected $fillable = [
        'user_id',
        'country',
        'recommended_model',
        'decision_path', // JSON със следа на решенията
        'completed_at'
    ];

    protected $casts = [
        'decision_path' => 'array',
        'completed_at' => 'datetime',
    ];
}
...

#### User Model
```php
// backend/app/Models/User.php (or Laravel Breeze)
class User extends Authenticatable
{
 public function decisionNodes(): HasMany
 {
 return $this->hasMany(DecisionNode::class);
 }
}
...

3. Контролери (Inertia)

WizardController
```php
// backend/app/Http/Controllers/WizardController.php
class WizardController extends Controller
{
    public function start(Request $request)
    {

```

```

        // Стартира нова сесия за избраната държава
        return Inertia::render('Wizard', [
            'country' => $request->country,
            'firstQuestion' => $this->engine->getFirstQuestion(),
        ]);
    }

    public function answer(Request $request)
    {
        // Обработка отговор и връща следващ въпрос или резултат
        $result = $this->engine->processAnswer(
            $request->questionId,
            $request->answerId
        );

        if ($result['isComplete']) {
            return Inertia::render('Result', [
                'recommendation' => $result['recommendation'],
                'trace' => $result['trace'],
            ]);
        }

        return Inertia::render('Wizard', [
            'nextQuestion' => $result['nextQuestion'],
            'progress' => $result['progress'],
        ]);
    }
}
...

```

Структура на базата данни

```

```mermaid
erDiagram
 users ||--o{ decision_nodes : creates
 questions ||--o{ answers : has

 users {
 bigint id PK
 string name
 string email UK
 timestamp email_verified_at
 string password
 timestamp created_at
 timestamp updated_at
 }

 questions {
 bigint id PK
 string country "BG or DE"
 text text
 integer order
 string category
 timestamp created_at
 timestamp updated_at
 }

 answers {
 bigint id PK
 }

```

```

 bigint question_id FK
 text text
 json model_filter "допустими модели"
 timestamp created_at
 timestamp updated_at
 }

 decision_nodes {
 bigint id PK
 bigint user_id FK "nullable"
 string country
 string recommended_model "BG-1..5 или DE-1..5"
 json decision_path
 timestamp completed_at
 timestamp created_at
 timestamp updated_at
 }
}

...

Миграции

Create Questions Table
```php
Schema::create('questions', function (Blueprint $table) {
    $table->id();
    $table->string('country', 2)->index(); // 'BG' или 'DE'
    $table->text('text');
    $table->integer('order')->default(0);
    $table->string('category')->nullable();
    $table->timestamps();
});
```

Create Answers Table
```php
Schema::create('answers', function (Blueprint $table) {
    $table->id();
    $table->foreignId('question_id')->constrained()-
>onDelete('cascade');
    $table->text('text');
    $table->json('model_filter'); // ["BG-1", "BG-2"]
    $table->timestamps();
});
```

Create Decision Nodes Table
```php
Schema::create('decision_nodes', function (Blueprint $table) {
    $table->id();
    $table->foreignId('user_id')->nullable()->constrained()-
>onDelete('set null');
    $table->string('country', 2);
    $table->string('recommended_model', 10);
    $table->json('decision_path');
    $table->timestamp('completed_at')->nullable();
    $table->timestamps();

    $table->index(['country', 'recommended_model']);
});

```

...

Поток на данните

1. Стартиране на сесия

```
```mermaid
sequenceDiagram
 participant U as User
 participant B as Browser (Vue)
 participant L as Laravel
 participant E as DecisionTreeEngine
 participant J as JSON Data

 U->>B: Посещава Номе страница
 B->>U: Показва модал за избор на държава
 U->>B: Избира България (BG)
 B->>L: POST /wizard/start {country: 'BG'}
 L->>E: startSession('BG')
 E->>J: Записва decision_tree.json
 J-->>E: Връща JSON структура
 E->>L: Връща първи въпрос
 L-->>B: Inertia::render('Wizard', [...])
 B-->>U: Показва първи въпрос
...

```

### 2. Обработка на отговор

```
```mermaid
sequenceDiagram
    participant U as User
    participant B as Browser (Vue)
    participant L as Laravel
    participant E as DecisionTreeEngine
    participant S as Session

    U->>B: Избира отговор A1
    B->>L: POST /wizard/answer {questionId: 1, answerId: 1}
    L->>S: Извлича текуща сесия
    L->>E: processAnswer(1, 1)
    E->>E: Филтрира допустими модели
    E->>E: Определя следващ въпрос
    E->>L: Връща Q2 + progress 20%
    L->>S: Записва текущо състояние
    L-->>B: Inertia::render('Wizard', [...])
    B-->>U: Показва въпрос Q2 + progress bar
...

```

3. Финална препоръка

```
```mermaid
sequenceDiagram
 participant U as User
 participant B as Browser (Vue)
 participant L as Laravel
 participant E as DecisionTreeEngine
 participant DB as MySQL

 U->>B: Отговаря на последен въпрос

```

```

B->>L: POST /wizard/answer {questionId: 5, answerId: 12}
L->>E: processAnswer(5, 12)
E->>E: Остава 1 модел -> BG-3
E->>L: {isComplete: true, model: 'BG-3', trace: [...]}
L->>DB: Записва DecisionNode
DB-->>L: Потвърждава запис
L-->>B: Inertia::render('Result', [...])
B-->>U: Показва препоръка BG-3 + обяснение
...

Decision Tree Engine

Структура на JSON файла

```json
{
  "country": "BG",
  "version": "1.0",
  "models": ["BG-1", "BG-2", "BG-3", "BG-4", "BG-5"],
  "questions": [
    {
      "id": 1,
      "text": "Каква е стойността на проекта?",
      "category": "budget",
      "answers": [
        {
          "id": 1,
          "text": "До 500 000 лв.",
          "allowedModels": ["BG-1", "BG-2"]
        },
        {
          "id": 2,
          "text": "Над 500 000 лв.",
          "allowedModels": ["BG-3", "BG-4", "BG-5"]
        }
      ]
    },
    {
      "id": 2,
      "text": "Имате ли опит в управление на строителни проекти?",
      "category": "experience",
      "answers": [
        {
          "id": 3,
          "text": "Да, имам значителен опит",
          "allowedModels": ["BG-1", "BG-3"]
        },
        {
          "id": 4,
          "text": "Не, нямам опит",
          "allowedModels": ["BG-2", "BG-4", "BG-5"]
        }
      ]
    }
  ],
  "recommendations": {
    "BG-1": {
      "title": "Модел BG-1: Традиционен договор",
      "description": "Подходящ за възложители с опит...",

```

```

        "advantages": [...],
        "disadvantages": [...]]
    }
}
...

### Алгоритъм за филтриране

```php
// Псевдокод на алгоритъма
function processAnswer($questionId, $answerId) {
 // 1. Извличаме отговора
 $answer = Answer::find($answerId);

 // 2. Пресичаме текущите допустими модели с тези от отговора
 $this->allowedModels = array_intersect(
 $this->allowedModels,
 $answer->model_filter
);

 // 3. Проверяваме дали сме стигнали до краен резултат
 if (count($this->allowedModels) === 1) {
 return [
 'isComplete' => true,
 'recommendation' => $this->allowedModels[0],
 'trace' => $this->getDecisionTrace()
];
 }

 // 4. Избираме следващ най-информативен въпрос
 $nextQuestion = $this->selectNextQuestion($this->allowedModels);

 return [
 'isComplete' => false,
 'nextQuestion' => $nextQuestion,
 'progress' => $this->calculateProgress()
];
}
...

Frontend архитектура

Vue компоненти

...

resources/js/
├── Pages/ # Inertia страници
│ ├── Home.vue # Начална страница + country selector
│ ├── Wizard.vue # Интерактивен въпросник
│ ├── Result.vue # Резултат с препоръка
│ └── Auth/ # Laravel Breeze страници
│ ├── Login.vue
│ ├── Register.vue
│ └── ForgotPassword.vue
├── Components/ # Преизползваеми компоненти
│ ├── CountrySelector.vue # Модал за избор BG/DE
│ ├── QuestionCard.vue # Показва въпрос + отговори
│ └── ProgressBar.vue # Индикатор на напредъка

```

```

├── DecisionTrace.vue # Визуализация на пътя
├── ModelCard.vue # Карта с описание на модел
├── Layouts/
│ ├── AppLayout.vue # Основен layout
│ └── GuestLayout.vue # Layout за guest потребители
...

```

### Състояние на приложението (Pinia Store)

```

```javascript
// stores/wizard.js
export const useWizardStore = defineStore('wizard', {
  state: () => ({
    country: null,           // 'BG' или 'DE'
    currentQuestion: null,
    answeredQuestions: [],   // История на отговорите
    progress: 0,            // 0-100%
    isComplete: false,
    recommendation: null
  }),

  actions: {
    async startWizard(country) {
      const response = await axios.post('/wizard/start', { country });
      this.country = country;
      this.currentQuestion = response.data.firstQuestion;
    },

    async submitAnswer(answerId) {
      const response = await axios.post('/wizard/answer', {
        questionId: this.currentQuestion.id,
        answerId: answerId
      });

      this.answeredQuestions.push({
        question: this.currentQuestion,
        answerId: answerId
      });

      if (response.data.isComplete) {
        this.isComplete = true;
        this.recommendation = response.data.recommendation;
      } else {
        this.currentQuestion = response.data.nextQuestion;
        this.progress = response.data.progress;
      }
    }
  }
});
```

```

## Backend архитектура

### Маршрути (routes/web.php)

```

```php
Route::get('/', [HomeController::class, 'index'])->name('home');

// Wizard маршрути

```

```

Route::prefix('wizard')->middleware('throttle:60,1')->group(function ()
{
    Route::post('/start', [WizardController::class, 'start'])->
>name('wizard.start');
    Route::post('/answer', [WizardController::class, 'answer'])->
>name('wizard.answer');
    Route::get('/result', [WizardController::class, 'result'])->
>name('wizard.result');
});

// Автентикация (Laravel Breeze)
Route::middleware('auth')->group(function () {
    Route::get('/history', [HistoryController::class, 'index'])->
>name('history');
    Route::get('/profile', [ProfileController::class, 'edit'])->
>name('profile.edit');
});

require __DIR__.'/auth.php';
```

Middleware

```php
// app/Http/Middleware/SetLocale.php
class SetLocale
{
    public function handle($request, $next)
    {
        // Винаги български за текущата версия
        app()->setLocale('bg');
        return $next($request);
    }
}
```

Сесии

Конфигурация: `config/session.php`

```php
'driver' => env('SESSION_DRIVER', 'file'),
'lifetime' => 120, // минути
'expire_on_close' => false,
```

Съхранение на състоянието на wizard в сесия:

```php
session()->put('wizard.country', $country);
session()->put('wizard.allowedModels', $allowedModels);
session()->put('wizard.answeredQuestions', $answeredQuestions);
```

Локализация

Езикови файлове

```
```

```

```

resources/lang/
└─ bg/
 └─ auth.php # Съобщения за автентикация
 └─ validation.php # Валидационни съобщения
 └─ wizard.php # Текстове за wizard
...

Използване във Vue

```vue
<template>
  <div>{{ $t('wizard.welcome') }}</div>
</template>
```

Производителност

Кеширане

1. **Config cache**: `php artisan config:cache`
2. **Route cache**: `php artisan route:cache`
3. **View cache**: `php artisan view:cache`
4. **File session driver**: Бързо четене/запис за shared hosting среда

Оптимизации

- Lazy loading на Vue компоненти
- Eager loading на Eloquent relationships
- JSON файлове кешират се в паметта на DecisionTreeEngine

Забележка: Тази архитектура е проектирана да работи оптимално в shared hosting среда с ограничени ресурси, използвайки file-based кеширане и сесии вместо Redis или други external services.

```

## # Deployment ръководство за ARCHModels

Това ръководство описва процеса на деплой на ARCHModels в production среда, с фокус върху cPanel/shared hosting платформи.

### ## Съдържание

- [Production изисквания] (#production-изисквания)
- [Деплой в cPanel/Shared Hosting] (#деплой-в-сpanelshared-hosting)
- [Деплой във VPS среда] (#деплой-във-vps-среда-опционално)
- [SSL/TLS конфигурация] (#ssl-tls-конфигурация)
- [Health Checks] (#health-checks)
- [Troubleshooting] (#troubleshooting)

### ## Production изисквания

#### ### Минимални системни изисквания

| Компонент            | Версия | Забележка                                                                           |
|----------------------|--------|-------------------------------------------------------------------------------------|
| PHP                  | 8.2+   | С разширения: bcmath, ctype, json, mbstring, openssl, PDO, tokenizer, xml, curl, gd |
| MySQL                | 8.0+   | Или MariaDB 10.3+                                                                   |
| Дисково пространство | 1 GB   | Минимум за приложение + база данни                                                  |
| RAM                  | 512 MB | Минимум за shared hosting                                                           |
| Node.js              | 18+    | Само за build процес, не е необходим в production                                   |

#### ### PHP разширения

Проверете дали следните PHP разширения са активирани:

```
```bash
php -m | grep -E
'bcmath|ctype|json|mbstring|openssl|PDO|pdo_mysql|tokenizer|xml|curl|gd
|zip'
```
```

Ако някое разширение липсва, активирайте го в cPanel > Select PHP Version или помолете хостинг провайдъра.

### ## Деплой в cPanel/Shared Hosting

#### ### Стъпка 1: Подготовка на локалното приложение

```
```bash
# 1. Клонирайте или изтеглете проекта локално
git clone https://github.com/your-org/archmodels.git
cd archmodels/backend

# 2. Инсталирайте production зависимости (без dev)
composer install --no-dev --optimize-autoloader

# 3. Компилирайте frontend assets
npm install
npm run build

# 4. Създайте .env файл за production
cp .env.example .env.production
```
```

...

### Стъпка 2: Конфигуриране на .env.production

Редактирайте ``.env.production`` със следните настройки:

```
```bash
# Application
APP_NAME="ARCHModels"
APP_ENV=production
APP_KEY=base64:GENERATED_KEY_HERE # Ще генерираме по-късно
APP_DEBUG=false
APP_URL=https://your-domain.com

# Database (попълнете от cPanel > MySQL Databases)
DB_CONNECTION=mysql
DB_HOST=localhost # или IP от cPanel
DB_PORT=3306
DB_DATABASE=your_database_name
DB_USERNAME=your_database_user
DB_PASSWORD=your_database_password

# Cache & Session (file driver за shared hosting)
CACHE_DRIVER=file
SESSION_DRIVER=file
SESSION_LIFETIME=120

# Queue (sync за shared hosting без worker)
QUEUE_CONNECTION=sync

# Mail (опционално, ако има автентикация)
MAIL_MAILER=smtp
MAIL_HOST=mail.your-domain.com
MAIL_PORT=587
MAIL_USERNAME=noreply@your-domain.com
MAIL_PASSWORD=your_mail_password
MAIL_ENCRYPTION=tls
MAIL_FROM_ADDRESS=noreply@your-domain.com
MAIL_FROM_NAME="${APP_NAME}"

# Localization
APP_LOCALE=bg
APP_FALLBACK_LOCALE=bg
APP_TIMEZONE=Europe/Sofia
```
```

### Стъпка 3: Качване на файловете в cPanel

#### Чрез File Manager

1. Влезте в cPanel > File Manager
2. Навигирайте до ``public_html/`` (или поддиректория за subdomain)
3. Създайте директория структура:

...

```
public_html/
├── archmodels/ # Laravel root (извън public_html!)
│ ├── app/
│ └── bootstrap/
```

```

├── config/
├── database/
├── resources/
├── routes/
├── storage/
├── vendor/
├── .env
├── artisan
└── public/ # Симлинк към archmodels/public
...

```

**\*\*ВАЖНО\*\*:** Laravel приложението трябва да е **\*\*извън\*\*** public\_html за сигурност!

4. Качете всички файлове в `public\_html/../archmodels/` (една директория над public\_html)
5. Копирайте съдържанието на `public/` директорията в `public\_html/`

#### Чрез FTP/SFTP

```

...`bash
От локалната машина с lftp или FileZilla
Качете backend/ -> /home/username/archmodels/
Качете backend/public/* -> /home/username/public_html/
...

```

### Стъпка 4: Настройка на permissions

```

...`bash
Във File Manager или чрез SSH (ако имате достъп):
chmod -R 755 storage/
chmod -R 755 bootstrap/cache/
...

```

Ако нямате SSH достъп, използвайте cPanel File Manager:

1. Маркирайте `storage/` и `bootstrap/cache/`
2. Кликнете на "Permissions"
3. Задайте `755` за директории и `644` за файлове

### Стъпка 5: Редактиране на index.php

Редактирайте `public\_html/index.php` (или където са публичните файлове):

```

...`php
<?php

// Променете пътя към Laravel root
require __DIR__.'../archmodels/vendor/autoload.php';
$app = require_once __DIR__.'../archmodels/bootstrap/app.php';

// Останалата част остава същата...
...

```

### Стъпка 6: Настройка на база данни

1. **\*\*Създаване на база данни\*\*** (cPanel > MySQL Databases):
  - Database name: `username\_archmodels`
  - Database user: `username\_arch\_user`

- Password: генерирайте силна парола
- 2. **\*\*Добавяне на потребител към базата\*\***:
  - Assigned privileges: `ALL PRIVILEGES`
- 3. **\*\*Импорт на структурата\*\*** (cPanel > phpMyAdmin):
  - Няма нужда, ще използваме миграции (вижте по-долу)

### Стъпка 7: Изпълнение на Laravel команди

Тъй като в shared hosting няма пряк SSH достъп до `php artisan`, използвайте един от следните методи:

#### Метод 1: Чрез cPanel Terminal (ако е налично)

```
```bash
cd ~/archmodels
php artisan key:generate
php artisan migrate --force
php artisan db:seed --force
php artisan storage:link
php artisan config:cache
php artisan route:cache
php artisan view:cache
```
```

#### Метод 2: Чрез cron job (еднократно)

Създайте cron job в cPanel > Cron Jobs:

```
```bash
/usr/local/bin/php ~/archmodels/artisan migrate --force
```
```

След изпълнение, изтрийте този cron job.

#### Метод 3: Създайте временен PHP файл

Създайте `setup.php` в `public\_html/`:

```
```php
<?php
require __DIR__.'../../archmodels/vendor/autoload.php';
$app = require_once __DIR__.'../../archmodels/bootstrap/app.php';

$kernel = $app->make(Illuminate\Contracts\Console\Kernel::class);

// Изпълнете команди
$kernel->call('key:generate');
$kernel->call('migrate', ['--force' => true]);
$kernel->call('db:seed', ['--force' => true]);
$kernel->call('storage:link');
$kernel->call('config:cache');
$kernel->call('route:cache');
$kernel->call('view:cache');

echo "Setup completed!";

// ИЗТРИЙТЕ ТОЗИ ФАЙЛ СЛЕД ИЗПЪЛНЕНИЕ!
```
```

```

Посетете `https://your-domain.com/setup.php` и след това ****изтрийте файла****!

Стъпка 8: Конфигурация на .htaccess

Уверете се, че `public\_html/.htaccess` съдържа:

```
```apache
<IfModule mod_rewrite.c>
 <IfModule mod_negotiation.c>
 Options -MultiViews -Indexes
 </IfModule>

 RewriteEngine On

 # Redirect всички заявки към index.php
 RewriteCond %{REQUEST_FILENAME} !-d
 RewriteCond %{REQUEST_FILENAME} !-f
 RewriteRule ^ index.php [L]

 # Force HTTPS (след настройка на SSL)
 # RewriteCond %{HTTPS} off
 # RewriteRule ^(.*)$ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]
</IfModule>

Disable directory browsing
Options -Indexes

Защита на .env файла (ако случайно е в public/)
<FilesMatch "^\.env">
 Order allow,deny
 Deny from all
</FilesMatch>
```
```

Стъпка 9: Проверка на деплоя

Посетете `https://your-domain.com` и проверете дали:

1. Начална страница се зарежда коректно
2. Не се показват грешки
3. Можете да започнете wizard за България
4. Може да се регистрирате/влезете (ако има автентикация)

Деплой във VPS среда (опционално)

Ако използвате VPS (DigitalOcean, Linode, AWS EC2), следвайте тези стъпки:

Стъпка 1: Подготовка на сървър (Ubuntu 22.04)

```
```bash
Обновяване на системата
sudo apt update && sudo apt upgrade -y

Инсталация на необходими пакети
sudo apt install -y software-properties-common
```

```

Добавяне на PHP 8.2 repository
sudo add-apt-repository ppa:ondrej/php -y
sudo apt update

Инсталация на PHP 8.2 и разширения
sudo apt install -y php8.2-fpm php8.2-mysql php8.2-mbstring php8.2-xml
\
 php8.2-bcmath php8.2-curl php8.2-gd php8.2-zip php8.2-cli

Инсталация на Nginx
sudo apt install -y nginx

Инсталация на MySQL
sudo apt install -y mysql-server

Инсталация на Composer
curl -sS https://getcomposer.org/installer | php
sudo mv composer.phar /usr/local/bin/composer

Инсталация на Node.js (за build)
curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -
sudo apt install -y nodejs
```

### Стъпка 2: Конфигурация на MySQL

```
```bash
sudo mysql_secure_installation

# Влизане в MySQL
sudo mysql -u root -p

# Създаване на база данни и потребител
CREATE DATABASE archmodels CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci;
CREATE USER 'arch_user'@'localhost' IDENTIFIED BY
'secure_password_here';
GRANT ALL PRIVILEGES ON archmodels.* TO 'arch_user'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

Стъпка 3: Клонирание на проекта

```
```bash
Създаване на директория
sudo mkdir -p /var/www/archmodels
sudo chown -R $USER:$USER /var/www/archmodels

Клонирание
cd /var/www
git clone https://github.com/your-org/archmodels.git
cd archmodels/backend

Конфигуриране на .env
cp .env.example .env
nano .env # Редактирайте с правилните credentials

```

```
Инсталация на зависимости
composer install --no-dev --optimize-autoloader
npm install
npm run build
```

```
Laravel setup
php artisan key:generate
php artisan migrate --force
php artisan db:seed --force
php artisan storage:link
php artisan optimize
```
```

Стъпка 4: Конфигурация на Nginx

Създайте `/etc/nginx/sites-available/archmodels`:`

```
```nginx
server {
 listen 80;
 server_name your-domain.com www.your-domain.com;
 root /var/www/archmodels/backend/public;

 add_header X-Frame-Options "SAMEORIGIN";
 add_header X-Content-Type-Options "nosniff";

 index index.php;

 charset utf-8;

 location / {
 try_files $uri $uri/ /index.php?$query_string;
 }

 location = /favicon.ico { access_log off; log_not_found off; }
 location = /robots.txt { access_log off; log_not_found off; }

 error_page 404 /index.php;

 location ~ /\.php$ {
 fastcgi_pass unix:/var/run/php/php8.2-fpm.sock;
 fastcgi_param SCRIPT_FILENAME
$realpath_root$fastcgi_script_name;
 include fastcgi_params;
 }

 location ~ /\.(!well-known).* {
 deny all;
 }
}
```
```

Активиране на сайта:

```
```bash
sudo ln -s /etc/nginx/sites-available/archmodels /etc/nginx/sites-
enabled/
sudo nginx -t
sudo systemctl reload nginx
```

```

Стъпка 5: Permissions

```
```bash
sudo chown -R www-data:www-data /var/www/archmodels/backend/storage
sudo chown -R www-data:www-data
/var/www/archmodels/backend/bootstrap/cache
sudo chmod -R 775 /var/www/archmodels/backend/storage
sudo chmod -R 775 /var/www/archmodels/backend/bootstrap/cache
```
```

SSL/TLS конфигурация

С Let's Encrypt (безплатен SSL)

За cPanel:

1. Отидете на cPanel > SSL/TLS Status
2. Кликнете на "Run AutoSSL"
3. Изчакайте завършване на процеса

За VPS с Certbot:

```
```bash
Инсталация на Certbot
sudo apt install -y certbot python3-certbot-nginx

Генериране на сертификат
sudo certbot --nginx -d your-domain.com -d www.your-domain.com

Автоматично обновяване
sudo certbot renew --dry-run
```
```

Force HTTPS

След настройка на SSL, редактирайте `.htaccess` (cPanel) или Nginx config (VPS):

За .htaccess (cPanel):

```
```apache
Uncomment следните редове
RewriteCond %{HTTPS} off
RewriteRule ^(.*)$ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]
```
```

За Nginx (VPS):

```
```nginx
server {
 listen 80;
 server_name your-domain.com www.your-domain.com;
 return 301 https://$server_name$request_uri;
}

server {
 listen 443 ssl http2;
```

```

server_name your-domain.com www.your-domain.com;

 ssl_certificate /etc/letsencrypt/live/your-
domain.com/fullchain.pem;
 ssl_certificate_key /etc/letsencrypt/live/your-
domain.com/privkey.pem;

 # Останалата конфигурация...
}
...

Health Checks

Създаване на health check endpoint

Създайте `routes/web.php`:

```php
Route::get('/health', function () {
    try {
        // Проверка на база данни
        DB::connection()->getPdo();

        // Проверка на storage директория
        $writable = is_writable(storage_path('app'));

        return response()->json([
            'status' => 'healthy',
            'database' => 'connected',
            'storage' => $writable ? 'writable' : 'not writable',
            'timestamp' => now()->toIso8601String(),
        ], 200);
    } catch (\Exception $e) {
        return response()->json([
            'status' => 'unhealthy',
            'error' => $e->getMessage(),
        ], 500);
    }
});
```

Проверка на здравето

```bash
curl https://your-domain.com/health
```

Очакван резултат:
```json
{
  "status": "healthy",
  "database": "connected",
  "storage": "writable",
  "timestamp": "2026-01-31T12:00:00+00:00"
}
```

Troubleshooting

```

### Проблем: 500 Internal Server Error

\*\*Решение\*\*:

1. Проверете дали ``.env`` файлът е създаден и попълнен
2. Проверете дали ``APP_KEY`` е генериран
3. Проверете permissions на ``storage/`` и ``bootstrap/cache/``
4. Проверете error logs:
  - cPanel: ``~/archmodels/storage/logs/laravel.log``
  - VPS: ``/var/www/archmodels/backend/storage/logs/laravel.log``

### Проблем: Не се зареждат CSS/JS файлове

\*\*Решение\*\*:

1. Проверете дали сте стартирали ``npm run build``
2. Проверете дали ``public/build/`` директорията съществува
3. В ``.env`` проверете ``APP_URL`` да съвпада с домейна
4. Изчистете кеша: ``php artisan optimize:clear``

### Проблем: Database connection error

\*\*Решение\*\*:

1. Проверете credentials в ``.env``
2. Проверете дали потребителят има права на базата
3. Проверете дали MySQL сървърът работи:
  - ```bash
  - # cPanel: чрез cPanel > MySQL Databases
  - # VPS:
  - sudo systemctl status mysql
  - ```

### Проблем: Storage symlink не работи

\*\*Решение\*\* (за cPanel):

Вместо ``php artisan storage:link``, създайте ръчно symlink:

1. File Manager > public\_html/
2. Създайте директория ``storage``
3. Копирайте съдържанието на ``../archmodels/storage/app/public/`` в ``storage/``

\*\*За VPS\*\*:

```
```bash
php artisan storage:link
```
```

### Проблем: Permission denied грешки

\*\*Решение\*\*:

```
```bash
# cPanel File Manager:
# Задайте 755 на storage/ и bootstrap/cache/

# VPS:
sudo chown -R www-data:www-data storage bootstrap/cache
```

```
sudo chmod -R 775 storage bootstrap/cache  
\\`
```

Контролен списък за деплой

- [] PHP 8.2+ с всички необходими разширения
- [] MySQL база данни създадена
- [] ``.env`` файл конфигуриран
- [] ``composer install --no-dev`` изпълнен
- [] ``npm run build`` изпълнен
- [] ``php artisan key:generate`` изпълнен
- [] ``php artisan migrate --force`` изпълнен
- [] ``php artisan storage:link`` изпълнен
- [] Permissions на `storage/` и `bootstrap/cache/` зададени правилно
- [] SSL сертификат инсталиран
- [] Force HTTPS активиран
- [] Health check endpoint работи
- [] `APP_DEBUG=false` в production
- [] `APP_ENV=production`

****Забележка**:** Този документ предполага използване на cPanel/shared hosting като основна target среда. За по-сложни deployments (Docker, Kubernetes, CI/CD) се препоръчва консултация с DevOps специалист.

Конфигурация на ARCHModels

Този документ описва всички конфигурационни параметри, environment variables и настройки на ARCHModels приложението.

Съдържание

- [Environment Variables (.env)](#environment-variables-env)
- [Laravel конфигурационни файлове](#laravel-конфигурационни-файлове)
- [Vite конфигурация](#vite-конфигурация)
- [Примерни .env файлове](#примерни-env-файлове)

Environment Variables (.env)

Основни настройки на приложението

| Променлива | Стойност | Описание |
|-------------|------------------------|--|
| `APP_NAME` | `ARCHModels` | Име на приложението, използва се в заглавия и имейли |
| `APP_ENV` | `local` / `production` | Среда на приложението |
| `APP_KEY` | `base64:...` | Ключ за криптиране (генерира се с `php artisan key:generate`) |
| `APP_DEBUG` | `true` / `false` | **ЗАДЪЛЖИТЕЛНО false в production!**
Показва детайлни грешки |
| `APP_URL` | `http://localhost` | Base URL на приложението |

Пример:

```
```bash
APP_NAME="ARCHModels"
APP_ENV=production
APP_KEY=base64:kZXJ0aW9uIGtleSBnZW5lcmF0ZWQgYnkgTGFiYXZlbA==
APP_DEBUG=false
APP_URL=https://archmodels.example
```
```

Важно:

- `APP\_KEY` се генерира автоматично с `php artisan key:generate`
- ****Никога**** не споделяйте `APP\_KEY` публично
- `APP\_DEBUG=true` в production е ****сериозен**** security риск

База данни

| Променлива | Стойност | Описание |
|-----------------|--------------|--|
| `DB_CONNECTION` | `mysql` | Database драйвер (mysql, pgsql, sqlite) |
| `DB_HOST` | `127.0.0.1` | Хост на database сървър |
| `DB_PORT` | `3306` | Порт на MySQL (3306) или PostgreSQL (5432) |
| `DB_DATABASE` | `archmodels` | Име на базата данни |
| `DB_USERNAME` | `root` | Потребител за достъп до базата |
| `DB_PASSWORD` | `password` | Парола за database потребител |

Пример за production:

```
```bash
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=username_archmodels
DB_USERNAME=username_arch_user
```
```

```
DB_PASSWORD=veryStrongPassword123!
```

```
```\n
```

```
Забележка за cPanel:
```

- Database names обикновено имат префикс: `username\_archmodels`
- Потребителски имена също: `username\_arch\_user`
- Проверете точното име в cPanel > MySQL Databases

```

```

```
Cache & Session
```

```
| Променлива | Стойност | Описание |
|-----|-----|-----|
| `CACHE_DRIVER` | `file` / `redis` / `database` | Драйвер за кеширане |
| `SESSION_DRIVER` | `file` / `redis` / `database` | Драйвер за сесии |
| `SESSION_LIFETIME` | `120` | Време на живот на сесия (в минути) |
| `SESSION_ENCRYPT` | `false` | Криптиране на session data |
| `SESSION_DOMAIN` | `null` | Domain за session cookie |
| `SESSION_SECURE_COOKIE` | `true` / `false` | Използва HTTPS за cookies
(true за production с SSL) |
```

```
За shared hosting (препоръчително):
```

```
```bash
```

```
CACHE_DRIVER=file
```

```
SESSION_DRIVER=file
```

```
SESSION_LIFETIME=120
```

```
SESSION_SECURE_COOKIE=true # Само ако имате SSL
```

```
```\n
```

```
За VPS с Redis (опционално):
```

```
```bash
```

```
CACHE_DRIVER=redis
```

```
SESSION_DRIVER=redis
```

```
SESSION_LIFETIME=120
```

```
REDIS_HOST=127.0.0.1
```

```
REDIS_PASSWORD=null
```

```
REDIS_PORT=6379
```

```
```\n
```

```

```

```
Queue (опашки за задачи)
```

```
| Променлива | Стойност | Описание |
|-----|-----|-----|
| `QUEUE_CONNECTION` | `sync` / `database` / `redis` | Драйвер за опашки |
```

```
За shared hosting:
```

```
```bash
```

```
QUEUE_CONNECTION=sync # Синхронно изпълнение (без worker)
```

```
```\n
```

```
За VPS (ако добавите queue worker):
```

```
```bash
```

```
QUEUE_CONNECTION=database # или redis
```

```
```\n
```

```
Забележка: ARCHModels не използва опашки понастоящем, `sync` е подходящо.
```

```

Mail (за автентикация)

| Променлива | Стойност | Описание |
|-----|-----|-----|
| `MAIL_MAILER` | `smtp` / `log` | Mailer драйвер |
| `MAIL_HOST` | `mail.example.com` | SMTP хост |
| `MAIL_PORT` | `587` / `465` | SMTP порт (587 за TLS, 465 за SSL) |
| `MAIL_USERNAME` | `noreply@example.com` | SMTP username |
| `MAIL_PASSWORD` | `password` | SMTP парола |
| `MAIL_ENCRYPTION` | `tls` / `ssl` | Encryption метод |
| `MAIL_FROM_ADDRESS` | `noreply@example.com` | Адрес на изпращач |
| `MAIL_FROM_NAME` | `ARCHModels` | Име на изпращач |

За production (с Laravel Breeze):
```bash
MAIL_MAILER=smtp
MAIL_HOST=mail.your-domain.com
MAIL_PORT=587
MAIL_USERNAME=noreply@your-domain.com
MAIL_PASSWORD=your_smtp_password
MAIL_ENCRYPTION=tls
MAIL_FROM_ADDRESS=noreply@your-domain.com
MAIL_FROM_NAME="${APP_NAME}"
```

За development (без изпращане на имейли):
```bash
MAIL_MAILER=log # Записва имейлите в storage/logs/laravel.log
```

Локализация

| Променлива | Стойност | Описание |
|-----|-----|-----|
| `APP_LOCALE` | `bg` | Основен език на приложението |
| `APP_FALLBACK_LOCALE` | `bg` | Резервен език |
| `APP_TIMEZONE` | `Europe/Sofia` | Часова зона |

Пример:
```bash
APP_LOCALE=bg
APP_FALLBACK_LOCALE=bg
APP_TIMEZONE=Europe/Sofia
```

Поддържани локали (бъдещи версии):
- `bg` - Български
- `de` - Немски (планиран)
- `en` - Английски (планиран)

Logging

| Променлива | Стойност | Описание |
|-----|-----|-----|
| `LOG_CHANNEL` | `stack` / `daily` / `single` | Канал за логове |
| `LOG_LEVEL` | `debug` / `info` / `error` | Ниво на логване |

```

```

| `LOG_DEPRECATIONS_CHANNEL` | `null` | Канал за deprecation warnings |

3a production:
```bash
LOG_CHANNEL=daily
LOG_LEVEL=error # Само грешки в production
LOG_DEPRECATIONS_CHANNEL=null
```

3a development:
```bash
LOG_CHANNEL=stack
LOG_LEVEL=debug
```

Broadcasting & Pusher (не се използват)

ARCHModels не използва real-time broadcasting, можете да оставите default стойности:

```bash
BROADCAST_DRIVER=log
```

Други променливи

| Променлива | Стойност | Описание |
|-----|-----|-----|
| `FILESYSTEM_DISK` | `local` / `s3` | Storage драйвер |
| `AWS_ACCESS_KEY_ID` | | AWS ключ (при S3 storage) |
| `AWS_SECRET_ACCESS_KEY` | | AWS secret (при S3 storage) |
| `AWS_DEFAULT_REGION` | `us-east-1` | AWS регион |
| `AWS_BUCKET` | | S3 bucket име |

3a shared hosting (локално хранилище):
```bash
FILESYSTEM_DISK=local
```

Laravel конфигурационни файлове

config/app.php

Основна конфигурация на приложението.

Ключови настройки:

```php
return [
    // Име на приложението
    'name' => env('APP_NAME', 'ARCHModels'),

    // Среда
    'env' => env('APP_ENV', 'production'),

    // Debug режим (ВИНАГИ false в production!)

```

```

'debug' => (bool) env('APP_DEBUG', false),

// URL на приложението
'url' => env('APP_URL', 'http://localhost'),

// Часова зона
'timezone' => env('APP_TIMEZONE', 'Europe/Sofia'),

// Локализация
'locale' => env('APP_LOCALE', 'bg'),
'fallback_locale' => env('APP_FALLBACK_LOCALE', 'bg'),

// Service providers
'providers' => [
    // Laravel service providers
    Illuminate\Auth\AuthServiceProvider::class,
    Illuminate\Broadcasting\BroadcastServiceProvider::class,
    // ...

    // Application service providers
    App\Providers\AppServiceProvider::class,
    App\Providers\RouteServiceProvider::class,
],

// Class aliases
'aliases' => [
    'App' => Illuminate\Support\Facades\App::class,
    'Artisan' => Illuminate\Support\Facades\Artisan::class,
    // ...
],
];
```

config/database.php

Конфигурация на базата данни.

Важни настройки за MySQL:
```

```

```php
'connections' => [
    'mysql' => [
        'driver' => 'mysql',
        'host' => env('DB_HOST', '127.0.0.1'),
        'port' => env('DB_PORT', '3306'),
        'database' => env('DB_DATABASE', 'archmodels'),
        'username' => env('DB_USERNAME', 'root'),
        'password' => env('DB_PASSWORD', ''),
        'charset' => 'utf8mb4',
        'collation' => 'utf8mb4_unicode_ci',
        'prefix' => '',
        'strict' => true,
        'engine' => 'InnoDB',

        // Connection options
        'options' => extension_loaded('pdo_mysql') ? array_filter([
            PDO::MYSQL_ATTR_SSL_CA => env('MYSQL_ATTR_SSL_CA'),
        ]) : [],
    ],
],
```

```

```

...

Забележка: `charset` и `collation` са настроени за правилна поддръжка
на български език.

config/session.php

Конфигурация на сесии.

За shared hosting:

```php
return [
    'driver' => env('SESSION_DRIVER', 'file'),
    'lifetime' => env('SESSION_LIFETIME', 120),
    'expire_on_close' => false,
    'encrypt' => false,
    'files' => storage_path('framework/sessions'),
    'cookie' => env(
        'SESSION_COOKIE',
        Str::slug(env('APP_NAME', 'laravel'), '_').'_session'
    ),
    'path' => '/',
    'domain' => env('SESSION_DOMAIN'),
    'secure' => env('SESSION_SECURE_COOKIE', true), // true за HTTPS
    'http_only' => true,
    'same_site' => 'lax',
];
```

config/cache.php

Конфигурация на кеширане.

За shared hosting:

```php
return [
    'default' => env('CACHE_DRIVER', 'file'),

    'stores' => [
        'file' => [
            'driver' => 'file',
            'path' => storage_path('framework/cache/data'),
        ],

        'database' => [
            'driver' => 'database',
            'table' => 'cache',
            'connection' => null,
            'lock_connection' => null,
        ],
    ],
];
```

За VPS с Redis (опционално):

```

```

```php
'stores' => [
    'redis' => [
        'driver' => 'redis',
        'connection' => 'cache',
        'lock_connection' => 'default',
    ],
],
```

```

---

### config/inertia.php

Конфигурация на Inertia.js.

**\*\*Стандартна конфигурация\*\*:**

```

```php
return [
    /*
     * Inertia SSR configuration
     */
    'ssr' => [
        'enabled' => false, // За shared hosting оставете false
        'url' => 'http://127.0.0.1:13714',
    ],

    /*
     * Shared data (достъпни във всички Vue страници)
     */
    'shared' => [
        'auth' => function () {
            return [
                'user' => Auth::user(),
            ];
        },
        'flash' => function () {
            return [
                'success' => session('success'),
                'error' => session('error'),
            ];
        },
    ],
];
```

```

---

### config/cors.php

Cross-Origin Resource Sharing (CORS) конфигурация.

**\*\*За Inertia.js SPA\*\* (стандартна конфигурация):**

```

```php
return [
    'paths' => ['api/*', 'sanctum/csrf-cookie'],
    'allowed_methods' => ['*'],
    'allowed_origins' => [env('APP_URL')],
    'allowed_origins_patterns' => [],
    'allowed_headers' => ['*'],
];

```

```

        'exposed_headers' => [],
        'max_age' => 0,
        'supports_credentials' => true,
    ];
    ...

---

## Vite конфигурация

### vite.config.js

Конфигурация на Vite за компилиране на frontend assets.

```javascript
import { defineConfig } from 'vite';
import laravel from 'laravel-vite-plugin';
import vue from '@vitejs/plugin-vue';

export default defineConfig({
 plugins: [
 laravel({
 input: ['resources/css/app.css', 'resources/js/app.js'],
 refresh: true,
 }),
 vue({
 template: {
 transformAssetUrls: {
 base: null,
 includeAbsolute: false,
 },
 },
 }),
],
 resolve: {
 alias: {
 '@': '/resources/js',
 },
 },
 server: {
 host: '0.0.0.0',
 port: 5173,
 hmr: {
 host: 'localhost',
 },
 },
});
...

```

**\*\*Компилиране за production\*\*:**

```

```bash
npm run build
...

```

Това генерира оптимизирани файлове в `public/build/`.

Примерни .env файлове

.env.local (Development)

```

```bash
APP_NAME="ARCHModels"
APP_ENV=local
APP_KEY=base64:GENERATED_KEY_HERE
APP_DEBUG=true
APP_URL=http://localhost:8000

LOG_CHANNEL=stack
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=archmodels_local
DB_USERNAME=root
DB_PASSWORD=

CACHE_DRIVER=file
SESSION_DRIVER=file
SESSION_LIFETIME=120

QUEUE_CONNECTION=sync

MAIL_MAILER=log

APP_LOCALE=bg
APP_FALLBACK_LOCALE=bg
APP_TIMEZONE=Europe/Sofia

FILESYSTEM_DISK=local
```

---

### .env.production (Production на cPanel)

```bash
APP_NAME="ARCHModels"
APP_ENV=production
APP_KEY=base64:GENERATED_KEY_HERE
APP_DEBUG=false
APP_URL=https://archmodels.example

LOG_CHANNEL=daily
LOG_LEVEL=error

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=username_archmodels
DB_USERNAME=username_arch_user
DB_PASSWORD=veryStrongPassword123!

CACHE_DRIVER=file
SESSION_DRIVER=file
SESSION_LIFETIME=120
SESSION_SECURE_COOKIE=true

QUEUE_CONNECTION=sync

MAIL_MAILER=smtp

```

```

MAIL_HOST=mail.your-domain.com
MAIL_PORT=587
MAIL_USERNAME=noreply@your-domain.com
MAIL_PASSWORD=your_smtp_password
MAIL_ENCRYPTION=tls
MAIL_FROM_ADDRESS=noreply@your-domain.com
MAIL_FROM_NAME="${APP_NAME}"

APP_LOCALE=bg
APP_FALLBACK_LOCALE=bg
APP_TIMEZONE=Europe/Sofia

FILESYSTEM_DISK=local
```

---

## Валидация на конфигурацията

### Проверка на .env файла

```bash
Проверка на database connection
php artisan tinker
>>> DB::connection()->getPdo();

Проверка на APP_KEY
php artisan key:generate --show

Проверка на mail конфигурация
php artisan tinker
>>> Mail::raw('Test', function($m) { $m->to('test@example.com')-
>subject('Test'); });
```

### Изчистване на кеширана конфигурация

След промяна на .env или config файлове:

```bash
php artisan config:clear
php artisan cache:clear
php artisan route:clear
php artisan view:clear

За production оптимизация след промени:
php artisan config:cache
php artisan route:cache
php artisan view:cache
```

---

## Сигурност на конфигурацията

### Контролен списък за сигурност



- [ ] `APP_DEBUG=false` в production
- [ ] `APP_ENV=production` в production
- [ ] Силна `APP_KEY` (никога не споделяйте)
- [ ] Силна database парола
- [ ] `.env` файлът **не е** в git repository (добавен в `.gitignore`)

```

- [] `SESSION\_SECURE\_COOKIE=true` при използване на HTTPS
- [] SMTP credentials са сигурни
- [] Регулярна смяна на credentials

Защита на .env файла

В `.gitignore`:

```
`.env
.env.backup
.env.production
.env.*.php`
```

В `public/.htaccess`:

```
``apache
<FilesMatch "^.env">
    Order allow,deny
    Deny from all
</FilesMatch>``
```

****Забележка**:** При промяна на конфигурацията винаги тествайте приложението преди деплой в production. Използвайте staging среда, ако е възможно.

Operations ръководство за ARCHModels

Това ръководство описва оперативното управление на ARCHModels в production среда, включително мониторинг, логове, troubleshooting и поддръжка.

Съдържание

- [Мониторинг] (#мониторинг)
- [Logging (Логове)] (#logging-логове)
- [Performance метрики] (#performance-метрики)
- [Troubleshooting] (#troubleshooting)
- [Поддръжка] (#поддръжка)
- [Често срещани проблеми] (#често-срещани-проблеми)

Мониторинг

Ключови метрики за наблюдение

| Метрика | Цел | Критичен праг | Инструмент |
|----------------------|--------------|---------------|---------------|
| Response Time | < 500ms | > 2s | Logs, APM |
| Error Rate | < 0.5% | > 5% | Laravel Logs |
| CPU Usage | < 70% | > 90% | cPanel / htop |
| RAM Usage | < 80% | > 95% | cPanel / free |
| Disk Usage | < 80% | > 90% | df -h |
| Database Connections | < 80% от max | 100% | MySQL status |

Health Check Endpoint

ARCHModels предоставя `/health` endpoint за проверка на здравето на системата.

****Използване**:**

```
```bash
curl https://your-domain.com/health
```
```

****Очакван отговор (200 OK)**:**

```
```json
{
 "status": "healthy",
 "database": "connected",
 "storage": "writable",
 "timestamp": "2026-01-31T12:00:00+00:00"
}
```
```

****Грешен отговор (500 Internal Server Error)**:**

```
```json
{
 "status": "unhealthy",
 "error": "SQLSTATE[HY000] [2002] Connection refused"
}
```
```

Мониторинг с Uptime Services

Препоръчани услуги за uptime monitoring:

1. ****UptimeRobot**** (безплатен)
 - Добавете `/health` endpoint`
 - Интервал: 5 минути
 - Alert при `down time > 2 минути`
2. ****Pingdom**** (платен)
 - По-подробни метрики
 - Real user monitoring

****Настройка**:**
```

URL: `https://your-domain.com/health`  
Interval: 5 minutes  
Keyword to match: "healthy"  
Alert contacts: `admin@your-domain.com`  
```

Application Performance Monitoring (APM)

За shared hosting, APM инструментите са ограничени. Опции:

1. ****Laravel Telescope**** (само за development!)
 - Не използвайте в production (performance overhead)
2. ****Custom logging**** (препоръчително за shared hosting)
 - Логване на бавни заявки
 - Логване на грешки

****Пример за custom performance logging**:**

```
```php
// app/Http/Middleware/LogPerformance.php
class LogPerformance
{
 public function handle($request, $next)
 {
 $start = microtime(true);
 $response = $next($request);
 $duration = microtime(true) - $start;

 if ($duration > 1.0) { // Повече от 1 секунда
 Log::warning('Slow request', [
 'url' => $request->fullUrl(),
 'method' => $request->method(),
 'duration' => round($duration, 2),
 'memory' => memory_get_peak_usage(true) / 1024 / 1024 .
 ' MB',
]);
 }

 return $response;
 }
}
```
```

Logging (Логове)

```

### Конфигурация на логове

**Production config** (`.env`):
```bash
LOG_CHANNEL=daily
LOG_LEVEL=error # Само грешки
```

**Development config**:
```bash
LOG_CHANNEL=stack
LOG_LEVEL=debug # Всички нива
```

### Местоположение на логове

| Среда | Път |
|-----|-----|
| cPanel | `~/archmodels/storage/logs/laravel.log` |
| VPS | `/var/www/archmodels/backend/storage/logs/laravel.log` |

### Формат на лог запис

```
[2026-01-31 12:00:00] production.ERROR: SQLSTATE[HY000] [2002]
Connection refused
{
 "exception": "PDOException",
 "file": "/path/to/file.php",
 "line": 123,
 "trace": [...]
}
```

### Нива на логване

| Ниво | Употреба | Пример |
|-----|-----|-----|
| **emergency** | Системата е недостъпна | Database corruption |
| **alert** | Изисква се незабавна реакция | Filesystem пълен |
| **critical** | Критична грешка | Невъзможност за запис в database |
| **error** | Runtime грешки | Database connection failed |
| **warning** | Предупреждения | Бавна заявка |
| **notice** | Нормални, но значими събития | User login |
| **info** | Информационни съобщения | Wizard completed |
| **debug** | Debug информация | Variable values |

### Използване на логове в кода

```php
use Illuminate\Support\Facades\Log;

// Error logging
try {
 DB::connection()->getPdo();
} catch (\Exception $e) {
 Log::error('Database connection failed', [
 'error' => $e->getMessage(),
 'trace' => $e->getTraceAsString(),
]);
}
```

```

```

    });
}

// Info logging
Log::info('User completed wizard', [
    'user_id' => auth()->id(),
    'country' => $country,
    'model' => $recommendedModel,
]);

// Warning logging
if ($duration > 2) {
    Log::warning('Slow decision tree processing', [
        'duration' => $duration,
        'country' => $country,
    ]);
}
...

### Анализ на логове

#### cPanel File Manager

1. Отидете на File Manager
2. Навигирайте до `archmodels/storage/logs/`
3. Отворете `laravel-YYYY-MM-DD.log`
4. Търсете грешки с `CTRL+F` за "ERROR"

#### SSH/Terminal

```bash
Последни 50 реда
tail -n 50 storage/logs/laravel.log

Следене на логове в реално време
tail -f storage/logs/laravel.log

Търсене на грешки
grep "ERROR" storage/logs/laravel.log

Преброяване на грешки за деня
grep -c "ERROR" storage/logs/laravel-$(date +%Y-%m-%d).log

Най-чести грешки
grep "ERROR" storage/logs/laravel.log | sort | uniq -c | sort -rn |
head -10
```

### Ротация на логове

Laravel автоматично ротира логовете при `LOG_CHANNEL=daily`.

**Почистване на стари логове**:
```

```

```bash
Изтриване на логове по-стари от 30 дни
find storage/logs -name "laravel-*.log" -mtime +30 -delete

Архивиране на логове

```

```

tar -czf logs-backup-$(date +%Y%m%d).tar.gz storage/logs/
```

**Cron job за автоматично почистване** (cPanel):

```bash
Всеки ден в 2:00 AM
0 2 * * * find ~/archmodels/storage/logs -name "laravel-*.log" -mtime
+30 -delete
```

## Performance метрики

### Database заявки

**Проверка на бавни заявки**:
```

```

```sql
-- MySQL slow query log
SET GLOBAL slow_query_log = 'ON';
SET GLOBAL long_query_time = 2; -- 2 секунди

-- Показване на running queries
SHOW FULL PROCESSLIST;

-- Database size
SELECT
 table_schema AS 'Database',
 ROUND(SUM(data_length + index_length) / 1024 / 1024, 2) AS 'Size
(MB)'
FROM information_schema.tables
GROUP BY table_schema;
```

**Оптимизация на таблици**:
```

```

```bash
php artisan tinker
>>> DB::statement('OPTIMIZE TABLE questions, answers, decision_nodes,
users');
```

### File storage метрики

```bash
Общ размер на storage
du -sh storage/

Breakdown по директории
du -h storage/ | sort -h

Най-големи файлове
find storage/ -type f -exec du -h {} + | sort -rh | head -20
```

### Memory usage

**Мониторинг в PHP**:
```

```

```php
// В контролер или middleware
$memoryUsage = memory_get_peak_usage(true) / 1024 / 1024;
Log::info("Memory usage: {$memoryUsage} MB");

// При превишаване на лимит
if ($memoryUsage > 100) {
 Log::warning('High memory usage detected', [
 'memory_mb' => $memoryUsage,
]);
}
```

## Troubleshooting

### Диагностични команди

```bash
Проверка на PHP версия
php -v

Проверка на PHP разширения
php -m

Проверка на Laravel конфигурация
php artisan config:show

Проверка на routes
php artisan route:list

Проверка на database connection
php artisan tinker
>>> DB::connection()->getPdo();

Проверка на permissions
ls -la storage/
ls -la bootstrap/cache/
```

### Изчистване на кеш

```bash
Изчистване на всички кешове
php artisan optimize:clear

Индивидуално
php artisan cache:clear # Application cache
php artisan config:clear # Config cache
php artisan route:clear # Route cache
php artisan view:clear # Compiled views

Production оптимизация (след деплой)
php artisan config:cache
php artisan route:cache
php artisan view:cache
```

### Проверка на database

```

```

```bash
Проверка на миграции
php artisan migrate:status

Проверка на database schema
php artisan db:show

Изпълнение на query за test
php artisan tinker
>>> \App\Models\Question::count();
>>> \App\Models\User::first();
```

## Поддръжка

### Редовни задачи

| Задача | Честота | Команда/Действие |
|-----|-----|-----|
| Backup на база данни | Дневно | Вижте [BACKUP_DISASTER_RECOVERY.md] (BACKUP_DISASTER_RECOVERY.md) |
| Проверка на логове | Седмично | `grep "ERROR" storage/logs/*.log` |
| Почистване на стари логове | Месечно | `find storage/logs -mtime +30 -delete` |
| Обновяване на dependencies | Тримесечно | `composer update && npm update` |
| Security audit | Месечно | `composer audit && npm audit` |
| Database optimization | Месечно | `OPTIMIZE TABLE ...` |

### Composer dependencies update

```bash
Проверка за outdated пакети
composer outdated

Актуализиране (само patch versions)
composer update --with-dependencies

Актуализиране с minor versions
composer update

След update:
php artisan optimize:clear
```

### NPM dependencies update

```bash
Проверка за vulnerabilities
npm audit

Fix vulnerabilities
npm audit fix

Update пакети
npm update

Rebuild assets
npm run build

```

```

` ``
Database maintenance

` `` bash
Изпълнение през cPanel > phpMyAdmin или SSH

Analyze tables
ANALYZE TABLE questions, answers, decision_nodes, users;

Optimize tables
OPTIMIZE TABLE questions, answers, decision_nodes, users;

Check tables for errors
CHECK TABLE questions, answers, decision_nodes, users;

Repair tables (ако има грешки)
REPAIR TABLE table_name;
` ``

Често срещани проблеми

Проблем 1: 500 Internal Server Error

Причини:
- `.env` файл липсва или невалиден
- `APP_KEY` не е генериран
- Permission грешки на `storage/` или `bootstrap/cache/`
- Database connection грешка

Решение:
` `` bash
1. Проверете error log
tail -50 storage/logs/laravel.log

2. Проверете .env
cat .env | grep APP_KEY

3. Ако APP_KEY липсва:
php artisan key:generate

4. Проверете permissions
chmod -R 775 storage bootstrap/cache

5. Изчистете кеша
php artisan optimize:clear
` ``

Проблем 2: Бавна производителност

Симптоми: Страници се зареждат > 3 секунди

Решение:
` `` bash
1. Активирайте config/route cache
php artisan config:cache
php artisan route:cache

```

```

2. Проверете бавни database заявки
Вижте MySQL slow query log

3. Проверете дисково пространство
df -h

4. Optimize database
php artisan tinker
>>> DB::statement('OPTIMIZE TABLE questions, answers, decision_nodes');
```

---

### Проблем 3: Database connection refused

**Симптоми**: `SQLSTATE[HY000] [2002] Connection refused`

**Решение**:
```bash
1. Проверете MySQL статус (VPS)
sudo systemctl status mysql

2. Проверете .env credentials
cat .env | grep DB_

3. Тествайте connection от terminal
mysql -h 127.0.0.1 -u username -p database_name

4. Проверете MySQL е достъпен
telnet 127.0.0.1 3306
```

---

### Проблем 4: Session/Auth не работи

**Симптоми**: Потребителят не може да се впише или излиза автоматично

**Решение**:
```bash
1. Изчистете session cache
php artisan cache:clear
rm -rf storage/framework/sessions/*

2. Проверете SESSION_DOMAIN в .env
Трябва да е празен или правилният domain

3. Проверете SESSION_SECURE_COOKIE
Трябва да е true само при HTTPS

4. Проверете permissions на storage/framework/sessions
chmod -R 775 storage/framework/sessions
```

---

### Проблем 5: CSS/JS файлове не се зареждат

```

```

**Симптоми**: Страницата е без стилове

**Решение**:
```bash
1. Rebuild assets
npm run build

2. Проверете public/build/ директория
ls -la public/build/

3. Проверете APP_URL в .env
Трябва да съвпада с домейна

4. Изчистете browser cache
CTRL+F5 или Hard refresh
```

---

### Проблем 6: CSRF token mismatch

**Симптоми**: `419 Page Expired` при submit на формуляр

**Решение**:
```bash
1. Проверете SESSION_DOMAIN
Трябва да е правилен или null

2. Изчистете cache
php artisan cache:clear

3. Проверете дали session cookie се записва
Отворете DevTools > Application > Cookies

4. Проверете SESSION_SECURE_COOKIE
false за HTTP, true за HTTPS
```

---

### Аларми и известия

### Email известия при грешки

**Настройка в** `app/Exceptions/Handler.php`:
```php
```php
use Illuminate\Support\Facades\Mail;

public function report(Throwable $exception)
{
    if ($this->shouldReport($exception) && app()-
>environment('production')) {
        $this->sendEmailNotification($exception);
    }

    parent::report($exception);
}
```

```

```

protected function sendEmailNotification(Throwable $exception)
{
 Mail::raw(
 "Error: {$exception->getMessage()}\n\n" .
 "File: {$exception->getFile()}:{$exception->getLine()}\n\n" .
 "Trace:\n{$exception->getTraceAsString()}",
 function ($message) {
 $message->to('admin@your-domain.com')
 ->subject('[ARCHModels] Production Error');
 }
);
}
...

Контакти за поддръжка

- **Technical Lead**: tech@your-domain.com
- **DevOps**: devops@your-domain.com
- **Emergency**: +359 XXX XXX XXX

Забележка: Този документ трябва да се актуализира при промени в
архитектурата или добавяне на нови функционалности.

```

## # Security ръководство за ARCHModels

Този документ описва security практики, управление на секрети и защита на ARCHModels приложението.

### ## Съдържание

- [OWASP Top 10 защита] (#owasp-top-10-защита)
- [Управление на секрети] (#управление-на-секрети)
- [Автентикация и авторизация] (#автентикация-и-авторизация)
- [Input валидация] (#input-валидация)
- [XSS и CSRF защита] (#xss-и-csrf-защита)
- [SQL Injection защита] (#sql-injection-защита)
- [Session сигурност] (#session-сигурност)
- [HTTPS и SSL/TLS] (#https-и-ssl-tls)
- [Dependency сигурност] (#dependency-сигурност)
- [Production security checklist] (#production-security-checklist)

### ## OWASP Top 10 защита

ARCHModels следва най-добрите практики за защита срещу [OWASP Top 10] (<https://owasp.org/www-project-top-ten/>) уязвимости.

#### ### 1. Broken Access Control

##### \*\*Защита\*\*:

- Laravel Middleware за authentication
- Gates и Policies за authorization
- Проверка на ownership на ресурси

##### \*\*Пример\*\*:

```
```php
// app/Http/Controllers/HistoryController.php
public function show($id)
{
    $decision = DecisionNode::findOrFail($id);

    // Проверка дали потребителят е собственик
    if ($decision->user_id !== auth()->id()) {
        abort(403, 'Unauthorized action.');
```

2. Cryptographic Failures

Защита:

- Силна `APP\_KEY` за encryption
- Bcrypt за hashing на пароли (Laravel default)
- HTTPS за всички комуникации

Важно:

```
```bash
Никога не използвайте слаб APP_KEY
php artisan key:generate
```

```

Проверете силата на APP_KEY
Трябва да е 32 символа base64 encoded
`,`

3. Injection (SQL, Command)

Защита:
- Eloquent ORM и Query Builder (prepared statements)
- Валидация на всички входни данни
- Никакви raw SQL заявки без binding

Безопасен код:
```php
// ДОБРЕ: Eloquent ORM
$questions = Question::where('country', $country)->get();

// ДОБРЕ: Query Builder с bindings
$answers = DB::table('answers')
    ->where('question_id', '=', $questionId)
    ->get();

// ЛОШО: Raw SQL без binding
// $results = DB::select("SELECT * FROM questions WHERE country =
'$country'");
`,`

### 4. Insecure Design

**Защита**:
- Rate limiting на всички wizard endpoints
- Session timeout (120 минути)
- Validation на всички inputs

### 5. Security Misconfiguration

**Вижте**: [Production Security Checklist] (#production-security-checklist)

### 6. Vulnerable and Outdated Components

**Вижте**: [Dependency сигурност] (#dependency-сигурност)

### 7. Identification and Authentication Failures

**Вижте**: [Автентикация и авторизация] (#автентикация-и-авторизация)

### 8. Software and Data Integrity Failures

**Защита**:
- Composer lock file за reproducible builds
- Package.json lock за NPM dependencies
- Integrity checks при деплой

### 9. Security Logging and Monitoring

**Вижте**: [OPERATIONS.md] (OPERATIONS.md) - Logging section

### 10. Server-Side Request Forgery (SSRF)

```

```

**Защита**:
- ARCHModels не прави external HTTP requests
- Ако се добавят, използвайте whitelist на URLs

---

## Управление на секрети

### Environment Variables

**Критични secrets**:
- `APP_KEY` - Laravel encryption key
- `DB_PASSWORD` - Database парола
- `MAIL_PASSWORD` - SMTP парола

**Правила**:
1. **Никога** не commit-вайте `.env` файл в git
2. Използвайте силни пароли (мин. 16 символа)
3. Ротирайте secrets редовно (на всеки 90 дни)
4. Различни secrets за dev/staging/production

**Генериране на силна парола**:
```bash
Linux/Mac
openssl rand -base64 32

PHP
php -r "echo bin2hex(random_bytes(16));"
```

### .gitignore настройка

Уверете се, че `.gitignore` съдържа:

```
.env
.env.backup
.env.production
.env.*.php
*.key
*.pem
```

### Защита на .env в production

**B .htaccess** (cPanel):
```
```apache
<FilesMatch "^\.env">
    Order allow,deny
    Deny from all
</FilesMatch>
```

B Nginx (VPS):
```
```nginx
location ~ /\.env {
 deny all;
 return 404;
}
```

```

```

...

---

## Автентикация и авторизация

### Laravel Breeze конфигурация

ARCHModels използва Laravel Breeze за автентикация.

**Security настройки**:

1. **Password requirements** (минимум):
   - Минимум 8 символа
   - Email verification (опционално)

2. **Rate limiting**:
   ```php
 // config/auth.php
 'throttle' => [
 'max_attempts' => 5,
 'decay_minutes' => 1,
],
   ```

3. **Session lifetime**:
   ```bash
 SESSION_LIFETIME=120 # 2 часа
   ```

### Two-Factor Authentication (2FA)

ARCHModels не използва 2FA понастоящем, но може да се добави с:
- Laravel Fortify
- Google Authenticator integration

**Препоръка**: Добавете 2FA за admin потребители в бъдещи версии.

### Password hashing

Laravel използва Bcrypt по подразбиране:

```php
// Автоматично при регистрация
$user->password = Hash::make($request->password);

// Проверка при login
if (Hash::check($request->password, $user->password)) {
 // Валидна парола
}
```

**Важно**: Никога не съхранявайте plain text пароли!

---

## Input валидация

### Validation rules

```

```

**Всички входни данни трябва да бъдат валидирани!**

**Пример за wizard**:
```php
public function answer(Request $request)
{
 $validated = $request->validate([
 'questionId' => 'required|integer|exists:questions,id',
 'answerId' => 'required|integer|exists:answers,id',
]);

 // Използвайте $validated, НЕ $request->all()
}
```

### Custom валидация

```php
// app/Rules/ValidCountry.php
class ValidCountry implements Rule
{
 public function passes($attribute, $value)
 {
 return in_array($value, ['BG', 'DE']);
 }

 public function message()
 {
 return 'Невалиден код на държава.';
 }
}

// Използване
$request->validate([
 'country' => ['required', new ValidCountry],
]);
```

### Sanitization

```php
// Laravel автоматично escape-ва данни в Blade и Inertia
// Но ако използвате raw HTML:

use Illuminate\Support\Str;

$clean = Str::of($input)->trim()->limit(255);
$clean = strip_tags($input); // Премахва HTML tags
$clean = htmlspecialchars($input, ENT_QUOTES, 'UTF-8');
```

---

## XSS и CSRF защита

### Cross-Site Scripting (XSS)

**Защита в Laravel**:

```

```

1. **Blade templates** (автоматично escaping):
```blade
{{-- БЕЗОПАСНО: Автоматично escape --}}
<p>{{ $userInput }}</p>

{{-- ОПАСНО: Без escape --}}
<p>{!! $userInput !!</p>
```

2. **Vue components**:
```vue
<!-- БЕЗОПАСНО: Автоматично escape -->
<p>{{ userInput }}</p>

<!-- ОПАСНО: v-html без sanitization -->
<p v-html="userInput"></p>
```

**Правило**: Никога не използвайте `v-html` с user-generated content
без sanitization!

### Cross-Site Request Forgery (CSRF)

**Защита в Laravel**:

Laravel автоматично проверява CSRF token за всички POST/PUT/DELETE
заявки.

**Inertia.js автоматично включва CSRF token**:
```javascript
// Inertia автоматично добавя X-CSRF-TOKEN header
router.post('/wizard/answer', { ... });
```

**За ръчни axios заявки**:
```javascript
axios.defaults.headers.common['X-CSRF-TOKEN'] =
document.querySelector('meta[name="csrf-token"]').content;
```

---

## SQL Injection защита

### Използване на Eloquent ORM

**БЕЗОПАСНО**:
```php
// Eloquent ORM
$questions = Question::where('country', $country)->get();

// Query Builder с bindings
$results = DB::table('answers')
 ->where('question_id', '=', $questionId)
 ->where('id', '=', $answerId)
 ->first();
```

```

```

**ОПАСНО**:
```php
// НИКОГА не правете това!
$results = DB::select("SELECT * FROM questions WHERE country =
'$country'");
```

### Raw queries (ако е необходимо)

**Ако трябва** да използвате raw SQL, винаги използвайте bindings:

```php
// ПРАВИЛНО: с parameter binding
$results = DB::select('SELECT * FROM questions WHERE country = ?',
[$country]);

// Или named bindings
$results = DB::select('SELECT * FROM questions WHERE country =
:country', [
 'country' => $country
]);
```

---

## Session сигурност

### Session конфигурация

**Production настройки** (`config/session.php`):

```php
return [
 'driver' => env('SESSION_DRIVER', 'file'),
 'lifetime' => env('SESSION_LIFETIME', 120),
 'expire_on_close' => false,
 'encrypt' => false, // false при file driver
 'http_only' => true, // Защита срещу XSS
 'same_site' => 'lax', // CSRF защита
 'secure' => env('SESSION_SECURE_COOKIE', true), // HTTPS only
];
```

**Environment variables**:
```bash
SESSION_LIFETIME=120
SESSION_SECURE_COOKIE=true # CAMO с HTTPS
SESSION_DOMAIN=null # или вашият domain
```

### Session hijacking защита

1. **Regenerate session ID при login**:
```php
// Laravel Breeze автоматично го прави
$request->session()->regenerate();
```

2. **Invalidate session при logout**:

```

```

```php
Auth::logout();
$request->session()->invalidate();
$request->session()->regenerateToken();
```

3. **Session timeout**:
```bash
SESSION_LIFETIME=120 # 2 часа
```

---

## HTTPS и SSL/TLS

### Force HTTPS

**B .htaccess** (cPanel):
```apache
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule ^(.*)$ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]
```

**B Nginx** (VPS):
```nginx
server {
 listen 80;
 server_name your-domain.com;
 return 301 https://$server_name$request_uri;
}
```

**B Laravel** (`app/Providers/AppServiceProvider.php`):
```php
use Illuminate\Support\Facades\URL;

public function boot()
{
 if ($this->app->environment('production')) {
 URL::forceScheme('https');
 }
}
```

### SSL/TLS Best Practices

1. **Используйте Let's Encrypt** за бесплатный SSL сертификат
2. **TLS 1.2+** (TLS 1.0/1.1 ca deprecated)
3. **Strong cipher suites**

**Nginx SSL config**:
```nginx
ssl_protocols TLSv1.2 TLSv1.3;
ssl_ciphers 'ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256';
ssl_prefer_server_ciphers off;
```

```

```

---

## Dependency сигурност

### Composer dependencies

**Проверка за уязвимости**:
```bash
Composer audit
composer audit

Актуализиране на dependencies
composer update --with-dependencies

Проверка за outdated пакети
composer outdated
```

**Best practices**:
1. Редовно update (месечно)
2. Review changelog преди update
3. Тествайте след update
4. Използвайте `composer.lock` за reproducibility

### NPM dependencies

**Проверка за уязвимости**:
```bash
NPM audit
npm audit

Автоматично fix
npm audit fix

Update dependencies
npm update
```

**Best practices**:
1. Редовно `npm audit` (седмично)
2. Използвайте `package-lock.json`
3. Avoid deprecated packages

### Dependency scanning automation

**GitHub Dependabot** (ако използвате GitHub):
- Автоматични pull requests за security updates
- Configurable update frequency

---

## Production security checklist

### Преди деплой

- [ ] `APP_ENV=production` в `.env`
- [ ] `APP_DEBUG=false` в `.env`
- [ ] Силна `APP_KEY` генерирана
- [ ] Силна database парола (мин. 16 символа)

```

- [] `.env` файлът **не е** в git repository
- [] `.env` файлът е защитен в `.htaccess`/`nginx.conf`
- [] `composer install --no-dev` (без dev dependencies)
- [] `npm run build` (production build)
- [] SSL/TLS сертификат инсталиран и валиден
- [] HTTPS forced (redirect от HTTP)
- [] `SESSION\_SECURE\_COOKIE=true` (при HTTPS)
- [] Rate limiting активиран на всички endpoints
- [] Error logging конфигуриран (не показвайте stack traces на user)
- [] Backup стратегия на място
- [] Database credentials различни от default (root/admin)

След деплой

- [] Health check endpoint работи (`/health`)
- [] SSL сертификат валиден (проверете с SSL Labs)
- [] Всички форми работят (CSRF tokens)
- [] Login/Register/Logout работят
- [] Wizard flow работи от край до край
- [] Error pages (404, 500) показват се правилно
- [] Logs се записват в `storage/logs/`
- [] Permissions на `storage/` и `bootstrap/cache/` са правилни

Редовна поддръжка

- [] Седмично: `composer audit` и `npm audit`
- [] Месечно: Update dependencies
- [] Месечно: Review security logs
- [] Тримесечно: Смяна на database password
- [] Тримесечно: SSL certificate renewal (ако не е auto)
- [] Годишно: Penetration testing (ако е възможно)

Security headers

Препоръчани HTTP headers

```

**B Nginx**:
```nginx
add_header X-Frame-Options "SAMEORIGIN" always;
add_header X-Content-Type-Options "nosniff" always;
add_header X-XSS-Protection "1; mode=block" always;
add_header Referrer-Policy "no-referrer-when-downgrade" always;
add_header Content-Security-Policy "default-src 'self'" always;
```

```

```

**B .htaccess**:
```apache
<IfModule mod_headers.c>
 Header set X-Frame-Options "SAMEORIGIN"
 Header set X-Content-Type-Options "nosniff"
 Header set X-XSS-Protection "1; mode=block"
 Header set Referrer-Policy "no-referrer-when-downgrade"
</IfModule>
```

```

```

## Incident response

### При security инцидент

1. **Незабавно**:
  - Активирайте maintenance mode: `php artisan down`
  - Изолирайте проблема
  - Променете всички credentials (DB, SMTP, APP_KEY)

2. **Разследване**:
  - Проверете логове: `storage/logs/laravel.log`
  - Проверете database за неоторизирани промени
  - Проверете `.env` за промени

3. **Remediation**:
  - Patch уязвимостта
  - Restore от backup (ако е необходимо)
  - Update всички dependencies

4. **Post-incident**:
  - Документирайте инцидента
  - Уведомете засегнати потребители (ако има data breach)
  - Добавете защита срещу подобни атаки

### Контакти за security

- **Security Lead**: security@your-domain.com
- **Emergency**: +359 XXX XXX XXX

---

## Допълнителни ресурси

- [OWASP Top 10] (https://owasp.org/www-project-top-ten/)
- [Laravel Security Best Practices] (https://laravel.com/docs/security)
- [Mozilla Web Security Guidelines] (https://infosec.mozilla.org/guidelines/web\_security)
- [SSL Labs Server Test] (https://www.ssllabs.com/ssltest/)

---

**Важно**: Security е постоянен процес. Редовно review-вайте този документ и актуализирайте security практики.

```

Backup & Disaster Recovery за ARCHModels

Този документ описва стратегиите за backup и възстановяване на ARCHModels при инциденти или загуба на данни.

Съдържание

- [Backup стратегия] (#backup-стратегия)
- [Какво трябва да се архивира] (#какво-трябва-да-се-архивира)
- [Автоматизация на backup] (#автоматизация-на-backup)
- [Съхранение на backups] (#съхранение-на-backups)
- [Тестване на backups] (#тестване-на-backups)
- [Disaster recovery план] (#disaster-recovery-план)
- [Recovery procedures] (#recovery-procedures)

Backup стратегия

3-2-1 Rule

ARCHModels следва ****3-2-1 backup правилото****:

- ****3 copies**** на данните (1 production + 2 backups)
- ****2 различни media**** (локално + облак/external)
- ****1 offsite copy**** (извън production сървъра)

Backup честота

| Данни | Честота | Retention (запазване) | Метод |
|-------------------------------|--------------------|-----------------------|-------------------|
| ----- | ----- | ----- | ----- |
| **Database** | Дневно (ежедневно) | 30 дни | mysqldump |
| **Application files** | Седмично | 4 седмици | tar/zip archive |
| **Storage директория** | Седмично | 4 седмици | rsync/tar |
| **env файл** | При промяна | Неограничено | Ръчно (encrypted) |
| **Logs** | Месечно (архив) | 6 месеца | tar.gz |

Какво трябва да се архивира

1. Database (MySQL)

****Критичност****: ● ****КРИТИЧНО****

****Съдържание****:

- Потребители (`users` table)
- Въпроси и отговори (`questions`, `answers`)
- История на решения (`decision\_nodes`)
- Sessions (опционално, може да се пропусне)

****Команда****:

```bash

# Пълен dump на database

mysqldump -u username -p database\_name > backup-\$(date +%Y%m%d).sql

# С компресия

mysqldump -u username -p database\_name | gzip > backup-\$(date +%Y%m%d).sql.gz

```
Само структура (без данни)
mysqldump -u username -p --no-data database_name > schema-$(date
+%Y%m%d).sql
```

```
Само важни таблици
mysqldump -u username -p database_name users questions answers
decision_nodes > backup-$(date +%Y%m%d).sql
```
```

2. Application Files

****Критичност**:** ● ****СРЕДНО**** (може да се restore от git)

****Директории**:**

```

```
archmodels/
├─ app/ # PHP код
├─ config/ # Конфигурационни файлове
├─ database/ # Миграции и seeders
├─ resources/ # Vue компоненти, CSS
├─ routes/ # Route definitions
└─ public/ # Compiled assets
```
```

****Команда**:**

```bash

# Tar archive на цялото приложение

```
tar -czf archmodels-backup-$(date +%Y%m%d).tar.gz archmodels/ \
--exclude='archmodels/storage/logs/*' \
--exclude='archmodels/vendor' \
--exclude='archmodels/node_modules'
```

# Или само важните файлове

```
tar -czf archmodels-code-$(date +%Y%m%d).tar.gz \
archmodels/app \
archmodels/config \
archmodels/database \
archmodels/resources \
archmodels/routes \
archmodels/public
```
```

3. Storage директория

****Критичност**:** ● ****СРЕДНО****

****Съдържание**:**

- `storage/app/` - User uploads (ако има)
- `storage/app/data/` - JSON decision tree файлове
- `storage/framework/` - Cache, sessions (може да се пропусне)
- `storage/logs/` - Логове (архивирайте месечно)

****Команда**:**

```

```bash
Backup на storage/app
tar -czf storage-app-backup-$(date +%Y%m%d).tar.gz storage/app/

Backup на decision tree JSON files
tar -czf decision-trees-$(date +%Y%m%d).tar.gz storage/app/data/
```

---

### 4. .env файл

**Критичност**: ● **КРИТИЧНО** (съдържа secrets)

**Важно**: `.env` файлът съдържа чувствителна информация (database passwords, APP_KEY).

**Backup**:
```bash
Криптиран backup на .env
gpg --symmetric --cipher-algo AES256 .env

Или просто копирайте на сигурно място (ИЗВЪН сървъра)
cp .env .env.backup-$(date +%Y%m%d)
```

**ВАЖНО**:
- **Никога** не archive-вайте `.env` без encryption
- Съхранявайте `.env` backups на сигурно, offsite място

---

### 5. Compiled assets (опционално)

**Критичност**: ● **НИСКО** (може да се rebuilds от source)

Може да се пропусне, тъй като може да се regenerate с `npm run build`.

---

## Автоматизация на backup

### Bash скрипт за автоматичен backup

Създайте `backup.sh`:

```bash
#!/bin/bash
ARCHModels Backup Script

Конфигурация
BACKUP_DIR="/home/username/backups"
DB_NAME="username_archmodels"
DB_USER="username_arch_user"
DB_PASS="your_password"
APP_DIR="/home/username/archmodels"
DATE=$(date +%Y%m%d)
RETENTION_DAYS=30

```

```

Създаване на backup директория
mkdir -p "$BACKUP_DIR/database"
mkdir -p "$BACKUP_DIR/storage"

1. Database backup
echo "Backing up database..."
mysqldump -u "$DB_USER" -p"$DB_PASS" "$DB_NAME" | gzip >
"$BACKUP_DIR/database/db-backup-$DATE.sql.gz"

2. Storage backup
echo "Backing up storage..."
tar -czf "$BACKUP_DIR/storage/storage-app-$DATE.tar.gz" -C "$APP_DIR"
storage/app/

3. .env backup (криптиран)
echo "Backing up .env file..."
cp "$APP_DIR/.env" "$BACKUP_DIR/.env-backup-$DATE"

4. Изтриване на стари backups (по-стари от RETENTION_DAYS)
echo "Cleaning old backups..."
find "$BACKUP_DIR/database" -name "db-backup-*.sql.gz" -mtime
+$RETENTION_DAYS -delete
find "$BACKUP_DIR/storage" -name "storage-app-*.tar.gz" -mtime
+$RETENTION_DAYS -delete
find "$BACKUP_DIR" -name ".env-backup-*" -mtime +$RETENTION_DAYS -
delete

echo "Backup completed successfully!"
```



```

Permissions:
```bash
chmod +x backup.sh
```

Автоматизация с Cron (cPanel)


```

**cPanel > Cron Jobs**:

```bash
Ежедневен backup в 2:00 AM
0 2 * * * /home/username/backup.sh > /dev/null 2>&1

Седмичен backup в неделя в 3:00 AM
0 3 * * 0 /home/username/backup-weekly.sh > /dev/null 2>&1
```

**Добавете email notification**:
```bash
MAILTO="admin@your-domain.com"
0 2 * * * /home/username/backup.sh
```

---

### Автоматизация с systemd timer (VPS)

```


```


```

```

**Създайте** `/etc/systemd/system/archmodels-backup.service`:

```ini
[Unit]
Description=ARCHModels Backup Service

[Service]
Type=oneshot
User=www-data
ExecStart=/var/www/archmodels/backup.sh
```

**Създайте** `/etc/systemd/system/archmodels-backup.timer`:

```ini
[Unit]
Description=ARCHModels Daily Backup Timer

[Timer]
OnCalendar=daily
Persistent=true

[Install]
WantedBy=timers.target
```

**Активирайте**:
```bash
sudo systemctl enable archmodels-backup.timer
sudo systemctl start archmodels-backup.timer
```

---

## Съхранение на backups

### Локално съхранение (cPanel)

**Директория**:
```
/home/username/backups/
├── database/
│ ├── db-backup-20260131.sql.gz
│ ├── db-backup-20260130.sql.gz
│ └── ...
├── storage/
│ ├── storage-app-20260131.tar.gz
│ └── ...
└── .env-backup-20260131
```

**Проблем**: Ако сървърът се crashне, локалните backups могат да се загубят.

**Решение**: Offsite backup (вижте по-долу).

---

```

```

#### Offsite backup (препоръчително)

#### Опция 1: FTP/SFTP към друг сървър

```bash
В backup.sh скрипта добавете:

Upload към remote server
lftp -u username,password sftp://backup-server.com <<EOF
cd /backups/archmodels
put $BACKUP_DIR/database/db-backup-$DATE.sql.gz
put $BACKUP_DIR/storage/storage-app-$DATE.tar.gz
bye
EOF
```

#### Опция 2: Cloud storage (AWS S3, DigitalOcean Spaces)

**Инсталирайте AWS CLI**:
```bash
pip install awscli
```

**Конфигурация**:
```bash
aws configure
AWS Access Key ID: YOUR_KEY
AWS Secret Access Key: YOUR_SECRET
Default region: eu-central-1
```

**Upload към S3**:
```bash
В backup.sh добавете:
aws s3 cp "$BACKUP_DIR/database/db-backup-$DATE.sql.gz" s3://your-
bucket/archmodels/database/
aws s3 cp "$BACKUP_DIR/storage/storage-app-$DATE.tar.gz" s3://your-
bucket/archmodels/storage/
```

#### Опция 3: rsync към remote server

```bash
rsync backup към remote server
rsync -avz --delete "$BACKUP_DIR/" username@backup-
server.com:/backups/archmodels/
```

---

## Тестване на backups

### Защо е важно?

**Backup без тестване = няма backup!**

Тествайте backups редовно, за да сте сигурни, че могат да бъдат възстановени.

```

Процедура за тестване

****Месечно тестване**:**

1. ****Download backup**:**

```bash

scp username@server:/backups/database/db-backup-YYYYMMDD.sql.gz /tmp/

```

2. ****Създайте test database**:**

```bash

mysql -u root -p -e "CREATE DATABASE archmodels\_test;"

```

3. ****Restore backup**:**

```bash

gunzip < /tmp/db-backup-YYYYMMDD.sql.gz | mysql -u root -p  
archmodels\_test

```

4. ****Проверете данните**:**

```bash

mysql -u root -p archmodels\_test -e "SELECT COUNT(\*) FROM users;"

mysql -u root -p archmodels\_test -e "SELECT COUNT(\*) FROM questions;"

```

5. ****Изтрийте test database**:**

```bash

mysql -u root -p -e "DROP DATABASE archmodels\_test;"

```

Disaster recovery план

Сценарии

| Сценарий | Вероятност | Impact | RTO* | RPO** | |
|-----------------------------------|-------------|----------|---------|-------|--|
| ----- | ----- | ----- | ----- | ----- | |
| **Database corruption** | Средна | Висок | 2 часа | 1 ден | |
| **Filesystem crash** | Ниска | Висок | 4 часа | 1 ден | |
| **Хостинг провайдер down** | Много ниска | Критичен | 24 часа | 1 ден | |
| **Ransomware attack** | Ниска | Критичен | 8 часа | 1 ден | |
| **Accidental deletion** | Средна | Среден | 1 час | 1 ден | |

*RTO = Recovery Time Objective (максимално време за възстановяване)

**RPO = Recovery Point Objective (максимална загуба на данни)

Recovery team

| Роля | Име | Контакт | Отговорности | |
|-------------------------------|-------|-------------|-------------------------|--|
| ----- | ----- | ----- | ----- | |
| **Incident Commander** | TBD | email/phone | Координация на recovery | |
| **Database Admin** | TBD | email/phone | Database restore | |
| **DevOps Engineer** | TBD | email/phone | Application restore | |

| **Communication Lead** | TBD | email/phone | Комуникация с потребители |

Recovery procedures

Сценарий 1: Database corruption

****Симптоми**:**

- Грешки при заявки към database
- Tables marked as crashed
- Connection refused

****Стъпки**:**

1. ****Активирайте maintenance mode**:**

```
```bash
php artisan down --message="Database maintenance in progress"
```
```

2. ****Опитайте repair на tables**:**

```
```bash
mysqlcheck -u root -p --auto-repair database_name
```
```

3. ****Ако repair не успее, restore от backup**:**

```
```bash
Download най-новия backup
cd /tmp
scp username@backup-server:/backups/database/db-backup-LATEST.sql.gz .
```
```

Backup на текущата (corrupt) database

```
mysqldump -u username -p database_name > corrupt-backup-$(date +%Y%m%d).sql
```

Drop текущата database

```
mysql -u username -p -e "DROP DATABASE database_name;"
mysql -u username -p -e "CREATE DATABASE database_name CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;"
```

Restore от backup

```
gunzip < db-backup-LATEST.sql.gz | mysql -u username -p database_name
```
```

4. **\*\*Тествайте приложението\*\*:**

```
```bash
php artisan tinker
>>> \App\Models\User::count();
>>> \App\Models\Question::count();
```
```

5. **\*\*Деактивирайте maintenance mode\*\*:**

```
```bash
php artisan up
```
```

**\*\*RTO\*\*:** ~2 часа

**\*\*RPO\*\*:** До 24 часа (ако backup е от предната вечер)

---

### ### Сценарий 2: Пълен filesystem crash

#### \*\*Симптоми\*\*:

- Сървърът не е достъпен
- cPanel не работи
- FTP/SSH connection refused

#### \*\*Стъпки\*\*:

1. **\*\*Контактувайте хостинг провайдъра\*\*:**
  - Определете дали е hardware или software проблем
  - Ако е unrecoverable, подгответе се за пълен restore
2. **\*\*Подгответе нов сървър/хостинг\*\*:**
  - Настройте нов cPanel акаунт или VPS
  - Инсталирайте необходимите компоненти (PHP 8.2, MySQL, Nginx)
3. **\*\*Restore application code\*\*:**

```
```bash
# Ако имате git repo:
git clone https://github.com/your-org/archmodels.git

# Или от backup:
tar -xzf archmodels-backup-LATEST.tar.gz
```
```
4. **\*\*Restore database\*\*:**

```
```bash
mysql -u username -p database_name < db-backup-LATEST.sql
```
```
5. **\*\*Restore .env файл\*\*:**

```
```bash
cp .env-backup-LATEST archmodels/.env
# Редактирайте с нови database credentials
```
```
6. **\*\*Restore storage\*\*:**

```
```bash
tar -xzf storage-app-LATEST.tar.gz -C archmodels/
```
```
7. **\*\*Laravel setup\*\*:**

```
```bash
cd archmodels/backend
composer install --no-dev
npm install
npm run build
php artisan key:generate # ако .env няма APP_KEY
php artisan storage:link
php artisan optimize
```
```
8. **\*\*Тествайте\*\*:**

```
```bash
curl https://new-domain.com/health
```
```

...

**\*\*RTO\*\*:** ~4-8 часа (в зависимост от хостинг provider response time)  
**\*\*RPO\*\*:** До 24 часа

---

### Сценарий 3: Accidental deletion (таблица или данни)

**\*\*Пример\*\*:** Някой случайно изтрил важна таблица или данни.

**\*\*Стъпки\*\*:**

1. **\*\*Спрете приложението веднага\*\*:**

```
```bash
php artisan down
```
```

2. **\*\*Identify какво е изтрито\*\*:**

```
```bash
# Проверете кои таблици липсват
mysql -u username -p -e "SHOW TABLES FROM database_name;"
```
```

3. **\*\*Restore само липсващите данни\*\*:**

```
```bash
# Extract само нужната таблица от backup
gunzip < db-backup-LATEST.sql.gz | sed -n '/CREATE
TABLE.*users/,/UNLOCK TABLES/p' | mysql -u username -p database_name
```
```

# Или restore цялата database (ако е по-бързо)

```
```
```

4. ****Тествайте**:**

```
```bash
mysql -u username -p database_name -e "SELECT COUNT(*) FROM users;"
```
```

5. ****Рестартирайте приложението**:**

```
```bash
php artisan up
```
```

****RTO**:** ~1 час

****RPO**:** До 24 часа

Сценарий 4: Ransomware attack

****Симптоми**:**

- Файловете са encrypted
- Ransom note в директории
- Database е locked

****Стъпки**:**

1. ****Незабавно изолирайте сървъра**:**

- Disconnect от интернет (ако имате достъп)

- Спрете web server и MySQL

2. ****НЕ плащайте ransom****

3. ****Анализирайте атаката****:

- Определете entry point (vulnerable plugin, weak password, etc.)
- Документирайте за forensics

4. ****Пълен restore от offsite backup****:

- Следвайте процедурата за "Пълен filesystem crash"
- ****ВАЖНО****: Използвайте backup ПРЕДИ атаката (може да е от 2-3 дни назад)

5. ****Patch уязвимостта****:

- Update всички dependencies
- Променете всички пароли
- Добавете security мерки

6. ****Тествайте за malware****:

```

```bash
clamscan -r /path/to/archmodels
```

```

****RTO****: ~8-24 часа
****RPO****: 1-3 дни (зависи от кога е започнала атаката)

Post-recovery checklist

След всяко успешно възстановяване:

- [] Документирайте инцидента (какво се случи, причина, resolution)
- [] Update disaster recovery plan, ако е необходимо
- [] Review и подобрете backup процеса
- [] Комуникирайте с потребители (ако имаше downtime)
- [] Проведете post-mortem meeting с team
- [] Добавете мерки за предотвратяване на подобен инцидент

Контакти

Emergency contacts

- ****Technical Lead****: tech@your-domain.com / +359 XXX XXX XXX
- ****Hosting Provider Support****: support@hosting.com / +XXX XXX XXX
- ****Backup Storage Provider****: support@backup-provider.com

Escalation path

1. ****Level 1****: DevOps Engineer
2. ****Level 2****: Technical Lead
3. ****Level 3****: CTO / Management

****Забележка****: Този документ трябва да се актуализира редовно и да се тества backup/recovery процедурите минимум веднъж на тримесечие.

Scaling ръководство за ARCHModels

Този документ описва стратегиите за мащабиране на ARCHModels при увеличаване на трафика и натоварването.

Съдържание

- [Текуща архитектура] (#текуща-архитектура)
- [Performance baseline] (#performance-baseline)
- [Вертикално мащабиране] (#вертикално-мащабиране)
- [Хоризонтално мащабиране] (#хоризонтално-мащабиране)
- [Кеширане стратегии] (#кеширане-стратегии)
- [Database оптимизация] (#database-оптимизация)
- [CDN и static assets] (#cdn-и-static-assets)
- [Load balancing] (#load-balancing)
- [Capacity planning] (#capacity-planning)

Текуща архитектура

ARCHModels понастоящем работи на ****shared hosting / single server**** конфигурация:

...

[Browser] → [Nginx/Apache] → [PHP-FPM] → [MySQL]

↓

[File Cache]

...

****Ограничения**:**

- Single point of failure
- Ограничени CPU/RAM ресурси
- File-based cache (по-бавно от Redis)
- Липса на load balancing

Performance baseline

Текущи метрики (shared hosting)

| Метрика | Стойност | Забележка |
|------------------------------|-----------|-------------------------------------|
| **Concurrent users** | ~50-100 | При повече, performance degradation |
| **Response time** | 300-800ms | За wizard страница |
| **Database queries/request** | 5-10 | Зависи от страницата |
| **Memory usage** | 50-80MB | Per request |
| **Storage** | ~200MB | Application + database |

Bottlenecks

1. ****CPU****: PHP изпълнение на decision tree логика
2. ****Database****: При много concurrent queries
3. ****File I/O****: File-based cache and sessions
4. ****Network****: Споделена bandwidth на shared hosting

```

## Вертикално мащабиране

**Vertical scaling** = Upgrade на сървърни ресурси.

### Нива на мащабиране

#### Level 1: Entry (Shared Hosting)
**Капацитет**: ~100-200 concurrent users

**Конфигурация**:
- 1 CPU core
- 512MB-1GB RAM
- 10GB SSD storage
- Shared bandwidth

**Цена**: $5-15/месец

**Подходящ за**: MVP, малки проекти

---

#### Level 2: Small VPS
**Капацитет**: ~500-1000 concurrent users

**Конфигурация**:
- 2 CPU cores
- 4GB RAM
- 50GB SSD storage
- Redis за cache/sessions
- Dedicated IP

**Миграция от shared hosting**:
1. Настройте VPS (Ubuntu 22.04)
2. Инсталирайте LEMP stack (Linux, Nginx, MySQL, PHP)
3. Инсталирайте Redis
4. Migrate database и application code
5. Update `.env`:
    ```bash
 CACHE_DRIVER=redis
 SESSION_DRIVER=redis
    ```

**Цена**: $20-40/месец (DigitalOcean, Linode)

**Подходящ за**: Малки до средни проекти

---

#### Level 3: Medium VPS + Redis
**Капацитет**: ~2000-5000 concurrent users

**Конфигурация**:
- 4 CPU cores
- 8GB RAM
- 100GB SSD storage
- Redis 7.0+ (dedicated instance)
- MySQL optimized (увеличени buffers)

**Оптимизации**:

```

```

1. **Redis конфигурация**:
```bash
redis.conf
maxmemory 2gb
maxmemory-policy allkeys-lru
save "" # Disable persistence за sessions
```

2. **MySQL tuning**:
```ini
my.cnf
[mysqld]
innodb_buffer_pool_size = 4G
max_connections = 200
query_cache_size = 64M
```

3. **PHP-FPM tuning**:
```ini
/etc/php/8.2/fpm/pool.d/www.conf
pm = dynamic
pm.max_children = 50
pm.start_servers = 10
pm.min_spare_servers = 5
pm.max_spare_servers = 20
```

**Цена**: $80-120/месес

**Подходящ за**: Средни проекти с растеж

---

#### Level 4: Large Server
**Капацитет**: ~10,000+ concurrent users

**Конфигурация**:
- 8+ CPU cores
- 16GB+ RAM
- 200GB+ SSD storage
- Dedicated Redis server
- Database read replicas (при необходимост)

**Цена**: $200-400/месец

**Подходящ за**: Големи проекти, enterprise

---

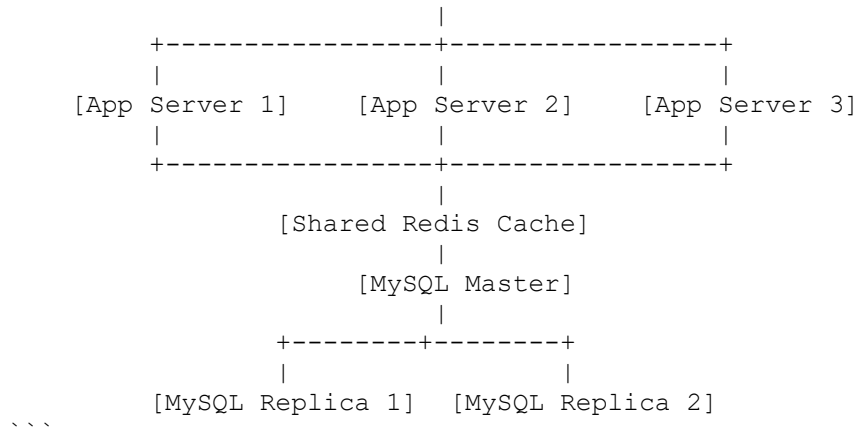
## Хоризонтално мащабиране

**Horizontal scaling** = Добавяне на повече сървъри.

### Архитектура за horizontal scaling
...

[Load Balancer]
(Nginx / HAProxy)

```



Компоненти

1. Load Balancer

****Nginx load balancer config**:**

```

```nginx
upstream archmodels_backend {
 least_conn; # Load balancing method
 server 192.168.1.10:80 max_fails=3 fail_timeout=30s;
 server 192.168.1.11:80 max_fails=3 fail_timeout=30s;
 server 192.168.1.12:80 max_fails=3 fail_timeout=30s;
}

server {
 listen 80;
 server_name archmodels.example;

 location / {
 proxy_pass http://archmodels_backend;
 proxy_set_header Host $host;
 proxy_set_header X-Real-IP $remote_addr;
 proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
 proxy_set_header X-Forwarded-Proto $scheme;
 }
}
```
...

```

2. Shared Redis

****Всички app servers трябва да използват централизиран Redis**:**

```

```bash
На app servers: .env
REDIS_HOST=redis-server.example # Dedicated Redis server IP
REDIS_PORT=6379
REDIS_PASSWORD=strong_password

CACHE_DRIVER=redis
SESSION_DRIVER=redis
```
...

```

3. Shared Storage (за user uploads)

****Опции**:**

****Опция A: NFS mount**** (просто, but single point of failure)

```
```bash
На app servers
sudo mount -t nfs redis-server.example:/storage
/var/www/archmodels/storage
```
```

****Опция B: S3-compatible storage**** (препоръчително)

```
```bash
.env
FILESYSTEM_DISK=s3
AWS_ACCESS_KEY_ID=your_key
AWS_SECRET_ACCESS_KEY=your_secret
AWS_DEFAULT_REGION=eu-central-1
AWS_BUCKET=archmodels-storage
```
```

Laravel config (`config/filesystems.php`):

```
```php
's3' => [
 'driver' => 's3',
 'key' => env('AWS_ACCESS_KEY_ID'),
 'secret' => env('AWS_SECRET_ACCESS_KEY'),
 'region' => env('AWS_DEFAULT_REGION'),
 'bucket' => env('AWS_BUCKET'),
],
```
```

4. Database read replicas

****3a read-heavy applications**:**

```
```php
// config/database.php
'mysql' => [
 'read' => [
 'host' => [
 '192.168.1.20', // Replica 1
 '192.168.1.21', // Replica 2
],
],
 'write' => [
 'host' => ['192.168.1.10'], // Master
],
 // ... other config
],
```
```

Laravel автоматично route-ва read queries към replicas.

Кеширане стратегии

1. Database query cache (Redis)

****Кеширане на въпроси и отговори**:**

```

```php
// app/Services/DecisionTreeEngine.php
public function getQuestion($id)
{
 return Cache::remember("question.$id", 3600, function () use ($id)
 {
 return Question::with('answers')->findOrFail($id);
 });
}

public function getDecisionTree($country)
{
 return Cache::remember("decision_tree.$country", 3600, function ()
 use ($country) {
 return
 file_get_contents(storage_path("app/data/decision_tree_$country.json"))
 ;
 });
}
```

```

****Изчистване на кеш при промяна**:**

```

```php
// app/Models/Question.php
protected static function booted()
{
 static::saved(function ($question) {
 Cache::forget("question.{ $question->id }");
 Cache::forget("decision_tree.{ $question->country }");
 });
}
```

```

2. View/Page cache

****Кеширане на цели страници** (за static съдържание):**

```

```php
// routes/web.php
Route::get('/', [HomeController::class, 'index'])
->middleware('cache.headers:public;max_age=600'); // 10 минути
```

```

3. API response cache

****Ако добавите API endpoints**:**

```

```php
public function index()
{
 return Cache::remember('api.questions.all', 600, function () {
 return Question::all();
 });
}
```

```

4. OPcache (PHP)

****Включете OPcache**** за да кешира compiled PHP код:

```
```ini
; /etc/php/8.2/fpm/conf.d/10-opcache.ini
opcache.enable=1
opcache.memory_consumption=256
opcache.interned_strings_buffer=16
opcache.max_accelerated_files=10000
opcache.revalidate_freq=60
opcache.fast_shutdown=1
```
```

Database оптимизация

1. Indexing

****Създайте индекси на често търсени полета**:**

```
```php
// database/migrations/XXXX_add_indexes_to_questions.php
Schema::table('questions', function (Blueprint $table) {
 $table->index('country');
 $table->index(['country', 'order']);
});

Schema::table('answers', function (Blueprint $table) {
 $table->index('question_id');
});

Schema::table('decision_nodes', function (Blueprint $table) {
 $table->index(['user_id', 'country']);
 $table->index('completed_at');
});
```
```

2. Eager loading

****Избягвайте N+1 query problem**:**

```
```php
// ЛОШО: N+1 queries
$questions = Question::all();
foreach ($questions as $question) {
 echo $question->answers->count(); // Separate query за всеки въпрос
}

// ДОБРЕ: Eager loading
$questions = Question::with('answers')->get();
foreach ($questions as $question) {
 echo $question->answers->count(); // Вече заредено
}
```
```

3. Database connection pooling

****PgBouncer за PostgreSQL**** или ****ProxySQL за MySQL****:

Намалява overhead от създаване на нови connections.

4. Query optimization

****Анализирайте бавни queries**:**

```
```sql
-- MySQL slow query log
SET GLOBAL slow_query_log = 'ON';
SET GLOBAL long_query_time = 1;

-- Analyze query performance
EXPLAIN SELECT * FROM questions WHERE country = 'BG';
```
```

CDN и static assets

Използване на CDN

****За по-бързо зареждане на CSS/JS/images**:**

****Опции**:**

- Cloudflare (безплатен tier)
- AWS CloudFront
- DigitalOcean Spaces CDN

****Конфигурация**:**

1. ****Upload compiled assets към CDN**:**

```
```bash
След npm run build
aws s3 sync public/build/ s3://archmodels-cdn/build/
```
```

2. ****Update asset URLs** (`config/app.php`):**

```
```php
'asset_url' => env('ASSET_URL', 'https://cdn.archmodels.example'),
```
```

3. ****Vite config** (`vite.config.js`):**

```
```javascript
export default defineConfig({
 build: {
 manifest: true,
 outDir: 'public/build',
 rollupOptions: {
 output: {
 assetFileNames: 'assets/[name]-[hash][extname]',
 },
 },
 },
});
```
```

Load balancing

```

### Health checks

**Load balancer трябва да проверява здравето на app servers**:

```nginx
upstream archmodels_backend {
 server 192.168.1.10:80 max_fails=3 fail_timeout=30s;
 server 192.168.1.11:80 max_fails=3 fail_timeout=30s;

 # Health check (Nginx Plus only)
 # health_check interval=10s fails=3 passes=2 uri=/health;
}
```

```

За **open-source Nginx**, използвайте external health check скрипт:

```

```bash
#!/bin/bash
health_check.sh
for server in 192.168.1.10 192.168.1.11; do
 curl -f http://$server/health || {
 echo "Server $server is down!"
 # Remove from upstream or alert
 }
done
```

```

Capacity planning

Оценка на капацитет

Формула:

```

```
Max concurrent users = (Available Memory / Memory per request) × 0.8
```

```

Пример:

- Available memory: 4GB = 4096MB
- Memory per request: 60MB (Laravel + PHP + overhead)
- Safety factor: 0.8 (20% buffer)

```

```

```

```

Max users = (4096 / 60) × 0.8 ≈ 54 concurrent users
```

```

Metrics за scaling decision

**Кога да upgrade-нете?*

| Метрика | Threshold за upgrade |
|--------------------------|--------------------------------|
| **CPU usage** | Sustained > 80% за > 10 минути |
| **RAM usage** | > 90% |
| **Response time** | p95 > 2s |
| **Error rate** | > 1% |
| **Database connections** | > 80% от max |

Growth projections

****Планирайте капацитет based на projected growth**:**

| Time | Expected users/day | Required tier |
|----------------|--------------------|--------------------|
| **Month 0-3** | < 500 | Shared hosting |
| **Month 3-6** | 500-2000 | Small VPS |
| **Month 6-12** | 2000-10000 | Medium VPS + Redis |
| **Year 2+** | 10000+ | Horizontal scaling |

Migration roadmap

От shared hosting към VPS

****Downtime**:** ~2-4 часа

****Стъпки**:**

1. Setup нов VPS
2. Мигрирайте database (backup → restore)
3. Мигрирайте application code
4. Update DNS записи
5. Тествайте
6. Switch traffic (update A record)

От single VPS към horizontal scaling

****Downtime**:** ~30 минути (DNS propagation)

****Стъпки**:**

1. Setup Redis server (shared cache/sessions)
2. Setup app server 2 и 3
3. Setup load balancer
4. Migrate sessions към Redis
5. Update DNS към load balancer IP
6. Monitor и adjust

Cost optimization

Serverless алтернативи (бъдещо)

За много малък или много голям трафик:

- ****AWS Lambda**** + ****API Gateway**** + ****RDS Aurora Serverless****
- ****Laravel Vapor**** (официален AWS serverless deployment)

****Предимства**:**

- Pay only for actual usage
- Auto-scaling

****Недостатъци**:**

- Cold start latency
- Vendor lock-in
- По-сложна конфигурация

```

---

## Performance testing

### Load testing tools

**Apache Bench**:
```bash
ab -n 1000 -c 50 https://archmodels.example/
```

**Siege**:
```bash
siege -c 100 -t 60s https://archmodels.example/
```

**k6** (препоръчително):
```javascript
// load-test.js
import http from 'k6/http';
import { check, sleep } from 'k6';

export let options = {
 stages: [
 { duration: '1m', target: 50 }, // Ramp up
 { duration: '3m', target: 50 }, // Sustained
 { duration: '1m', target: 100 }, // Peak
 { duration: '1m', target: 0 }, // Ramp down
],
};

export default function () {
 let res = http.get('https://archmodels.example/');
 check(res, { 'status is 200': (r) => r.status === 200 });
 sleep(1);
}
```

Run:
```bash
k6 run load-test.js
```

---

## Monitoring scaled infrastructure

**Препоръчани инструменти**:

- **Server monitoring**: Prometheus + Grafana
- **Application monitoring**: Laravel Horizon, Telescope (staging only)
- **Error tracking**: Sentry, Rollbar
- **Uptime monitoring**: UptimeRobot, Pingdom

---

## Summary

```

| Scale level | Users | Monthly cost | Complexity |
|-----------------------------------|-----------|--------------|------------|
| ----- | ----- | ----- | ----- |
| **Shared hosting** | < 200 | \$10 | Low |
| **Small VPS** | 500-1000 | \$30 | Medium |
| **Medium VPS + Redis** | 2000-5000 | \$100 | Medium |
| **Horizontal (3 servers)** | 10000+ | \$300+ | High |

****Препоръка**:** Започнете с shared hosting или Small VPS. Upgrade базирано на реални метрики, не на прогнози.

****Забележка**:** Scaling е итеративен процес. Мониторирайте, оптимизирайте и upgrade-вайте постепенно.

API Документация - ARCHModels

ARCHModels използва Inertia.js като мост между Laravel backend и Vue frontend, което означава че **няма** традиционен REST API. Вместо това, комуникацията се извършва чрез Inertia маршрути и формуляри.

Съдържание

- [Какво е Inertia.js?] (#какво-е-inertiajs)
- [Публични маршрути] (#публични-маршрути)
- [Wizard маршрути] (#wizard-маршрути)
- [Автентикация маршрути] (#автентикация-маршрути)
- [Защитени маршрути] (#защитени-маршрути)
- [Формат на данните] (#формат-на-данните)
- [Обработка на грешки] (#обработка-на-грешки)

Какво е Inertia.js?

Inertia.js е библиотека, която позволява създаване на Single Page Applications (SPA) без необходимост от отделен REST API. Вместо това:

1. **Laravel контролерите** връщат данни чрез `Inertia::render()`
2. **Vue компонентите** получават данните като props
3. **Формулярите** изпращат данни директно към Laravel routes
4. **Навигацията** се извършва без пълно презареждане на страницата

Предимства на Inertia.js

- Няма нужда от API endpoints и сериализация
- Автоматично валидиране и error handling
- Бърза разработка (Laravel routing + Vue components)
- Server-side rendering възможности

Публични маршрути

GET /

Описание: Начална страница на приложението

Laravel Route:

```
```php
Route::get('/', [HomeController::class, 'index'])->name('home');
```

**Controller**:

```
```php
class HomeController extends Controller
{
    public function index()
    {
        return Inertia::render('Home', [
            'countries' => [
                ['code' => 'BG', 'name' => 'България', 'flag' => 'BG'],
                ['code' => 'DE', 'name' => 'Бавария (Германия)', 'flag'
=> 'DE'],
```

```

        ],
    });
}
...

**Vue Component**: `resources/js/Pages/Home.vue`

**Props**:
```javascript
{
 countries: [
 { code: 'BG', name: 'България', flag: 'BG' },
 { code: 'DE', name: 'Бавария (Германия)', flag: 'DE' }
]
}
...

Отговор: Рендира начална страница с избор на държава

Wizard маршрути

POST /wizard/start

Описание: Стартира нова сесия за избраната държава

Laravel Route:
```php
Route::post('/wizard/start', [WizardController::class, 'start'])
    ->name('wizard.start')
    ->middleware('throttle:60,1'); // Rate limiting
...

**Request Body** (от Vue формуляр):
```javascript
{
 country: 'BG' // или 'DE'
}
...

Validation:
```php
$request->validate([
    'country' => 'required|in:BG,DE',
]);
...

**Controller Logic**:
```php
public function start(Request $request)
{
 $validated = $request->validate([
 'country' => 'required|in:BG,DE',
]);

 // Стартиране на decision tree engine

```

```

$engine = new DecisionTreeEngine();
$engine->loadTree($validated['country']);

$firstQuestion = $engine->getFirstQuestion();

// Съхранение в сесия
session([
 'wizard.country' => $validated['country'],
 'wizard.allowedModels' => $engine->getAllModels(),
 'wizard.answeredQuestions' => [],
]);

return Inertia::render('Wizard', [
 'country' => $validated['country'],
 'question' => $firstQuestion,
 'progress' => 0,
]);
}
...

Отговор (към Vue):
```javascript
{
  country: 'BG',
  question: {
    id: 1,
    text: 'Каква е стойността на проекта?',
    answers: [
      { id: 1, text: 'До 500 000 лв.' },
      { id: 2, text: 'Над 500 000 лв.' }
    ]
  },
  progress: 0
}
...

---

### POST /wizard/answer

**Описание**: Обработка отговор на въпрос и връща следващ въпрос или
резултат

**Laravel Route**:
```php
Route::post('/wizard/answer', [WizardController::class, 'answer'])
 ->name('wizard.answer')
 ->middleware('throttle:60,1');
...

Request Body:
```javascript
{
  questionId: 1,
  answerId: 2
}
...

**Validation**:

```

```

` ``php
$request->validate([
    'questionId' => 'required|integer',
    'answerId' => 'required|integer|exists:answers,id',
]);
` ``

**Controller Logic**:
` ``php
public function answer(Request $request)
{
    $validated = $request->validate([
        'questionId' => 'required|integer',
        'answerId' => 'required|integer|exists:answers,id',
    ]);

    $engine = new DecisionTreeEngine();
    $engine->restoreFromSession();

    $result = $engine->processAnswer(
        $validated['questionId'],
        $validated['answerId']
    );

    // Записване в сесия
    session()->push('wizard.answeredQuestions', [
        'questionId' => $validated['questionId'],
        'answerId' => $validated['answerId'],
    ]);

    if ($result['isComplete']) {
        // Записване на резултата в база данни
        DecisionNode::create([
            'user_id' => auth()->id(),
            'country' => session('wizard.country'),
            'recommended_model' => $result['recommendation']['model'],
            'decision_path' => session('wizard.answeredQuestions'),
            'completed_at' => now(),
        ]);

        return Inertia::render('Result', [
            'recommendation' => $result['recommendation'],
            'trace' => $result['trace'],
        ]);
    }

    return Inertia::render('Wizard', [
        'country' => session('wizard.country'),
        'question' => $result['nextQuestion'],
        'progress' => $result['progress'],
    ]);
}
` ``

**Отговор (следващ въпрос)**:
` ``javascript
{
    country: 'BG',
    question: {

```

```

        id: 2,
        text: 'Имате ли опит в управление на строителни проекти?',
        answers: [
            { id: 3, text: 'Да' },
            { id: 4, text: 'Не' }
        ]
    },
    progress: 33
}
...

**Отговор (финален резултат)**:
```javascript
{
 recommendation: {
 model: 'BG-3',
 title: 'Модел BG-3: Генерално изпълнение',
 description: 'Подходящ за проекти със средна сложност...',
 advantages: ['Преимущество 1', 'Преимущество 2'],
 disadvantages: ['Недостатък 1']
 },
 trace: [
 { question: 'Каква е стойността?', answer: 'Над 500 000 лв.' },
 { question: 'Имате ли опит?', answer: 'Не' }
]
}
...

GET /wizard/result

Описание: Показва резултата (ако потребителят обнови страницата)

Laravel Route:
```php
Route::get('/wizard/result', [WizardController::class, 'result'])->
>name('wizard.result');
```

Controller Logic:
```php
public function result()
{
    // Проверка дали има резултат в сесията
    if (!session()->has('wizard.result')) {
        return redirect()->route('home');
    }

    return Inertia::render('Result', session('wizard.result'));
}
...

---

## Автентикация маршрути

ARCHModels използва Laravel Breeze за автентикация.

```

```

### POST /register

**Описание**: Регистрация на нов потребител

**Request Body**:
```javascript
{
 name: 'Иван Иванов',
 email: 'ivan@example.com',
 password: 'password123',
 password_confirmation: 'password123'
}
```

**Validation**:
```php
$request->validate([
 'name' => 'required|string|max:255',
 'email' => 'required|string|email|max:255|unique:users',
 'password' => 'required|string|min:8|confirmed',
]);
```

**Отговор**: Redirect към `/` след успешна регистрация

---

### POST /login

**Описание**: Вход в системата

**Request Body**:
```javascript
{
 email: 'ivan@example.com',
 password: 'password123',
 remember: true
}
```

**Validation**:
```php
$request->validate([
 'email' => 'required|email',
 'password' => 'required',
]);
```

**Отговор**: Redirect към `/` при успешен вход

---

### POST /logout

**Описание**: Изход от системата

**Request**: Празен POST request (CSRF token автоматично)

**Отговор**: Redirect към `/`

```

```

---

## Защитени маршрути

### GET /history

**Описание**: История на препоръките за вписани потребители

**Laravel Route**:
```php
Route::middleware('auth')->group(function () {
 Route::get('/history', [HistoryController::class, 'index'])->name('history');
});
```

**Controller**:
```php
public function index()
{
 $history = DecisionNode::where('user_id', auth()->id())
 ->orderBy('completed_at', 'desc')
 ->get();

 return Inertia::render('History', [
 'decisions' => $history,
]);
}
```

**Props**:
```javascript
{
 decisions: [
 {
 id: 1,
 country: 'BG',
 recommended_model: 'BG-3',
 decision_path: [...],
 completed_at: '2026-01-31T12:00:00Z'
 }
]
}
```

---

### GET /profile

**Описание**: Профил на потребителя (Laravel Breeze)

**Laravel Route**:
```php
Route::middleware('auth')->group(function () {
 Route::get('/profile', [ProfileController::class, 'edit'])->name('profile.edit');
 Route::patch('/profile', [ProfileController::class, 'update'])->name('profile.update');
});
```

```

```

Route::delete('/profile', [ProfileController::class, 'destroy'])->
name('profile.destroy');
});
```

```

**\*\*PUT/PATCH /profile\*\*:** Актуализиране на профил

**\*\*DELETE /profile\*\*:** Изтриване на акаунт

---

## Формат на данните

### Формат на въпрос

```

```javascript
{
  id: 1,
  text: 'Текст на въпроса?',
  category: 'budget', // Опционално: budget, experience, complexity
  answers: [
    {
      id: 1,
      text: 'Отговор 1'
    },
    {
      id: 2,
      text: 'Отговор 2'
    }
  ]
}
```

```

### Формат на препоръка

```

```javascript
{
  model: 'BG-3',
  title: 'Модел BG-3: Генерално изпълнение',
  description: 'Подробно описание на модела...',
  advantages: [
    'Преимущество 1',
    'Преимущество 2'
  ],
  disadvantages: [
    'Недостатък 1'
  ],
  regulations: 'Съгласно ЗОП, чл. 123...'
}
```

```

### Формат на decision trace

```

```javascript
[
  {
    question: 'Каква е стойността на проекта?',
    answer: 'Над 500 000 лв.',
    questionId: 1,
  }
]
```

```

```

 answerId: 2
 },
 {
 question: 'Имате ли опит?',
 answer: 'Не',
 questionId: 2,
 answerId: 4
 }
]
 },
 ...

Обработка на грешки

Validation грешки

Laravel отговор::
```php
// Автоматично от Inertia при validation failure
return back()->withErrors([
    'email' => 'Невалиден имейл адрес',
]);
```

Vue получава::
```javascript
{
  errors: {
    email: 'Невалиден имейл адрес'
  }
}
```

Използване във Vue::
```vue
<template>
  <form @submit.prevent="submit">
    <input v-model="form.email" />
    <div v-if="form.errors.email">{{ form.errors.email }}</div>
  </form>
</template>

<script setup>
import { useForm } from '@inertiajs/vue3';

const form = useForm({
  email: '',
});

const submit = () => {
  form.post('/login');
};
</script>
```

HTTP грешки

```

| Код | Описание              | Причина                           |
|-----|-----------------------|-----------------------------------|
| 419 | CSRF Token Mismatch   | Изтекъл CSRF token                |
| 422 | Validation Error      | Невалидни данни                   |
| 429 | Too Many Requests     | Rate limiting (throttle)          |
| 500 | Internal Server Error | Сървърна грешка                   |
| 503 | Service Unavailable   | Приложението е в maintenance mode |

**\*\*Обработка във Vue\*\*:**

```

```javascript
form.post('/wizard/answer', {
  onError: (errors) => {
    console.error('Грешка:', errors);
  },
  onSuccess: (page) => {
    console.log('Успешно!');
  },
});
```

```

---

## Rate Limiting

Всички wizard маршрути имат rate limiting:

```

```php
->middleware('throttle:60,1') // 60 заявки на минута
```

```

При превишаване на лимита:

- HTTP 429 Too Many Requests
- Header: `Retry-After: 60` (секунди)

---

## Примерна употреба във Vue

### Стартиране на wizard

```

```vue
<script setup>
import { router } from '@inertiajs/vue3';

const startWizard = (country) => {
  router.post('/wizard/start', {
    country: country,
  });
};
</script>

<template>
  <button @click="startWizard('BG')">България</button>
  <button @click="startWizard('DE')">Бавария</button>
</template>
```

```

### Изпращане на отговор

```

````vue
<script setup>
import { useForm } from '@inertiajs/vue3';

const props = defineProps({
  question: Object,
});

const form = useForm({
  questionId: props.question.id,
  answerId: null,
});

const submitAnswer = (answerId) => {
  form.answerId = answerId;
  form.post('/wizard/answer');
};
</script>

<template>
  <div>
    <h2>{{ question.text }}</h2>
    <button
      v-for="answer in question.answers"
      :key="answer.id"
      @click="submitAnswer(answer.id)"
    >
      {{ answer.text }}
    </button>
  </div>
</template>
````

```

---

## ## Заключение

ARCHModels не използва традиционен REST API. Вместо това, Inertia.js осигурява:

1. **\*\*Директна комуникация\*\*** между Laravel контролери и Vue компоненти
2. **\*\*Автоматично валидиране\*\*** и error handling
3. **\*\*SPA преживяване\*\*** без нужда от отделен API layer
4. **\*\*Опростена разработка\*\*** с Laravel routing и Vue components

За повече информация за Inertia.js, вижте [официалната документация](https://inertiajs.com/).

© Дипл.-инж., арх. Стоян Тодоров