MASARYKOVA
UNIVERZITA

FAKULTA INFORMATIKY

E-learningová aplikace pro přípravu na zkoušky žadatele o zbrojní oprávnění

Bakalářská práce

LUKÁŠ LANČA

Brno, jaro 2026

MASARYKOVA
UNIVERZITA

FAKULTA INFORMATIKY

E-learningová aplikace pro přípravu na zkoušky žadatele o zbrojní oprávnění

Bakalářská práce

LUKÁŠ LANČA

Vedoucí práce: RNDr. Jaroslav Ráček, Ph.D.

Brno, jaro 2026


FACULTAS ARTIS INFORMATICÆ
UNIVERSITAS MASARYKIANA

Prohlášení

Prohlašuji, že tato bakalářská práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

Při zpracování této práce jsem využil nástroje umělé inteligence, konkrétně ChatGPT, Gemini a GitHub Copilot. ChatGPT jsem použil zejména k pravopisné korektuře, dílčím stylistickým úpravám, úpravám slovosledu textu, generování ikon aplikace a podpoře při tvorbě programového kódu. Nástroj Gemini byl využit jako pomocný prostředek při tvorbě a kontrole návrhů vysvětlení k testovým otázkám. GitHub Copilot jsem využil především jako podpůrný nástroj při implementaci programové části práce. Výstupy těchto nástrojů nebyly přebírány automaticky. Veškerý obsah práce i výsledné aplikace jsem následně zkontroloval, upravil podle potřeby a přebírám za něj plnou odpovědnost. Tyto nástroje jsem používal v souladu s principy akademické integrity.

Lukáš Lanča

Vedoucí práce: RNDr. Jaroslav Ráček, Ph.D.

Poděkování

Rád bych poděkoval vedoucímu své bakalářské práce za odborné vedení, cenné připomínky, trpělivost a vstřícný přístup při konzultacích během zpracování této práce. Dále děkuji své rodině a blízkým za podporu, motivaci a pochopení po celou dobu studia.

Poděkování patří také všem, kteří mi poskytli zpětnou vazbu při návrhu, testování a vyhodnocování výsledné aplikace.

Shrnutí

Tato bakalářská práce se zabývá návrhem a implementací e-learningové webové aplikace určené pro přípravu na zkoušku odborné způsobilosti žadatele o zbrojní oprávnění podle legislativy účinné od roku 2026. V práci je provedena rešerše současných webových aplikací zaměřených na přípravu k této zkoušce, zejména z hlediska nabízených funkcí, způsobu práce s testovými otázkami a uživatelského rozhraní. Na základě zjištěných poznatků a analýzy potřeb cílových uživatelů jsou stanoveny požadavky na navrhované řešení.

Praktická část práce se věnuje návrhu a realizaci responzivní progresivní webové aplikace (PWA), která umožňuje procvičování oficiálních testových otázek, jejich filtrování podle tematických oblastí, spuštění cvičného testu a simulaci reálné zkoušky včetně automatického vyhodnocení výsledků. Součástí aplikace je také režim chytrého opakování využívající algoritmus FSRS (Free Spaced Repetition Scheduler, plánovač rozloženého opakování), který přizpůsobuje opakování otázek dosavadním odpovědím uživatele.

Aplikace podporuje uživatelské účty, ukládání postupu, synchronizaci mezi zařízeními a přihlášení pomocí externích poskytovatelů identity Google a Microsoft. Součástí řešení je administrační rozhraní pro správu databáze otázek a odpovědí. Důraz byl kladen na offline dostupnost, vysokou použitelnost na mobilních i stolních zařízeních a testování. Funkčnost aplikace byla úspěšně ověřena v rámci testovacího nasazení reálnými uživateli.

Klíčová slova

zbrojní oprávnění, zkouška odborné způsobilosti, e-learning, progresivní webová aplikace, simulace zkoušky, rozložené opakování, FSRS

Obsah

Úvod **1**

1 Východiska a rešerše současných řešení **3**
1.1 Zbrojní oprávnění a zkouška odborné způsobilosti od roku 2026 . . . 3
1.2 Obsahový základ přípravy na teoretickou část . . . 4
1.3 Současná digitální řešení přípravy na zbrojní oprávnění 5
1.4 Limity současných řešení . . . 6
1.5 Spaced repetition a algoritmické plánování opakování . 7
1.5.1 Přehled používaných přístupů ke spaced repetition . . . 8
1.5.2 Volba algoritmu FSRS . . . 10
1.5.3 Křivka zapomínání a opakování s odstupem . . 11
1.6 Progresivní webová aplikace . . . 12
1.7 Shrnutí a vymezení prostoru pro navrhované řešení . . 13

2 Analýza požadavků **14**
2.1 Funkční požadavky . . . 14
2.2 Nefunkční požadavky . . . 15
2.3 Omezení a předpoklady řešení . . . 17
2.4 Případy užití . . . 17
2.4.1 Aktéři . . . 17
2.4.2 Přehled případů užití . . . 19
2.4.3 Popis hlavních případů užití . . . 19

3 Návrh aplikace **22**
3.1 Databázový model . . . 22
3.1.1 Obsahová data . . . 23
3.1.2 Legislativní podklady . . . 23
3.1.3 Uživatelská data a FSRS . . . 24
3.1.4 Testové pokusy . . . 24
3.1.5 Technické tabulky . . . 25
3.2 Návrh uživatelského rozhraní . . . 25
3.2.1 Úvodní stránka . . . 25
3.2.2 Obrazovka procvičování . . . 26
3.2.3 Obrazovka simulace zkoušky . . . 28

4 Implementace aplikace **29**
4.1 Použité technologie a nástroje 29
4.1.1 Frontend 29
4.1.2 Backend a aplikační logika 30
4.1.3 Databáze 31
4.1.4 Autentizace a autorizace 31
4.2 Vývojové prostředí 33
4.2.1 Laravel Sail a Docker 33
4.2.2 Vite 33
4.3 Testování během vývoje 34
4.4 Struktura projektu 35
4.4.1 Základní adresářová struktura 35
4.4.2 Backendová část 35
4.4.3 Databázová vrstva 36
4.4.4 Frontendová část 36
4.4.5 Klientské služby 36
4.4.6 PWA a veřejné soubory 37
4.4.7 Routování 37
4.5 Procvičování otázek 37
4.6 Simulace zkoušky 38
4.7 Uživatelský účet 38
4.8 Ukládání postupu a synchronizace 39
4.9 Administrační rozhraní 39
4.10 Podpora progresivní webové aplikace 40
4.11 Příprava a import obsahových dat 41
4.11.1 Import testových otázek 41
4.11.2 Import legislativních podkladů 42
4.11.3 Import vysvětlení otázek 44

5 Ověření, nasazení a zhodnocení **45**
5.1 Ověření funkčnosti 45
5.2 Měření pomocí Lighthouse 45
5.3 Nasazení 46
5.3.1 Hosting 46
5.3.2 Zpřístupnění aplikace pomocí Cloudflare Tunnel 47
5.4 Uživatelské testování 47
5.5 Dokumentace a licence 49

Závěr **50**

Bibliografie **51**

A Přílohy **57**

Seznam obrázků

1.1 Schematické znázornění křivky zapomínání a spaced repetition 11
2.1 Diagram případů užití aplikace 18
3.1 Diagram vztahů mezi entitami 22
3.2 Wireframe úvodní stránky aplikace 26
3.3 Wireframe obrazovky procvičování 27
3.4 Wireframe obrazovky simulace zkoušky 28
4.1 Zjednodušená struktura projektu 35
A.1 Úvodní obrazovka při první návštěvě. 57
A.2 Úvodní obrazovka přihlášeného uživatele se staženými daty pro offline funkcionalitu. 58
A.3 Procházení databáze testových otázek. 58
A.4 Procvičování testových otázek. 59
A.5 Volitelná nastavení procvičování. 59
A.6 Úprava nastavení procvičování podle zpětné vazby od uživatelů. Je zapnuté ovládání klávesnicí a zobrazení časů dalších opakování. 60
A.7 Vyhodnocení relace procvičování. 60
A.8 Průběh simulace zkoušky. 61
A.9 Vyhodnocení simulace zkoušky. 61
A.10 Přehled pokroku a historie zkoušek. 62
A.11 Přehled studijního postupu uživatele. 62
A.12 Přehled legislativních podkladů. 63
A.13 Zobrazení podrobnosti vyhlášky a možnost kopírovat odkaz na danou část. 63
A.14 Administrační rozhraní správy otázek. 64
A.15 Administrační rozhraní správy uživatelů. 65

Úvod

Zkouška odborné způsobilosti pro získání zbrojního oprávnění představuje jednu z klíčových podmínek pro legální nabytí a držení zbraně. Její součástí je ověření teoretických znalostí prostřednictvím testových otázek, jejichž zvládnutí je důležité nejen pro úspěšné složení zkoušky, ale také pro bezpečné zacházení se zbraněmi a střelivem v praxi.

V současné době mohou uchazeči při přípravě využívat různé zdroje, například webové stránky, tištěné materiály nebo jednoduché testovací nástroje. Tato řešení však často neposkytují jednotné, přehledné a uživatelsky přívětivé prostředí pro systematickou přípravu. Pro uchazeče je přitom klíčové nejen procvičovat jednotlivé otázky, ale také sledovat vlastní pokrok, identifikovat slabá místa a ověřit si připravenost v podmínkách podobných reálné zkoušce.

Cílem této bakalářské práce je na základě rešerše současných aplikací navrhnout a implementovat webovou aplikaci pro přípravu na teoretickou část zkoušky odborné způsobilosti pro získání zbrojního oprávnění. Aplikace umožňuje procházení a filtrování oficiálních testových otázek podle tematických oblastí, spouštění cvičných testů a simulaci reálné zkoušky s automatickým vyhodnocením výsledků. Součástí řešení je také uživatelský účet, ukládání a synchronizace postupu mezi zařízeními, adaptivní opakování otázek pomocí algoritmu FSRS, offline režim a administrační rozhraní pro správu databáze otázek a odpovědí.

Postup řešení vycházel z rešerše platné legislativy, oficiálních testových otázek a existujících aplikací pro přípravu na zkoušku. Na základě této rešerše byly stanoveny funkční a nefunkční požadavky, navržen datový model, uživatelské rozhraní a technická architektura aplikace. Následně byla aplikace implementována jako progresivní webová aplikace s využitím technologií Laravel, React, TypeScript a MariaDB. Podpora offline režimu byla řešena pomocí service workeru, Cache API a IndexedDB.

Výsledkem práce je funkční progresivní webová aplikace, která uchazečům poskytuje ucelené prostředí pro přípravu na teoretickou část zkoušky odborné způsobilosti. Funkčnost řešení byla ověřena automatizovanými testy, ručním testováním hlavních scénářů a uživatelským testováním.

Práce je členěna do pěti kapitol. První kapitola vymezuje východiska práce, zejména problematiku zbrojního oprávnění, průběh zkoušky odborné způsobilosti, obsahový základ přípravy, rešerši současných digitálních řešení a jejich limity. Součástí je také shrnutí principu rozloženého opakování, volby algoritmu FSRS a zdůvodnění využití progresivní webové aplikace. Druhá kapitola se věnuje analýze funkčních a nefunkčních požadavků včetně případů užití. Třetí kapitola popisuje návrh aplikace, především databázový model a návrh uživatelského rozhraní. Čtvrtá kapitola se zaměřuje na implementaci aplikace, včetně použitých technologií, vývojového prostředí, struktury projektu a přípravy obsahových dat pro jejich následný import do systému. Dále popisuje realizaci hlavních částí aplikace, jako jsou procvičování otázek, simulace zkoušky, uživatelský účet, synchronizace postupu, administrační rozhraní a podpora progresivní webové aplikace. Pátá kapitola popisuje ověření funkčnosti, měření pomocí nástroje Lighthouse, nasazení aplikace do produkčního prostředí, dokumentaci a licenci výsledného řešení.

1 Východiska a rešerše současných řešení

Od 1. ledna 2026 je v České republice účinná nová právní úprava nakládání se zbraněmi a střelivem, tvořená zejména zákonem č. 90/2024 Sb., o zbraních a střelivu, a navazujícími prováděcími předpisy [1, 2].

Nová úprava neznamená pouze dílčí změnu terminologie, ale představuje širší koncepční proměnu celé agendy. Ministerstvo vnitra v souvislosti s účinností nové právní úpravy zdůrazňuje zejména vyšší míru digitalizace, navázání evidence oprávnění a zbraní na centrální registr zbraní a omezení fyzických dokladů ve prospěch elektronicky vedených údajů [3, 4, 5].

Nová právní úprava mění nejen obsah zkoušky, ale i prostředí, v němž uchazeč se státem komunikuje. Digitální studijní nástroj proto odpovídá oblasti, která se stále více opírá o elektronické evidence, online informace a digitalizované procesy [3, 5].

1.1 Zbrojní oprávnění a zkouška odborné způsobilosti od roku 2026

Zákon č. 90/2024 Sb., o zbraních a střelivu, účinný od 1. ledna 2026, nově pracuje s pojmem „zbrojní oprávnění“, které se dělí na obecné a rozšířené. Rozšířené zbrojní oprávnění v sobě současně zahrnuje i oprávnění obecné [1]. Tato změna nahrazuje dřívější systém skupin zbrojního průkazu a z pohledu uživatele i navrhovaného softwarového řešení zjednodušuje základní členění oprávnění, pro něž se uchazeč připravuje.

Odborná způsobilost se i nadále ověřuje zkouškou, která má teoretickou a praktickou část. Pro tuto práci je rozhodující zejména část teoretická, protože právě ta je vhodná pro podporu prostřednictvím digitálního studijního nástroje.

Zákon stanoví, že teoretická část je vykonávána formou písemného testu o 30 otázkách v délce nejvýše 40 minut. K úspěšnému složení je třeba správně zodpovědět alespoň 26 otázek v případě obecného zbrojního oprávnění a alespoň 28 otázek v případě rozšířeného zbrojního oprávnění [1].

Podrobnější strukturu testu stanovuje nařízení vlády č. 238/2025 Sb. Podle něj se test skládá ze 17 otázek z právních předpisů a souvisejících oblastí, 5 otázek z nauky o zbraních a střelivu, 5 otázek ze zásad bezpečného nakládání se zbraněmi a střelivem a 3 otázek ze zdravotnického minima. U každé otázky se volí jedna správná odpověď ze tří možností [2].

Tato struktura je pro navrhovanou aplikaci klíčová, protože určuje nejen obsah, ale i jasný vzor realistické simulace zkoušky. Kvalitní přípravný systém proto nemá pouze zpřístupnit otázky, ale musí umět respektovat oficiální parametry testu včetně počtu otázek, tematického rozložení, časového limitu a způsobu hodnocení.

Současně je třeba jasně vymezit hranici navrhovaného řešení. Digitální aplikace může smysluplně podporovat především přípravu na teoretickou část zkoušky, případně zprostředkovat základní orientaci v požadavcích části praktické. Nemůže však nahradit samotný výkon zkoušky ani praktický výcvik bezpečné manipulace a střelby, které zůstávají vázány na prezenční a kontrolované prostředí [1, 6].

1.2 Obsahový základ přípravy na teoretickou část

Pro návrh nástroje je zásadní odlišit primární a sekundární zdroje studijního obsahu. Primární normativní základ tvoří samotný zákon, prováděcí předpisy a zejména oficiální soubor testových otázek, z něhož jsou sestavovány testy pro teoretickou část zkoušky odborné způsobilosti. Ministerstvo vnitra k nové zbraňové legislativě výslovně uvádí, že zveřejňuje kompletní soubor testových otázek s vyznačenými správnými odpověďmi, platný od 1. 1. 2026, a že právě z tohoto souboru jsou sestavovány testy ke zkoušce [5, 7]. Pro navrhovanou práci je to zásadní východisko: obsah aplikace nesmí být založen na neoficiální interpretaci, ale musí vycházet z oficiálně publikovaného zdroje.

Vedle toho existují zdroje, které oficiální obsah zpřístupňují v uživatelsky přívětivější podobě. Komunitní a tématicky orientované weby, například ZbraněKvalitně.cz [8], nabízejí přehledy otázek, tréninkové režimy nebo tematické členění. Nejsou však normativní oporou řešení, spíše dokládají, jak je oficiální obsah zprostředkováván a jaké funkce uživatelé očekávají.

1.3 Současná digitální řešení přípravy na zbrojní oprávnění

Pro přípravu na zkoušku již existují webové portály, mobilní aplikace i komunitní weby. Následující část porovnává jejich funkce a limity ve vztahu k navrhované aplikaci.

První skupinu představují webové testovací portály. Typickým příkladem je zpoplatněná služba eTesty, která nabízí online testy pro obecné i rozšířené zbrojní oprávnění podle pravidel účinných od roku 2026 [9, 10]. Výhodou těchto řešení je bezprostřední dostupnost v prohlížeči a nízká bariéra vstupu. Zároveň bývají dobře použitelné pro rychlé procvičování a pro základní simulaci testu.

Druhou skupinu tvoří nativní mobilní aplikace. Na platformách Google Play a App Store je dostupná řada aplikací deklarující přípravu na teoretickou část zkoušky podle pravidel platných od 1. 1. 2026. Popisy těchto aplikací často zdůrazňují práci s aktuální sadou otázek, možnost offline použití, historii výsledků, test nanečisto nebo doprovodné vysvětlivky k odpovědím. [11, 12, 13] To ukazuje, že uživatelská poptávka nesměřuje pouze k samotnému testování, ale také k pohodlí mobilního používání a k průběžnému studiu mimo stálé internetové připojení.

Třetí skupinu představují komunitní nebo obsahově orientované weby, které kombinují přehledy legislativy, otázky, návody k přípravě a často i širší informační servis pro držitele zbraní a uchazeče [8]. Tato řešení mají výhodu v šíři doprovodného obsahu, ale jejich primárním cílem nebývá budování konzistentní aplikační architektury.

Současný stav tedy lze charakterizovat jako funkčně využitelný, avšak heterogenní. Jednotlivá řešení se liší platformou, obchodním modelem, rozsahem funkcí, úrovní vysvětlujícího obsahu i způsobem, jak deklarují návaznost na oficiální zdroje a aktuální verzi testových otázek. Pro účely porovnání byli vybráni hlavní zástupci dostupní prostřednictvím webového vyhledávání a distribučních platforem App Store a Google Play, tedy řešení, se kterými se může běžný uživatel při hledání přípravy na test nejpravděpodobněji setkat. Následující porovnání vybraných platforem proto slouží k přehlednému zachycení těchto rozdílů a k vymezení prostoru pro navrhované řešení.

Tabulka 1.1: Komparativní analýza funkcí aplikací pro přípravu na teoretickou část zkoušky odborné způsobilosti k získání zbrojního oprávnění.

Aplikace	Platforma			Testování a procvičování				Obsah a studium			Pokročilé funkce		
	Web	*Android*	*iOS*	*Oficiální 2026 otázky*	*Test nanečisto*	*Okamžitá zpětná vazba*	*Statistiky / historie*	*Vysvětlení odpovědí*	*Zákony / materiály*	*Označení / filtr otázek*	*Offline režim*	*Synchronizace účtu*	*Opakování s odstupem*
eTesty	●	●	●	◐	◐	–	●	–	–	●	◐	◐	–
eTesty Premium	●	●	●	●	●	–	●	–	–	●	◐	◐	–
ZbraněKvalitně.cz	●	–	–	●	●	●	–	–	●	–	–	–	–
ZBRANĚ.cz	●	–	–	●	●	●	–	–	–	◐	–	–	–
Zbrojní Průkaz Plus 2026 (Metastream)	–	●	●	●	●	◐	●	–	–	◐	●	–	●
Zbrojní Průkaz Pro 2026 (Metastream)	–	●	●	●	●	◐	●	–	–	◐	●	–	●
Zbrojní oprávnění – 2026 (M. Černý)	–	●	–	●	●	●	●	–	●	●	●	–	–
Zbrojní průkaz 2026 (James Deer)	–	●	●	●	◐	●	●	–	●	–	●	–	–
Zbrojní průkaz 2026 Prémium (James Deer)	–	●	–	●	◐	●	●	–	●	–	●	–	–
Zbrojní oprávnění 2026 (Aleš Kozubík)	–	●	●	●	●	●	●	◐	●	●	●	–	–
testy-zbrojni-prukaz.cz	●	–	–	●	●	●	●	–	–	●	–	–	●

● = vlastnost je plně přítomná; ◐ = vlastnost je přítomná částečně, omezeně nebo jen na části platforem; – = vlastnost není přítomná nebo nebyla veřejně doložena.

1.4 Limity současných řešení

Přestože existující nástroje pokrývají základní potřebu procvičování testových otázek, samotná možnost testování nepředstavuje dostatečné kritérium kvality. Při návrhu robustního a odborně obhajitelného řešení je nutné zohlednit také práci s obsahem, technické provedení a podporu studijního procesu. Z těchto hledisek lze u současných řešení identifikovat několik opakujících se omezení.

Prvním omezením je roztříštěnost mezi platformami a distribučními modely. Uživatel si typicky vybírá mezi webovým portálem a nativní mobilní aplikací. Webová řešení jsou snadno dostupná bez instalace, ale jejich podpora offline provozu, možnost připnutí na plochu zařízení nebo chování při slabé konektivitě nebývají vždy výslovně uvedeny. Nativní aplikace naopak nabízejí dobrý mobilní komfort, ale za cenu oddělené distribuce a zpravidla i odděleného vývoje pro jednotlivé platformy [9, 11, 13].

Druhým omezením je nejednotná míra transparentnosti vůči primárním zdrojům. U části existujících řešení je zřejmé, že vycházejí z aktuálních otázek pro rok 2026, avšak není vždy stejně transparentní, z jakého konkrétního oficiálního souboru vycházejí, jaká je verze obsahu a jak je zajištěna aktualizace při opravách nebo změnách [5, 7]. V regulované oblasti, jakou je zbraňová legislativa, je přitom vazba na oficiální zdroj zásadní.

Třetím omezením je slabší podpora porozumění oproti pouhému drilu. Mnoho řešení se soustředí primárně na výběr správné odpovědi a dosažení skóre v modelovém testu. Z hlediska dlouhodobější studijní efektivity je však důležité, aby systém podporoval i pochopení souvislostí, například vazbou otázky na relevantní právní ustanovení, tematickým členěním nebo vysvětlením, proč je správná právě jedna z odpovědí. To je zvláště významné po rozsáhlé legislativní změně účinné od 1. 1. 2026, kdy se uchazeč neorientuje pouze ve „znění otázky“, ale i v novém systémovém uspořádání celé oblasti [1, 2, 3].

1.5 Spaced repetition a algoritmické plánování opakování

Samotná dostupnost databáze testových otázek ještě sama o sobě nezaručuje efektivní přípravu. Uchazeč může otázky opakovaně procházet nebo vyplňovat modelové testy, avšak bez systematického plánování hrozí, že bude zbytečně často procvičovat již zvládnuté otázky a naopak se nebude včas vracet k otázkám, u nichž dochází k zapomínání. Digitální studijní nástroj by proto neměl fungovat pouze jako statický seznam otázek nebo jako simulátor testu, ale měl by podporovat také řízené opakování učiva v čase.

Teoretickým východiskem tohoto přístupu je princip *spaced repetition*, tedy opakování s odstupem. Ten navazuje na výzkum zapomínání a na tzv. křivku zapomínání, podle níž pravděpodobnost vybavení informace po naučení postupně klesá, pokud není paměťová stopa znovu posílena [14]. Smyslem opakování s odstupem není opakovat látku stále stejným tempem, ale plánovat další setkání s položkou přibližně do okamžiku, kdy je ještě vybavitelná, avšak již dostatečně oslabená na to, aby její úspěšné vybavení vedlo k posílení dlouhodobé paměti.

Účinnost rozloženého učení je dobře podložena v kognitivní psychologii. Meta-analýza Cepedy a kol. ukazuje, že distribuovaná praxe vede k lepšímu dlouhodobému uchování informací než nahuštěné učení v krátkém čase [15]. Podobně Dunlosky a kol. řadí distribuovanou praxi a procvičovací testování mezi studijní techniky s vysokou užitečností napříč různými typy učiva i skupinami studentů [16]. Pro testové otázky je důležitý také princip aktivního vybavování, protože uživatel se neučí pouze pasivním čtením správných odpovědí, ale musí odpověď opakovaně vybavit a použít. Výzkum retrieval practice ukazuje, že aktivní vybavování může být pro dlouhodobé učení účinnější než opakované čtení [17, 18].

1.5.1 Přehled používaných přístupů ke spaced repetition

V digitálních výukových systémech se spaced repetition realizuje pomocí různě složitých algoritmů. Jejich společným cílem je rozhodnout, kdy má být konkrétní položka uživateli znovu předložena. Jednotlivé přístupy se však liší tím, zda pracují pouze s pevně danými intervaly, jednoduchými kategoriemi úspěšnosti, nebo s adaptivním modelem paměti.

Mezi historicky významné přístupy patří Pimsleurův model, který pracuje s předem danou posloupností intervalů. Jeho výhodou je jednoduchost, nevýhodou však nízká míra personalizace. Na něj navazuje Leitnerův systém, v němž jsou kartičky rozděleny do několika přihrádek. Položky zodpovězené správně se přesouvají do přihrádek s delším intervalem opakování, zatímco chybně zodpovězené položky se vracejí zpět. Leitnerův systém je snadno pochopitelný a implementovatelný, ale pracuje pouze s hrubým rozlišením úspěšnosti a neodhaduje přímo pravděpodobnost zapomenutí [19].

Další významnou skupinu představují algoritmy rodiny SuperMemo, zejména SM-2. Algoritmus SM-2 pracuje s tzv. faktorem snadnosti, který se upravuje podle kvality odpovědi a ovlivňuje délku dalších intervalů [20, 21]. Tento přístup byl dlouhodobě používán v řadě digitálních kartičkových systémů, včetně starších plánovačů v aplikaci Anki [22]. Oproti Leitnerovu systému přináší jemnější práci s obtížností položky, stále však jde převážně o heuristický přístup, nikoli o explicitní model pravděpodobnosti vybavení.

Modernější systémy se snaží modelovat stav paměti přímo. Příkladem je HLR (*Half-Life Regression*), který byl použit ve výzkumu jazykového vzdělávání a odhaduje poločas zapomínání konkrétní položky na základě historie odpovědí uživatele [19]. Podobným směrem se ubírají optimalizační a adaptivní modely, které plánují opakování podle cílové retence a průběžně reagují na výkon studenta [23, 24]. Do této skupiny patří také FSRS (*Free Spaced Repetition Scheduler*), zvolený pro navrhovanou aplikaci.

Pro přehledné srovnání lze hlavní přístupy shrnout následovně:

Pimsleurův model
Používá předem danou posloupnost intervalů mezi opakováními. Je jednoduchý, ale má nízkou míru adaptace na konkrétního uživatele a konkrétní položku.

Leitnerův systém
Přesouvá položky mezi přihrádkami podle správnosti odpovědi. Vyšší přihrádky odpovídají delším intervalům opakování. Systém je snadno pochopitelný a implementovatelný, ale pracuje pouze s hrubým členěním znalosti a neodhaduje přímo pravděpodobnost vybavení.

SM-2
Upravuje intervaly pomocí faktoru snadnosti podle kvality předchozích odpovědí. Oproti Leitnerovu systému umožňuje jemnější plánování, stále však jde převážně o heuristický model, který nepracuje přímo s cílovou retencí.

HLR
Odhaduje poločas zapomínání položky pomocí trénovaného modelu na základě historie odpovědí uživatele. Tento přístup je datově orientovaný a umožňuje přesnější predikci zapomínání, vyžaduje však kvalitní historická data.

FSRS
Pracuje s modelem obtížnosti, stability a vybavitelnosti paměťové stopy. V navrhované aplikaci využívá hodnocení odpovědi pomocí možností „Znovu“, „Těžké“, „Dobré“ a „Snadné“, podle nichž stanovuje následný interval opakování. Oproti jednodušším systémům je implementačně náročnější, ale umožňuje přesnější personalizaci studijního plánu.

1.5.2 Volba algoritmu FSRS

Pro navrhovanou aplikaci byl zvolen algoritmus FSRS, protože lépe odpovídá cíli práce než jednoduché opakování otázek nebo pevně dané intervaly. FSRS je otevřený algoritmus pro plánování opakování, který vychází z modelování tří vlastností paměťové stopy: obtížnosti, stability a vybavitelnosti [25]. Obtížnost vyjadřuje, jak náročné je danou položku osvojit, stabilita popisuje odolnost paměťové stopy vůči zapomínání a vybavitelnost představuje pravděpodobnost, že si uživatel položku v daném okamžiku správně vybaví.

Důležitou vlastností FSRS je práce s cílovou retencí. Aplikace tak nemusí pouze pasivně zaznamenávat, zda uživatel odpověděl správně nebo špatně, ale může plánovat další opakování tak, aby se uživatel pohyboval kolem požadované pravděpodobnosti zapamatování. V navrhované aplikaci bude FSRS využit v podobě, kdy uživatel po zodpovězení otázky neobdrží pouze informaci o správnosti odpovědi, ale současně subjektivně vyhodnotí obtížnost vybavení pomocí čtyř možností: „Znovu“, „Těžké“, „Dobré“ a „Snadné“. Toto hodnocení odpovídá běžnému způsobu práce ve spaced repetition systémech a slouží algoritmu k určení následného intervalu opakování [25].

Volba „Znovu“ znamená, že uživatel položku neovládá nebo ji nedokázal správně vybavit, a proto má být zopakována v krátkém intervalu. Volba „Těžké“ vyjadřuje, že odpověď byla sice vybavena, ale s výraznou nejistotou nebo obtížemi. Volba „Dobré“ představuje standardní úspěšné vybavení a volba „Snadné“ signalizuje, že je položka pro uživatele již dobře osvojená. Na základě tohoto vstupu FSRS upravuje odhad obtížnosti, stability a vybavitelnosti dané otázky a stanovuje vhodný termín dalšího opakování.

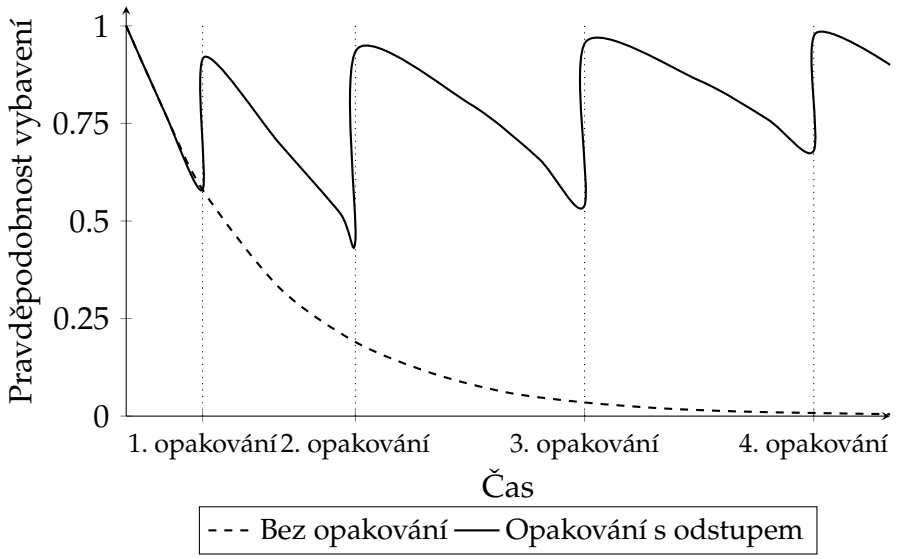
Tento způsob je pro přípravu na zkoušku vhodnější než pouhé rozlišování správné a nesprávné odpovědi. U testových otázek se totiž může stát, že uživatel odpoví správně náhodným odhadem nebo s nejistotou. Samotná správnost odpovědi by v takovém případě mohla vést k příliš dlouhému intervalu dalšího opakování. Dodatečné hodnocení obtížnosti proto poskytuje algoritmu jemnější informaci o skutečném stavu znalosti a umožňuje lépe odlišit otázky, které jsou již bezpečně zvládnuté, od otázek, které je vhodné brzy zopakovat.

Z rešerše současných aplikací pro přípravu na zbrojní oprávnění vyplývá, že veřejně prezentované funkce se zpravidla soustředí

zejména na procvičování otázek, simulaci testu, historii výsledků, označování otázek a případně offline režim [8, 9, 11, 13]. U některých nástrojů je sice princip spaced repetition zmíněn, avšak veřejné popisy obvykle nerozvádějí konkrétní použitý paměťový model ani způsob optimalizace intervalů [26, 27]. Z tohoto důvodu je vhodnější netvrdit, že ostatní aplikace chytré plánování vůbec nepoužívají, ale že u analyzovaných řešení není sofistikované a transparentně popsané plánování opakování běžnou nebo jasně doloženou vlastností.

Navrhovaná aplikace se proto od existujících řešení neodlišuje pouze zvolenou platformou nebo zpracováním testových otázek, ale také didaktickým způsobem práce s učením. Namísto toho, aby uživateli pouze nabízela další náhodné otázky nebo opakovaný test nanečisto, využívá FSRS k plánování návratu k jednotlivým otázkám podle jejich skutečné obtížnosti pro konkrétního uživatele. Tím se aplikace přibližuje moderním adaptivním studijním systémům, jejichž cílem je nejen zpřístupnit obsah, ale také optimalizovat proces jeho dlouhodobého zapamatování.

1.5.3 Křivka zapomínání a opakování s odstupem



Obrázek 1.1: Schematické znázornění křivky zapomínání a spaced repetition

Princip spaced repetition lze znázornit pomocí schematické křivky zapomínání. Bez opakování pravděpodobnost vybavení informace postupně klesá. Pokud je však informace ve vhodných intervalech znovu aktivně vybavena, paměťová stopa se posílí a další pokles je pomalejší. Algoritmy spaced repetition se proto snaží plánovat opakování přibližně do okamžiku, kdy znalost začíná slábnout, ale ještě nedošlo k úplnému zapomenutí [14, 15].

Přerušovaná křivka na obrázku 1.1 znázorňuje postupné zapomínání informace bez dalšího procvičování. Plná křivka představuje situaci, kdy dochází k opakování ve vhodných časových intervalech. Každé úspěšné vybavení informace vede k opětovnému posílení paměťové stopy a současně prodlužuje interval do dalšího opakování. Výsledkem je pomalejší pokles pravděpodobnosti vybavení a efektivnější dlouhodobé uchování znalostí.

1.6 Progresivní webová aplikace

Progresivní webová aplikace (PWA, z anglického „Progressive Web App“) kombinuje dostupnost webové aplikace s vybranými vlastnostmi nativních aplikací, zejména instalovatelností, jednotným kódovým základem a podporou offline režimu [28]. Z technického hlediska je pro tuto práci důležitý zejména service worker, který umožňuje řídit cache a zpřístupnit vybrané části aplikace i při omezeném nebo chybějícím připojení k internetu [29, 30].

Tento přístup odpovídá potřebám navrhované aplikace. Uživatel potřebuje nástroj dostupný na telefonu, tabletu i stolním počítači, aplikace však nevyžaduje hlubokou integraci s operačním systémem ani specifickým hardwarem zařízení. PWA proto umožňuje zachovat jeden kódový základ, nabídnout částečný offline režim a současně průběžně aktualizovat obsah bez nutnosti distribuce prostřednictvím aplikačních obchodů.

Nevýhodou PWA je rozdílná podpora některých funkcí mezi platformami a prohlížeči [28, 30]. Pro účely této práce však nejde o zásadní omezení, protože hlavním cílem aplikace je zpřístupnění, procvičování a průběžná aktualizace testového obsahu, nikoli využití pokročilých nativních funkcí zařízení.

1.7 Shrnutí a vymezení prostoru pro navrhované řešení

Nová právní úprava účinná od 1. ledna 2026 významně mění rámec, v němž uchazeči získávají zbrojní oprávnění. Změny se netýkají pouze samotného označení oprávnění, ale také podoby zkoušky odborné způsobilosti a širšího prostředí celé agendy [1, 2, 3]. Pro přípravu na teoretickou část zkoušky je podstatné zejména to, že stát zveřejňuje oficiální soubor testových otázek a že právní předpisy přesně stanovují strukturu i způsob hodnocení testu [5, 7]. Digitální nástroj proto nemá oficiální obsah nahrazovat, ale spíše jej zpřístupnit tak, aby byl pro uchazeče přehledný, aktuální a prakticky použitelný.

Z provedeného přehledu současných řešení vyplývá, že uchazeči již dnes mají k dispozici několik webových i mobilních nástrojů pro procvičování testových otázek. Problém tedy nespočívá v tom, že by podobné aplikace vůbec neexistovaly. Jejich nabídka je však poměrně roztříštěná a jednotlivá řešení se liší rozsahem funkcí, platformou, způsobem práce s aktualizacemi i mírou, v jaké je patrná vazba na primární právní zdroje [8, 9, 11, 13].

Prostor pro navrhované řešení proto nevzniká z absence digitálních pomůcek, ale spíše z možnosti spojit jejich klíčové funkce do jednoho uceleného systému. Takový systém by měl vycházet z oficiálního souboru testových otázek a respektovat parametry zkoušky platné od 1. ledna 2026. Zároveň by měl nabídnout instalovatelnou a částečně offline použitelnou webovou formu, přehlednou správu obsahu a podporu učení, která nebude založena pouze na opakovaném vyplňování testů.

V tomto směru je progresivní webová aplikace vhodné technické řešení, protože umožňuje propojit dostupnost webu s některými vlastnostmi běžně spojovanými s nativními aplikacemi, zejména instalovatelností, podporou offline režimu a jednotným vývojem pro více typů zařízení [28, 30]. Cílem této práce proto není vytvořit pouze další databázi otázek, ale navrhnout a implementovat webovou aplikaci, která bude sloužit k systematické přípravě na teoretickou část zkoušky odborné způsobilosti. Přínos práce spočívá především v propojení obsahové správnosti, technické udržitelnosti a uživatelské podpory při procvičování problémových oblastí.

2 Analýza požadavků

Požadavky na aplikaci vycházejí ze zadání práce, rešerše existujících řešení a potřeb cílové skupiny uživatelů. Cílovou skupinu tvoří především uchazeči o získání zbrojního oprávnění, kteří se připravují na teoretickou část zkoušky. Druhou skupinou jsou administrátoři zajišťující správu databáze testových otázek a odpovědí.

2.1 Funkční požadavky

Funkční požadavky popisují činnosti, které musí aplikace podporovat. Jednotlivé požadavky jsou označeny zkratkou FP.

FP1: Procvičování testových otázek. Aplikace musí umožnit uživateli procvičovat oficiální testové otázky. U každé otázky musí zobrazit její zadání, dostupné odpovědi a po zodpovězení také správné řešení.

FP2: Okamžité vyhodnocení odpovědi. Aplikace musí po zodpovězení otázky okamžitě vyhodnotit, zda byla odpověď správná nebo chybná.

FP3: Procházení databáze otázek. Aplikace musí umožnit uživateli procházet databázi testových otázek bez nutnosti spuštění testu nebo simulace zkoušky.

FP4: Filtrování otázek podle tematických oblastí. Aplikace musí umožnit filtrování otázek podle tematických oblastí. Uživatel si tak bude moci zvolit konkrétní část učiva, kterou chce procvičovat.

FP5: Cvičný test. Aplikace musí umožnit spuštění cvičného testu sloužícího k průběžnému ověření znalostí uživatele. Tento režim nemusí dodržovat všechny podmínky reálné zkoušky.

FP6: Simulace reálné zkoušky. Aplikace musí umožnit spuštění simulace reálné zkoušky. Simulace musí odpovídat podmínkám teoretické části zkoušky, zejména výběrem otázek, časovým omezením a způsobem vyhodnocení.

FP7: Vyhodnocení výsledků testu. Aplikace musí po dokončení cvičného testu nebo simulované zkoušky zobrazit výsledek uživatele. Výsledek musí obsahovat alespoň počet správných a chybných odpovědí.

FP8: Uživatelský účet. Aplikace musí umožnit vytvoření a používání uživatelského účtu, ke kterému budou vázány informace o studijním postupu, procvičených otázkách a výsledcích testů.

FP9: Přihlášení pomocí externích poskytovatelů identity. Aplikace musí umožnit přihlášení prostřednictvím externích poskytovatelů identity, například služeb Google a Microsoft.

FP10: Ukládání studijního postupu. Aplikace musí ukládat studijní postup uživatele a zpřístupnit jej po opětovném přihlášení.

FP11: Synchronizace mezi zařízeními. Aplikace musí umožnit dostupnost uložených dat napříč zařízeními, na kterých je uživatel přihlášen ke svému účtu.

FP12: Sledování studijního postupu. Aplikace musí zobrazovat přehled studijního postupu uživatele. Přehled musí zahrnovat zejména procvičené otázky, výsledky testů a úspěšnost v jednotlivých tematických oblastech.

FP13: Administrační rozhraní. Aplikace musí obsahovat administrační rozhraní určené pro správu databáze testových otázek a odpovědí.

FP14: Správa testových otázek. Administrátor musí mít možnost přidávat, upravovat a aktualizovat testové otázky a jejich odpovědi.

FP15: Omezení přístupu do administrace. Přístup do administračního rozhraní musí být umožněn pouze oprávněným uživatelům.

2.2 Nefunkční požadavky

Nefunkční požadavky určují vlastnosti aplikace, které ovlivňují její použitelnost, bezpečnost, spolehlivost, provoz a další rozvoj. Jednotlivé požadavky jsou označeny zkratkou NP.

NP1: Použitelnost. Uživatelské rozhraní musí být navrženo tak, aby uživatel mohl snadno najít hlavní části aplikace, zejména procvičování otázek, spuštění cvičného testu, simulaci zkoušky a přehled výsledků.

NP2: Jazyk rozhraní. Uživatelské i administrační rozhraní aplikace musí být dostupné v českém jazyce.

NP3: Responzivita. Aplikace musí využívat responzivní webový návrh, který umožní přizpůsobení rozvržení velikosti obrazovky použitého zařízení. Aplikace musí být použitelná na mobilních telefonech, tabletech i stolních počítačích [31].

NP4: Bezpečnost přihlášení. Aplikace musí zajistit bezpečné přihlášení uživatelů a ochranu uložených uživatelských dat.

NP5: Řízení přístupových práv. Aplikace musí rozlišovat oprávnění běžného uživatele a administrátora. Funkce administračního rozhraní musí být dostupné pouze administrátorům.

NP6: Spolehlivost ukládání dat. Aplikace musí spolehlivě ukládat studijní postup uživatele, výsledky testů a historii procvičování.

NP7: Konzistentní vyhodnocování odpovědí. Vyhodnocení odpovědí musí být konzistentní a musí vycházet z aktuální databáze správných odpovědí.

NP8: Výkon. Aplikace musí reagovat bez zbytečných prodlev při běžných uživatelských akcích, zejména při zobrazení otázky, vyhodnocení odpovědi, filtrování otázek a načtení výsledků.

NP9: Udržovatelnost. Zdrojový kód aplikace musí být strukturován tak, aby umožňoval další úpravy, opravy a rozšíření aplikace.

NP10: Aktualizovatelnost obsahu. Databáze testových otázek musí být navržena tak, aby umožňovala snadnou aktualizaci otázek a odpovědí.

2.3 Omezení a předpoklady řešení

Tato část shrnuje požadavky a omezení, která nesouvisí přímo s funkcionalitou aplikace, ale určují podobu výsledného řešení. Jednotlivé položky jsou označeny zkratkou O.

O1: Nasazení do testovacího provozu. Výsledná aplikace bude nasazena do testovacího provozu, aby bylo možné ověřit její funkčnost v reálném prostředí.

O2: Dokumentace. Součástí řešení bude dokumentace popisující použití aplikace, správu obsahu a základní technické řešení.

O3: Licence. Zdrojové kódy aplikace budou odevzdány pod licencí MIT.

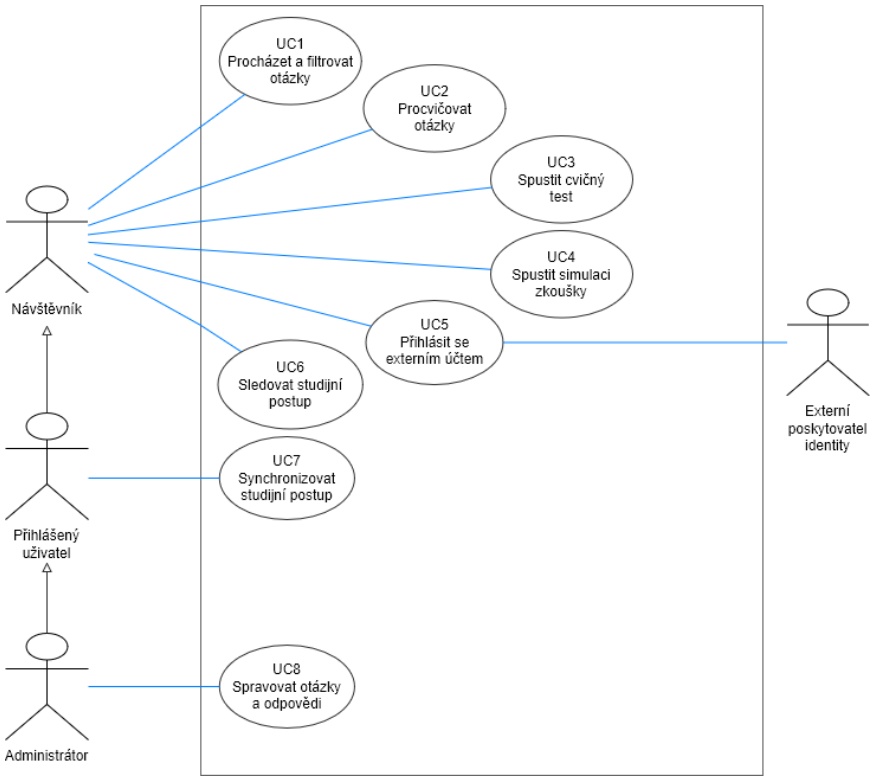
2.4 Případy užití

Tato část popisuje hlavní aktéry systému a případy užití navrhované aplikace. Případy užití vycházejí z funkčních požadavků uvedených v předchozí části a zachycují nejdůležitější způsoby interakce uživatelů se systémem. Přehled vztahů mezi aktéry a případy užití je znázorněn na obrázku 2.1.

2.4.1 Aktéři

Návštěvník

Návštěvník je uživatel, který používá aplikaci bez přihlášení. Může procházet databázi otázek, filtrovat otázky podle tematických oblastí, procvičovat je, spustit cvičný test nebo simulaci zkoušky a sledovat svůj studijní postup v rámci daného zařízení. Jeho postup je však uchováván pouze lokálně a není dostupný na jiných zařízeních. Nemá tedy k dispozici funkce navázané na uživatelský účet, zejména synchronizaci dat mezi zařízeními.



Obrázek 2.1: Diagram případů užití aplikace

Registrovaný uživatel

Registrovaný uživatel je přihlášený uživatel s vlastním uživatelským účtem. Oproti návštěvníkovi může svůj studijní postup ukládat k uživatelskému účtu a synchronizovat jej mezi více zařízeními. Díky tomu má k dispozici stejnou historii procvičování, výsledky testů a přehled úspěšnosti bez ohledu na zařízení, ze kterého aplikaci používá.

Administrátor

Administrátor je oprávněný uživatel, který spravuje obsah aplikace. Má přístup do administračního rozhraní, ve kterém může přidávat, upravovat a aktualizovat testové otázky, odpovědi a související metadata. Přístup do administrační části systému je omezen pouze na uživatele s příslušným oprávněním.

Externí poskytovatel identity

Externí poskytovatel identity je služba, která zajišťuje ověření identity uživatele při přihlášení. V navrhované aplikaci se uvažuje zejména o přihlášení pomocí služeb Google a Microsoft. Tento aktér nevystupuje jako běžný uživatel aplikace, ale jako externí systém zapojený do procesu autentizace.

2.4.2 Přehled případů užití

V aplikaci byly identifikovány hlavní případy užití označené zkratkou UC. Jejich přehled, včetně hlavních aktérů a vazby na funkční požadavky, uvádí tabulka 2.1.

Tabulka 2.1: Přehled hlavních případů užití

ID	Případ užití	Aktér	FP
UC1	Procházet a filtrovat otázky	N, U	3, 4
UC2	Procvičovat otázky	N, U	1, 2, 10
UC3	Spustit cvičný test	N, U	5, 7
UC4	Spustit simulaci zkoušky	N, U	6, 7
UC5	Přihlásit se externím účtem	N	8, 9
UC6	Sledovat studijní postup	N, U	10, 12
UC7	Synchronizovat studijní postup	U	11
UC8	Spravovat otázky a odpovědi	A	13, 14, 15

Pozn.: N – návštěvník, U – registrovaný uživatel, A – administrátor. Čísla ve sloupci FP odkazují na funkční požadavky.

2.4.3 Popis hlavních případů užití

UC1: Procházet a filtrovat otázky

Uživatel otevře část aplikace určenou pro procházení otázek. Aplikace mu zobrazí dostupné testové otázky a jejich odpovědi. Uživatel může otázky procházet postupně nebo omezit jejich výběr podle tematické oblasti, například na právní předpisy, nauku o zbraních a střelivu, bezpečné nakládání se zbraněmi nebo zdravotnické minimum. Výstupem je zobrazení otázek odpovídajících zvolenému filtru.

UC2: Procvičovat otázky

Uživatel spustí režim procvičování. Aplikace zobrazí otázku a dostupné odpovědi. Po výběru odpovědi aplikace okamžitě vyhodnotí, zda byla odpověď správná, a zobrazí správné řešení. Uživatel tak získá okamžitou zpětnou vazbu a může pokračovat na další otázku. Výsledek odpovědi se uloží do studijního postupu. U nepřihlášeného uživatele je tento postup uchováván pouze lokálně, zatímco u přihlášeného uživatele je navázán na jeho uživatelský účet.

UC3: Spustit cvičný test

Uživatel spustí cvičný test, který slouží k průběžnému ověření znalostí. Aplikace sestaví sadu testových otázek, uživatel na ně odpoví a po dokončení testu aplikace zobrazí výsledek. Cvičný test umožňuje uživateli ověřit aktuální úroveň znalostí. Výsledek se uloží do studijního postupu. U nepřihlášeného uživatele je dostupný pouze na daném zařízení, u přihlášeného uživatele je uložen k jeho účtu.

UC4: Spustit simulaci zkoušky

Uživatel spustí simulaci reálné zkoušky. Aplikace sestaví test podle pravidel teoretické části zkoušky, nastaví časový limit a umožní uživateli odpovídat na jednotlivé otázky. Po dokončení testu nebo po vypršení časového limitu aplikace odpovědi vyhodnotí a zobrazí počet správných a chybných odpovědí včetně informace o úspěšnosti. Výsledek simulace se uloží do studijního postupu. U nepřihlášeného uživatele je tento postup dostupný pouze lokálně, zatímco u přihlášeného uživatele je uložen k jeho uživatelskému účtu.

UC5: Přihlásit se externím účtem

Uživatel zvolí přihlášení pomocí externího účtu. Aplikace jej přesměruje k poskytovateli identity, například Google nebo Microsoft, kde proběhne ověření uživatele. Po úspěšném ověření je uživatel přihlášen do aplikace, případně je mu vytvořen nový uživatelský účet. Přihlášení umožňuje využívat funkce navázané na účet, zejména ukládání studijního postupu k uživatelskému účtu a jeho zpřístupnění po opětovném přihlášení.

UC6: Sledovat studijní postup

Uživatel otevře přehled svého studijního postupu. Aplikace zobrazí data o procvičování, výsledcích testů a úspěšnosti v jednotlivých tematických oblastech. Uživatel tak může sledovat svůj pokrok a identifikovat oblasti, ve kterých má slabší výsledky.

Pokud uživatel používá aplikaci bez přihlášení, je jeho postup dostupný pouze lokálně na daném zařízení. Pokud je uživatel přihlášen, aplikace zobrazuje studijní postup navázaný na jeho uživatelský účet.

UC7: Synchronizovat studijní postup

Registrovaný uživatel používá aplikaci po přihlášení ke svému uživatelskému účtu, například na jiném zařízení nebo po opětovném přihlášení. Aplikace načte studijní postup uložený k jeho účtu a zpřístupní mu historii procvičování, výsledky testů a přehled úspěšnosti napříč zařízeními. Uživatel tak může pokračovat ve studiu bez ohledu na zařízení, ze kterého aplikaci používá.

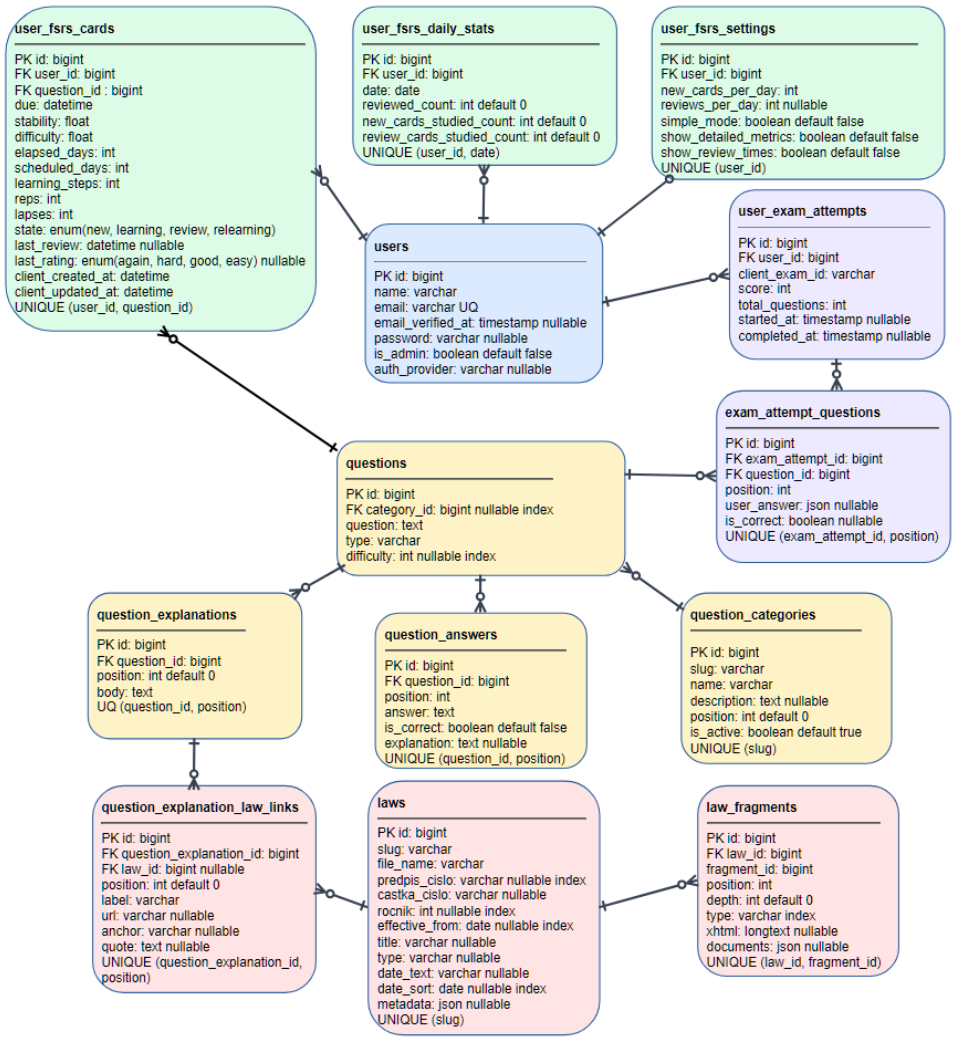
Synchronizace studijního postupu je dostupná pouze přihlášenému uživateli. Nepřihlášenému uživateli se postup uchovává pouze lokálně na daném zařízení a mezi zařízeními se nepřenáší.

UC8: Spravovat otázky a odpovědi

Administrátor otevře administrační rozhraní a aplikace ověří jeho oprávnění. Po úspěšném ověření může administrátor spravovat databázi testových otázek a odpovědí. Správa zahrnuje přidávání nových otázek, úpravu existujících otázek, nastavení správných odpovědí a zařazení otázek do tematických oblastí. Pokud uživatel nemá administrátorské oprávnění, aplikace mu přístup do této části systému neumožní.

3 Návrh aplikace

3.1 Databázový model



Obrázek 3.1: Diagram vztahů mezi entitami

Obrázek 3.1 zobrazuje databázový model aplikace. Model je rozdělen na obsahová data, uživatelská data, data pro plánování opakování

pomocí algoritmu FSRS, testové pokusy a legislativní podklady. Hlavními entitami jsou uživatelé, otázky, odpovědi, vysvětlení, právní předpisy a záznamy o průběhu procvičování a testování.

3.1.1 Obsahová data

Obsahová část modelu je postavena kolem tabulky `questions`. Každá otázka může být zařazena do tematické kategorie pomocí tabulky `question_categories`. Na otázky dále navazují možné odpovědi uložené v tabulce `question_answers`. Pořadí odpovědí je určeno atributem `position` a v rámci jedné otázky je unikátní.

Otázky mohou mít také jedno nebo více vysvětlení uložených v tabulce `question_explanations`. Vysvětlení jsou řazena pomocí atributu `position`, přičemž kombinace otázky a pozice je unikátní. Díky tomu je možné k jedné otázce uložit více samostatných vysvětlujících částí v přesně určeném pořadí.

3.1.2 Legislativní podklady

Legislativní data jsou od samotných otázek oddělena a ukládána do samostatných tabulek. Jejich struktura vychází z exportu právních předpisů ze služby `e-sbirka.gov.cz` [32]. Tabulka `laws` obsahuje základní informace o právních předpisech, jako je název, typ, rok, číslo předpisu, datum účinnosti a další metadata odvozená z importovaných dat, která umožňují určit použitou verzi předpisu a při opětovné synchronizaci rozpoznat, zda je nutné data aktualizovat. Konkrétní části jednotlivých právních předpisů jsou poté ukládány do tabulky `law_fragments`.

Propojení vysvětlení s právními předpisy zajišťuje tabulka `question_explanation_law_links`. Jeden výklad tak může odkazovat na jeden nebo více právních předpisů, jejich části nebo externí zdroje. Pořadí odkazů je určeno atributem `position`, který je v rámci jednoho vysvětlení unikátní.

3.1.3 Uživatelská data a FSRS

Uživatelská část modelu vychází z tabulky `users`. Ta obsahuje základní údaje o uživateli, například jméno, e-mail, ověření e-mailu, heslo, informaci o administrátorském oprávnění a poskytovateli autentizace.

Tabulky související s plánováním opakování jsou navrženy podle datového modelu používaného knihovnou FSRS. Stav procvičování je uložen v tabulce `user_fsrs_cards`. Každý záznam reprezentuje vztah mezi jedním uživatelem a jednou otázkou. Kombinace `user_id` a `question_id` je unikátní, takže pro jednu otázku existuje u daného uživatele nejvýše jeden stav procvičování. Tabulka ukládá hodnoty potřebné pro algoritmus FSRS, například stabilitu, obtížnost, počet opakování, plánovaný termín dalšího opakování a aktuální stav karty. Uživatelské nastavení procvičování je uloženo v tabulce `user_fsrs_settings`. Každý uživatel má nejvýše jeden záznam nastavení, což je zajištěno unikátním omezením nad sloupcem `user_id`. Tabulka obsahuje například počet nových karet za den, počet opakování za den a volby pro zobrazení detailních metrik nebo času opakování.

Tabulka `user_fsrs_daily_stats` uchovává denní statistiky procvičování pro každého uživatele. Pro každý den a uživatele existuje nejvýše jeden záznam, což je zajištěno unikátním omezením nad dvojicí `user_id` a `date`. Tabulka obsahuje počty zobrazených otázek, nově studovaných karet a opakovaných karet.

3.1.4 Testové pokusy

Cvičné testy a simulace zkoušky jsou reprezentovány tabulkami `user_exam_attempts` a `exam_attempt_questions`. Jednotlivé otázky v rámci pokusu jsou uloženy v tabulce `exam_attempt_questions`. Každý záznam odkazuje na konkrétní pokus a konkrétní otázku, ukládá pořadí otázky v testu, odpověď uživatele a informaci o správnosti odpovědi.

Pořadí otázky je v rámci jednoho pokusu unikátní díky omezení nad dvojicí `exam_attempt_id` a `position`.

3.1.5 Technické tabulky

Součástí návrhu databázového modelu jsou kromě výše uvedených tabulek také technické tabulky používané frameworkem Laravel, například pro správu relací, přístupových tokenů, obnovu hesla, fronty úloh nebo cache. Tyto tabulky nejsou v diagramu uvedeny, protože nesouvisí přímo s obsahem a funkcemi aplikace, ale jsou potřeba pro běžný provoz.

3.2 Návrh uživatelského rozhraní

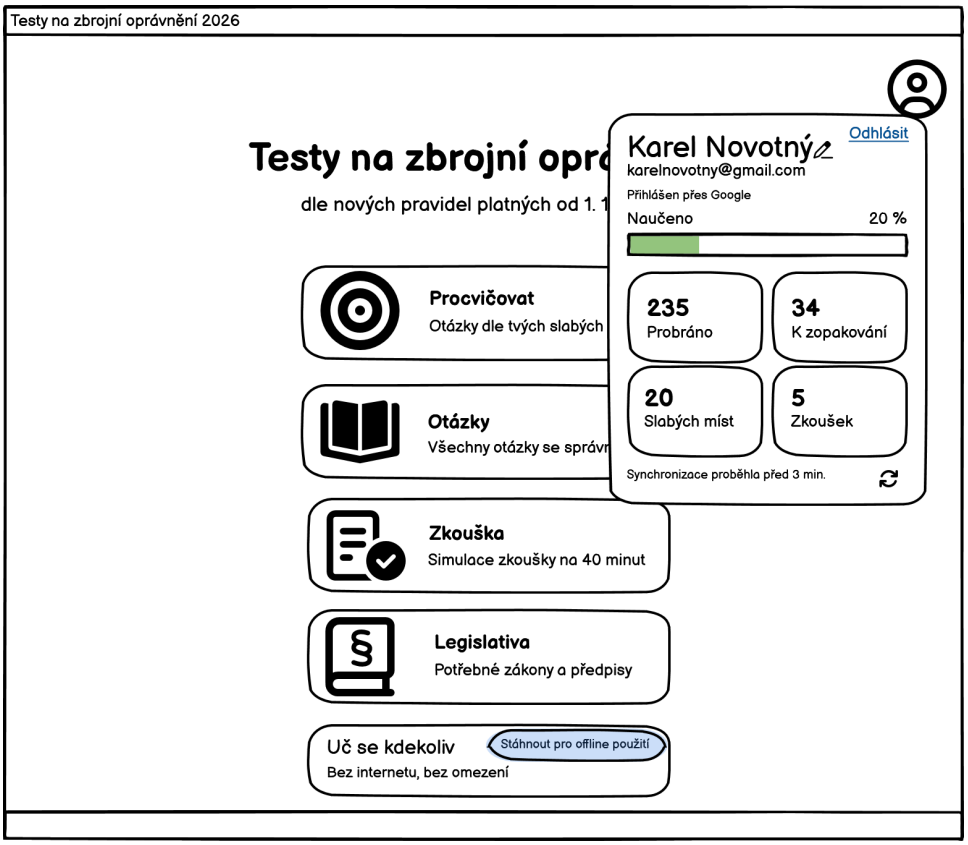
Návrh uživatelského rozhraní byl vytvořen ve formě wireframů v nástroji Balsamiq Wireframes [33]. Wireframy zachycují rozmístění hlavních prvků, tok uživatele a dostupnost důležitých funkcí aplikace. Důraz je kladen na strukturu obrazovek, hierarchii informací a základní navigaci mezi jednotlivými částmi aplikace.

3.2.1 Úvodní stránka

Úvodní stránka slouží jako hlavní rozcestník aplikace. Uživatel se z ní může dostat do jednotlivých režimů, zejména do procvičování, procházení otázek, simulace zkoušky a legislativní části. Návrh využívá výrazné karty, které obsahují název funkce, krátký popis a ilustrační ikonu. Díky tomu lze rychle rozlišit jednotlivé režimy aplikace.

Součástí úvodní stránky je také uživatelský panel se základními informacemi o přihlášeném uživateli. Panel zobrazuje postup učení, počet probraných otázek, počet otázek k zopakování, slabá místa a počet absolvovaných zkoušek. Uživatel tak ihned po otevření aplikace vidí svůj aktuální stav a může pokračovat v doporučeném procvičování nebo zvolit jiný režim.

Úvodní stránka obsahuje také prvek pro práci s offline režimem. Uživatel zde může stáhnout obsah pro použití mimo aktivní připojení k internetu, což podporuje učení v různých situacích. Zároveň zde dostane možnost v případě podpory aplikaci nainstalovat, což zlepší přístup a uživatelský zážitek.



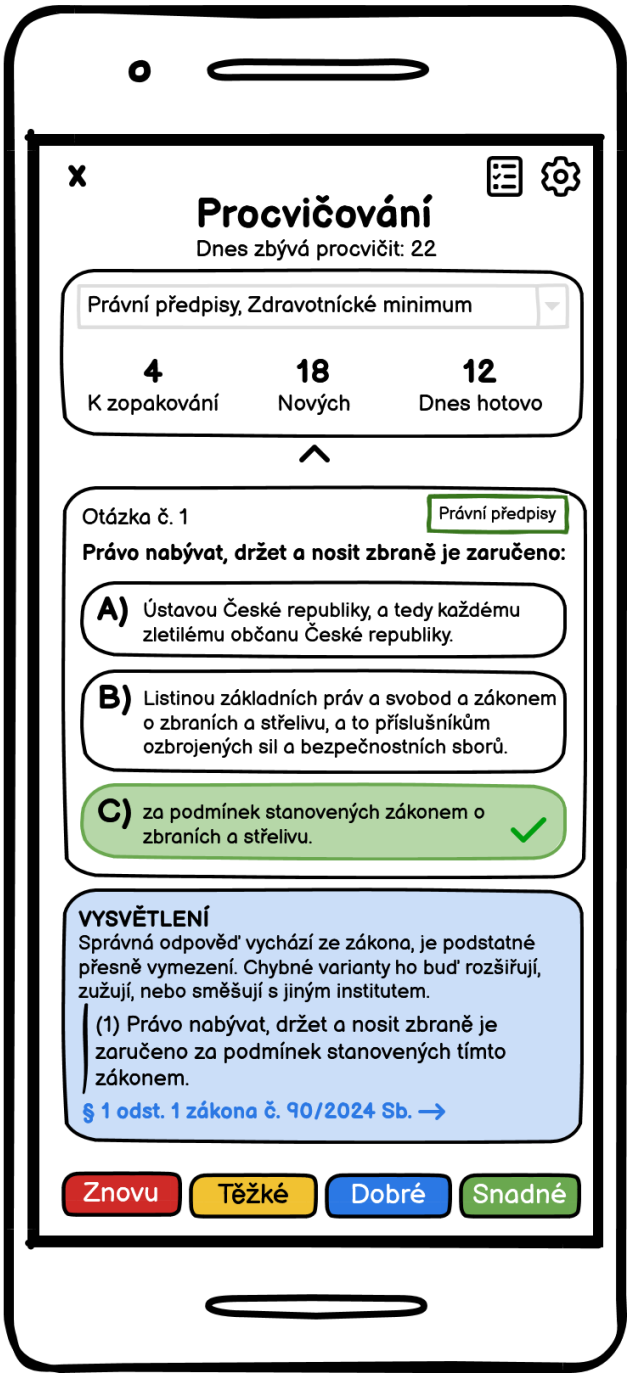
Obrázek 3.2: Wireframe úvodní stránky aplikace

3.2.2 Obrazovka procvičování

Obrazovka procvičování slouží jako hlavní prostor pro práci s otázkami. Uživatel zde vybírá okruh otázek, sleduje průběžný stav procvičování a odpovídá na jednotlivé otázky.

Po zodpovězení se zobrazí správná odpověď spolu s vysvětlením a odkazem na příslušný právní předpis nebo jeho část. Tím aplikace podporuje nejen zapamatování odpovědi, ale i pochopení jejího zdůvodnění.

Součástí obrazovky jsou také hodnoticí tlačítka *Znovu*, *Těžké*, *Dobré* a *Snadné*. Uživatel jimi označí obtížnost otázky a tato volba slouží jako vstup pro plánování dalšího opakování pomocí algoritmu FSRS.



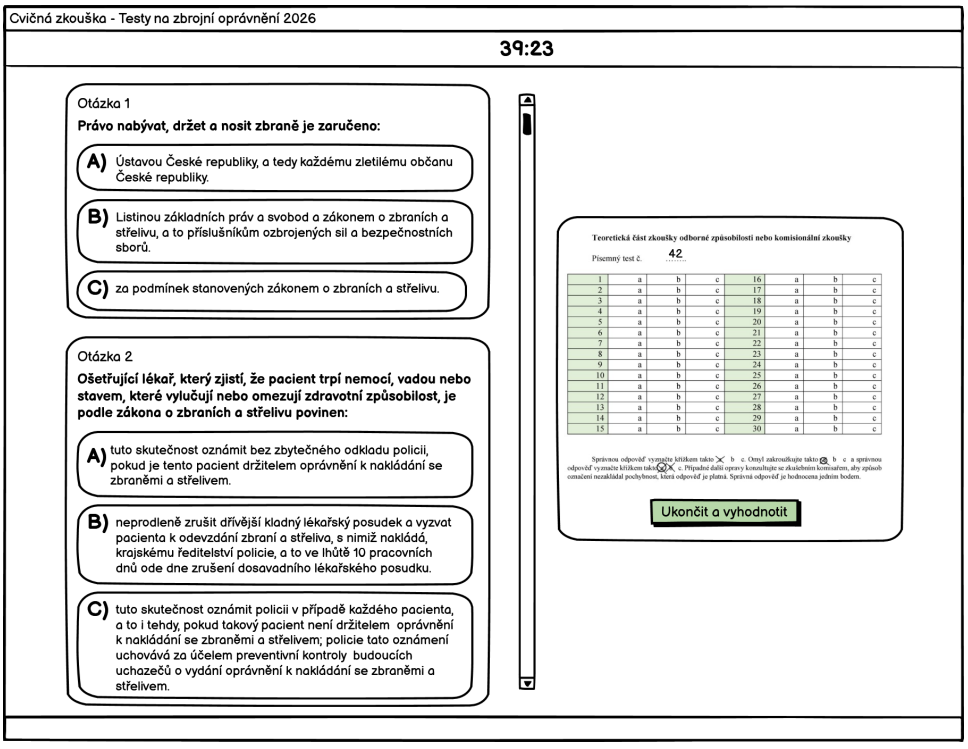
Obrázek 3.3: Wireframe obrazovky procvičování

3.2.3 Obrazovka simulace zkoušky

Obrazovka simulace zkoušky slouží k procvičení teoretické části za podmínek podobných skutečné zkoušce. Uživatel odpovídá na otázky v časovém limitu a postupně prochází jednotlivé položky testu.

Součástí desktopového návrhu je také záznamový formulář vycházející ze vzoru uvedeného v příloze č. 1 nařízení vlády č. 238/2025 Sb. [2]. Uživatel si jednak může vyzkoušet odpovídání způsobem podobným reálné zkoušce, jednak se seznámí s tím, jak se ve formuláři značí správná odpověď a jakým způsobem se opravuje chybně označená možnost. Tím aplikace podporuje nejen znalost samotných otázek, ale i praktickou přípravu na administrativní podobu zkoušky.

Na menších obrazovkách je rozhraní zjednodušeno, protože současné zobrazení otázek i formuláře by bylo prostorově nepraktické. Mobilní verze proto počítá s přímým výběrem odpovědí u jednotlivých otázek.



Obrázek 3.4: Wireframe obrazovky simulace zkoušky

4 Implementace aplikace

4.1 Použité technologie a nástroje

Tato kapitola popisuje technologie a nástroje použité při návrhu, implementaci, vývoji a nasazení výsledné webové aplikace. Volba vycházela především z požadavku na vytvoření responzivní webové aplikace a možností dalšího rozšiřování. Důraz byl kladen také na udržovatelnost zdrojového kódu a dostupnost dokumentace použitých technologií.

4.1.1 Frontend

React

Uživatelské rozhraní je implementováno pomocí knihovny React. React umožňuje vytvářet rozhraní z menších znovupoužitelných komponent. [34]. Zde je to vhodné u komponent pro režimy učení, společné záhlaví studijních obrazovek, zobrazení vysvětlení odpovědí, administrační formuláře nebo panel uživatelského účtu a synchronizace. Díky tomu lze dynamické části aplikace snadněji skládat, upravovat a opakovaně používat na více místech bez zbytečné duplicity kódu.

TypeScript

Pro klientskou část je použit jazyk TypeScript. JavaScriptu přidává statické typování, díky kterému je možné odhalit část chyb už během vývoje [35].

Typy jsou využity u datových struktur, jako jsou otázky, odpovědi a výsledky. Díky tomu je jasnější, jaká data se mezi komponentami předávají, a snižuje se riziko chyb při jejich zpracování.

FSRS

Pro režim chytrého opakování byla využita knihovna `ts-fsrs`, která poskytuje TypeScriptovou implementaci plánovače FSRS pro systémy rozloženého opakování [36]. V klientské části slouží k výpočtu nového stavu procvičované otázky a k určení termínu jejího dalšího opakování.

Výpočet probíhá přímo na zařízení uživatele. Po zodpovězení otázky se lokálně vyhodnotí zvolené hodnocení odpovědi a podle něj se aktualizují data potřebná pro další plánování. Serverová část tím není zatěžována jednotlivými výpočty plánovače a slouží především k dlouhodobému uložení a synchronizaci studijního postupu.

Toto řešení snižuje závislost procvičování na síťové komunikaci. Uživatel dostává další krok bez čekání na odpověď serveru a stejný mechanismus lze použít také v offline režimu. Pokud jsou potřebná data dostupná lokálně, může procvičování i plánování dalších opakování pokračovat bez připojení k internetu a uložený postup se synchronizuje až po jeho obnovení.

Tailwind CSS

Pro stylování uživatelského rozhraní je použit framework Tailwind CSS [37]. Ten využívá přístup utility-first, při němž se vzhled prvků skládá pomocí předpřipravených tříd přímo v komponentách. V této práci je využit zejména pro rychlou tvorbu konzistentního a responzivního rozhraní použitelného na mobilních i desktopových zařízeních.

4.1.2 Backend a aplikační logika

Laravel

Backend je implementován pomocí frameworku Laravel. Ten zajišťuje routování, práci s databází, autentizaci, autorizaci, validaci vstupů a obsluhu administrační části [38].

Laravel byl zvolen kvůli přehledné struktuře projektu a dostupným nástrojům pro běžné části webového vývoje. V aplikaci je využit pro správu otázek, odpovědí, uživatelů a výsledků. Framework zároveň podporuje architekturu Model-View-Controller, která pomáhá oddělit jednotlivé části systému [39].

Propojení backendu s frontendovou částí

Frontendová část je do aplikace začleněna prostřednictvím balíčku `laravel-vite-plugin`, který umožňuje načítání JavaScriptových a CSS souborů přímo v rámci serverově generovaných částí aplikace [40].

Při produkčním sestavení jsou frontendové soubory optimalizovány a následně používány aplikací v produkčním prostředí. Díky tomu může být frontendová část součástí stejného nasazení jako backend, bez nutnosti provozovat ji jako samostatnou aplikaci.

4.1.3 Databáze

MariaDB

Pro ukládání dat je v aplikaci použita relační databáze MariaDB [41]. Uchovává zejména uživatelské účty, testové otázky, odpovědi, tematické kategorie, výsledky testů a údaje o studijním postupu.

Implementace aplikace však není pevně závislá pouze na MariaDB. Laravel poskytuje databázovou abstrakční vrstvu a Eloquent ORM, díky nimž je většina práce s daty oddělena od konkrétního databázového systému. Při zachování relačního návrhu by proto bylo možné použít i jinou podporovanou databázi, například MySQL, PostgreSQL, SQLite nebo SQL Server.

MariaDB byla zvolena kvůli jednoduché integraci s Laravelem, dobré kompatibilitě s ekosystémem MySQL, nízkým provozním nárokům a dostupnosti jako open-source řešení. Pro potřeby této aplikace poskytuje dostatečný výkon, stabilitu a snadnou správu.

4.1.4 Autentizace a autorizace

Autentizace a autorizace jsou v aplikaci řešeny pomocí nástrojů frameworku Laravel. Pro přístup k chráněným částem API je použit Laravel Sanctum, který umožňuje vytvářet přístupové tokeny pro jednotlivá zařízení uživatele [42]. Po registraci nebo přihlášení aplikace vrátí uživatelská data společně s tokenem, který je následně používán při komunikaci s chráněnými endpointy.

Běžná registrace a přihlášení jsou implementovány v kontroleru `AuthController`. Uživatel se může registrovat pomocí jména, e-mailu a hesla. Při přihlášení se ověřuje e-mail a heslo a při úspěšném ověření je vytvořen nový přístupový token. Chráněné části API, například synchronizace postupu, úprava profilu, změna hesla nebo export dat, jsou dostupné pouze přes middleware `auth:sanctum`.

Přihlášení pomocí externích poskytovatelů

Vedle běžného přihlášení aplikace podporuje také přihlášení pomocí externích poskytovatelů Google a Microsoft. Přihlášení je založeno na protokolech OAuth 2.0, případně OpenID Connect, kdy uživatel ověří svou identitu u externí služby a aplikace následně získá údaje potřebné pro vytvoření nebo nalezení lokálního uživatelského účtu. Tato část je implementována pomocí balíčku Laravel Socialite a rozšíření Socialite Providers pro Microsoft [43, 44].

Přihlášení probíhá přes endpointy ve tvaru `/api/auth/{provider}/redirect` a `/api/auth/{provider}/callback`, kde podporovanými hodnotami `provider` jsou `google` a `microsoft`. Po úspěšném ověření u externího poskytovatele aplikace vyhledá existujícího uživatele podle poskytovatele a e-mailové adresy. Pokud uživatel neexistuje, vytvoří se nový účet. Následně je vytvořen přístupový token a uživatel je přesměrován zpět do frontendové části aplikace.

Technicky by bylo snadno možné doplnit také přihlášení pomocí Apple účtu, protože by šlo o obdobný autentizační tok. Tato možnost však nebyla do aplikace zahrnuta, protože pro její praktické nasazení je nutné využít placený Apple Developer Program. Naproti tomu přihlášení přes Google a Microsoft bylo možné zprovoznit bezplatně.

Autorizace

Přístup k chráněným částem aplikace je řízen pomocí middleware. Endpointy pro synchronizaci, správu profilu a práci s uživatelskými daty vyžadují platný Sanctum token. Administrační API je navíc chráněno vlastním middleware `admin`, které kontroluje hodnotu `is_admin` u přihlášeného uživatele.

Administrátorský příznak může být uložen přímo u uživatele v databázi nebo nastaven konfigurační hodnotou `ADMIN_EMAILS`. Tím je možné oddělit běžné uživatele od správců aplikace a omezit přístup k funkcím pro správu uživatelů, otázek a odpovědí.

4.2 Vývojové prostředí

4.2.1 Laravel Sail a Docker

Pro lokální vývoj je použit Laravel Sail. Laravel Sail je nástroj pro práci s výchozím Dockerovým vývojovým prostředím Laravelu [45]. Umožňuje spustit aplikaci a její závislosti v kontejnerech, aniž by bylo nutné všechny služby instalovat přímo do operačního systému vývojáře.

Dockerové prostředí je vhodné zejména proto, že sjednocuje konfiguraci mezi různými zařízeními. Aplikace může používat samostatný kontejner pro PHP, databázi MariaDB a případně další služby. Díky tomu lze vývojové prostředí snadněji reprodukovat a předejít rozdílům způsobeným různými verzemi lokálně nainstalovaných nástrojů.

Sail tedy usnadňuje spuštění projektu, práci s databází, spouštění migrací a testování aplikace. Zároveň umožňuje přiblížit lokální vývojové prostředí produkčnímu nasazení, kde jsou jednotlivé služby také provozovány odděleně.

4.2.2 Vite

Pro vývoj frontendové části aplikace je použit nástroj Vite [46]. Ten zpracovává zdrojové soubory napsané v TypeScriptu, React komponenty a styly definované pomocí Tailwind CSS.

Během vývoje Vite poskytuje vývojový server založený na nativních ES modulech, díky čemuž není nutné při každé změně znovu sestavovat celou aplikaci. Změny ve zdrojových souborech jsou do prohlížeče promítány pomocí mechanismu Hot Module Replacement (HMR), který aktualizuje pouze změněné moduly bez úplného obnovení stránky. V případě Reactu Vite využívá integraci React Fast Refresh, která umožňuje rychlou aktualizaci komponent při zachování stavu aplikace, což výrazně urychluje úpravy klientské části aplikace.

4.3 Testování během vývoje

Součástí vývoje aplikace bylo průběžné ověřování správnosti backendové části pomocí automatizovaných testů a kontrola sestavení frontendové části. Cílem testování během vývoje bylo zachytit chyby v aplikační logice ještě před nasazením nové verze a současně ověřit, že změny v jedné části systému neporuší již existující funkcionalitu.

Backendová část je testována pomocí frameworku PHPUnit, který je součástí Laravelu. Testy jsou spouštěny v prostředí Laravel Sail, tedy ve stejném kontejnerizovaném prostředí, které se používá i při lokálním vývoji aplikace. Díky tomu odpovídají podmínky pro spouštění testů vývojovému prostředí projektu.

Automatizované testy jsou zaměřeny především na API aplikace. Ověřují správnost stavových kódů, strukturu JSON odpovědí, zpracování validačních chyb, autorizaci uživatelů a ukládání dat do databáze. Testovány jsou zejména části související s autentizací, správou uživatelského profilu, otázkami, legislativou, synchronizací uživatelského postupu a administračními funkcemi.

Součástí testů jsou také chybové scénáře. Kontroluje se například chování aplikace při požadavku nepřihlášeného uživatele, při odeslání neplatných vstupních dat nebo při požadavku na neexistující záznam. Tyto testy pomáhají ověřit nejen správnou funkci běžných scénářů, ale také bezpečné a předvídatelné chování aplikace v nestandardních situacích.

Frontendová část je během vývoje kontrolována pomocí TypeScriptu a build procesu nástroje Vite. Při sestavení se ověřuje typová správnost zdrojového kódu a schopnost vytvořit produkční build aplikace. Tím lze zachytit část chyb ještě před samotným nasazením.

Testy jsou využívány průběžně během vývoje a zároveň jsou součástí procesu nasazení nové verze aplikace pomocí GitLab CI/CD pipeline.

4.4 Struktura projektu

4.4.1 Základní adresářová struktura

```
/
|-- app/
|   |-- Http/
|   |   |-- Controllers/
|   |   |   `-- Api/
|   |   `-- Middleware/
|   `-- Models/
|-- database/
|   |-- migrations/
|   `-- seeders/
|-- public/
|   |-- manifest.json
|   |-- sw.js
|   `-- images/
|-- resources/
|   `-- js/
|       |-- components/
|       |-- hooks/
|       |-- pages/
|       |-- services/
|       `-- types/
`-- routes/
    |-- api.php
    `-- web.php
```

Obrázek 4.1: Zjednodušená struktura projektu

4.4.2 Backendová část

Backendová část je umístěna především v adresáři `app`. Adresář `app/Http/Controllers/Api` obsahuje API controllery pro práci s otázkami, legislativou, offline daty, uživatelským účtem, synchronizací postupu a administrací. Přístup k vybraným částem API je chráněn autentizací pomocí Laravel Sanctum. Administrační rozhraní je navíc omezeno middlewarem, který kontroluje oprávnění administrátora.

Modely aplikace jsou umístěny v adresáři `app/Models`. Reprezentují hlavní databázové entity, například uživatele, otázky, legislativní dokumenty, vysvětlení odpovědí a výsledky zkoušek.

4.4.3 Databázová vrstva

Databázová vrstva je definována v adresáři `database/migrations`. Migrace vytvářejí tabulky pro uživatele, otázky, legislativní dokumenty, vysvětlení odpovědí, odpovědi uživatele, denní statistiky procvičování, nastavení uživatele a výsledky zkoušek.

Adresář `database/seeders` slouží k naplnění databáze počátečními daty, zejména testovými otázkami, vysvětleními odpovědí a legislativními podklady.

4.4.4 Frontendová část

Frontendová část se nachází v adresáři `resources/js`. Složka `pages` obsahuje hlavní obrazovky aplikace, například procházení otázek, procvičování, simulaci zkoušky, přehled pokroku, legislativu, profil a administraci.

Složka `components` obsahuje znovupoužitelné prvky uživatelského rozhraní. V adresáři `hooks` jsou vlastní React hooky, například pro práci s autentizací, synchronizací nebo offline otázkami. Adresář `types` obsahuje TypeScript typy používané napříč aplikací.

4.4.5 Klientské služby

Adresář `resources/js/services` obsahuje klientské služby, které oddělují aplikační logiku od samotných komponent. Tyto služby zajišťují komunikaci s API, práci s IndexedDB, offline ukládání otázek, simulaci zkoušky, FSRS procvičování, synchronizaci uživatelského postupu a práci s legislativou.

Toto rozdělení zjednodušuje údržbu frontendové části, protože komponenty řeší hlavně vykreslení uživatelského rozhraní, zatímco práce s daty je soustředěna ve službách.

4.4.6 PWA a veřejné soubory

Adresář `public` obsahuje veřejně dostupné soubory. Soubor `manifest.json` definuje metadata progresivní webové aplikace, například název, ikony a režim spuštění. Soubor `sw.js` implementuje Service Worker, který cachuje základní soubory aplikace, buildované assety, obrázky a vybrané API odpovědi.

Ve složce `public/images` jsou uloženy obrázky používané u testových otázek.

4.4.7 Routování

Soubor `routes/api.php` definuje REST API používané frontendovou částí aplikace. Obsahuje veřejné endpointy pro otázky, legislativu a offline data a chráněné endpointy pro účet, synchronizaci a administraci.

Soubor `routes/web.php` zajišťuje webové routy potřebné pro načtení jednostránkové aplikace.

4.5 Procvičování otázek

Procvičování otázek je založeno na algoritmu FSRS, který slouží k plánování opakování podle úspěšnosti uživatele. Cílem tohoto přístupu je zobrazovat otázky v takových intervalech, aby si uživatel učivo efektivněji zapamatoval a zároveň nemusel zbytečně často opakovat otázky, které již dobře ovládá.

Při výběru otázek aplikace nezobrazuje pouze náhodné položky. Upřednostňuje nové otázky a otázky, u kterých již nastal termín dalšího opakování. Otázky, se kterými má uživatel dlouhodobě potíže, jsou zařazovány častěji, zatímco otázky, které uživatel opakovaně zvládá, se zobrazují s delšími časovými odstupy. Procvičování tak lépe odpovídá individuálním potřebám konkrétního uživatele.

Po zodpovězení otázky uživatel ohodnotí svou odpověď jednou z nabízených možností, například `Znovu`, `Těžké`, `Dobré` nebo `Snadné`. Na základě tohoto hodnocení algoritmus FSRS vypočítá nový termín dalšího opakování a aktualizuje stav dané otázky.

U každé procvičované otázky je evidován stav opakování, obtížnost, stabilita, datum dalšího opakování, počet opakování a počet chybných odpovědí. Tato data jsou ukládána lokálně do IndexedDB,

aby bylo možné s aplikací pracovat i bez okamžité komunikace s backendem. Pokud je uživatel přihlášen, změna lokálního stavu zároveň vyvolá událost, která spustí synchronizaci s backendem.

Procvičování dále podporuje filtrování otázek podle kategorií. U každé otázky lze zobrazit správnou odpověď a doplňující vysvětlení.

4.6 Simulace zkoušky

Test obsahuje 30 otázek s časovým limitem 40 minut [2] a odpovídá skladbě teoretické části zkoušky:

- 17 otázek z právních předpisů,
- 5 otázek z bezpečného nakládání,
- 5 otázek z nauky o zbraních a střelivu,
- 3 otázky ze zdravotnického minima [2].

Otázky jsou vybírány náhodně z příslušných rozsahů. Stav rozpracované zkoušky se ukládá do IndexedDB. Ukládají se vybrané otázky, odpovědi uživatele, čas začátku a zbývající čas. Díky tomu lze zkoušku obnovit i po zavření nebo obnovení stránky.

Po dokončení zkoušky aplikace spočítá počet správných odpovědí, zobrazí výsledek a uloží jej do historie. Historie obsahuje otázky, odpovědi, skóre, datum dokončení a délku trvání zkoušky. Tato data se ukládají lokálně a po přihlášení jsou synchronizována se serverem.

4.7 Uživatelský účet

Uživatelské účty jsou implementovány pomocí Laravel Sanctum. API umožňuje registraci, přihlášení, odhlášení, změnu profilu, změnu hesla, export dat a odstranění účtu. Přihlášení vrací přístupový token, který frontend ukládá do `localStorage` a posílá jej v hlavičce autorizovaných požadavků.

Aplikace podporuje také přihlášení přes Google a Microsoft. Tato část je řešena pomocí Laravel Socialite a příslušných providerů.

Účet není nutný pro běžné používání aplikace. Nepřihlášený uživatel může procházet otázky, procvičovat i spustit simulaci zkoušky. Účet slouží hlavně k záloze dat a k přenosu pokroku mezi zařízeními.

4.8 Ukládání postupu a synchronizace

Pokrok se primárně ukládá lokálně do IndexedDB. Aplikace ukládá hlavně stav FSRS algoritmu, denní statistiky, nastavení procvičování a výsledky zkoušek. Lokální ukládání je důležité pro offline režim a zároveň zrychluje práci aplikace, protože po stažení jsou studijní data dostupná přímo v prohlížeči i bez internetu.

Po přihlášení se lokální data synchronizují se serverem přes endpointy `/api/sync`. Synchronizační payload obsahuje:

- pokrok procvičování,
- denní statistiky procvičování,
- nastavení procvičování,
- výsledky dokončených zkoušek.

Při synchronizaci pokroku se porovnává čas poslední klientské úpravy. Starší data se nepřepisují novějšími. Synchronizace se spouští automaticky po změně pokroku a také po obnovení online připojení. Pokud uživatel není online, aplikace pokračuje pouze s lokálními daty a synchronizace proběhne později.

4.9 Administrační rozhraní

Administrační část je dostupná na trase `/admin`. Přístup je chráněn autentizací a middlewarem, který kontroluje, zda má uživatel příznak administrátora, který lze nastavit v `.env` souboru nebo přímo v databázi. Tím je zajištěno, že administrace není přístupná běžným uživatelům.

Administrace obsahuje správu uživatelů a správu otázek. U uživatelů se zobrazují základní údaje a statistiky studia, například počet studovaných otázek, počet otázek k opakování, počet slabých otázek,

počet výsledků zkoušek a průměrná úspěšnost. Administrátor může uživateli změnit heslo nebo účet odstranit.

U otázek může administrátor upravovat odpovědi, správnou odpověď, vysvětlení a odkazy na související legislativu. Úprava otázky probíhá v databázové transakci, protože se v jedné operaci mohou měnit data v několika souvisejících tabulkách.

4.10 Podpora progresivní webové aplikace

Aplikace obsahuje prvky progresivní webové aplikace (PWA), které zlepšují použitelnost na mobilních zařízeních a umožňují použití bez připojení k internetu. Základním prvkem je soubor `manifest.json`, který definuje název aplikace, ikony, úvodní adresu, barvu motivu a způsob zobrazení po instalaci [47]. Díky tomu může být aplikace v podporovaných prohlížečích nainstalována a spuštěna podobně jako běžná mobilní nebo desktopová aplikace [48].

Po načtení aplikace je v prohlížeči registrován service worker `sw.js`. Service worker funguje jako mezivrstva mezi aplikací, sítí a lokální cache prohlížeče [49]. V této aplikaci ukládá do cache základní soubory aplikace, manifest, ikony a sestavené soubory z adresáře `/build`. Pro statické soubory je použita strategie *cache first*, zatímco pro vybraná API data, například otázky a legislativu, strategie *network first*. Aplikace se tak nejprve pokusí získat aktuální data ze serveru, ale při nedostupnosti sítě může použít dříve uloženou odpověď.

Offline použitelnost není řešena pouze na úrovni statických souborů, ale také na úrovni aplikačních dat. Uživatel si může stáhnout balíček otázek pro offline použití. Otázky a vysvětlení jsou ukládány do IndexedDB, která je vhodná pro ukládání většího množství strukturovaných dat v prohlížeči [50]. Metadata o staženém balíčku jsou uložena v `localStorage`. Obrázky k otázkám a legislativní dokumenty jsou ukládány pomocí Cache API [51].

Aplikace zároveň eviduje verzi staženého offline balíčku. Pokud je na serveru dostupná novější verze dat, může na tuto skutečnost uživatele upozornit a nabídnout aktualizaci. Součástí uživatelského rozhraní je také tlačítko pro instalaci aplikace jako PWA, které využívá manifest aplikace a zobrazuje se v případě, že prohlížeč instalaci podporuje.

4.11 Příprava a import obsahových dat

Obsahová část aplikace je tvořena třemi hlavními typy dat: testovými otázkami, legislativními podklady a vysvětleními správných odpovědí. Každý z těchto typů dat měl odlišný zdroj i způsob zpracování, proto jsou popsány samostatně.

4.11.1 Import testových otázek

Pro naplnění databáze byl využit oficiální dokument Ministerstva vnitra ve formátu PDF, který obsahuje testové otázky s vizuálně vyznačenými správnými odpověďmi [7]. Přestože dokument obsahuje textovou vrstvu a nejde o čistý sken, jeho vnitřní struktura nebyla vhodná pro přímý import do databáze.

Nejprve byly vyzkoušeny běžné způsoby převodu PDF do textu, HTML a dokumentu Word. Výstupy však nebyly použitelné pro spolehlivé rozdělení dat na otázky, odpovědi, správné odpovědi a případné obrázky. Při převodu docházelo ke ztrátě struktury, narušení vazeb mezi částmi otázky a výskytu neplatných nebo nestandardních znaků.

Hlavní komplikací bylo, že správné odpovědi nejsou v dokumentu uloženy jako samostatná data. Jsou určeny pouze vizuálně, zejména kurzívou a šedým podkladem. Tento podklad navíc není reprezentován jako textové zvýraznění nebo anotace, ale jako grafický objekt umístěný pod textem. Běžné konverzní nástroje proto tuto informaci nedokázaly zachovat. Dalším problémem bylo nestandardní mapování některých znaků a ligatur v textové vrstvě, kvůli kterému bylo nutné extrahovaný text dále normalizovat.

Z těchto důvodů byl pro prvotní import vytvořen pomocný skript. Při jeho přípravě byl použit nástroj ChatGPT, zejména pro návrh postupu extrakce, práci s knihovnami pro zpracování PDF a úpravy dílčích částí implementace. Skript je přiložen ke zdrojovému kódu projektu jako pomocný nástroj pro reprodukovatelnost importu.

Skript nejprve pomocí regulárních výrazů vyhledává začátky otázek podle jejich číselného označení a jednotlivé odpovědi ve formátu `A)`, `B)` a `C)`. Podporuje také malé varianty písmen odpovědí, které převádí na jednotný formát. Pokud otázka nebo odpověď pokračuje na

dalších řádcích, skript tyto řádky připojuje k aktuálně zpracovávané položce.

Správná odpověď je určována heuristicky podle vizuálních vlastností textu, zejména podle kurzívy a přítomnosti šedého podkladu. U otázek obsahujících obrázek skript sleduje obrazové objekty nacházející se u aktuální otázky a po jejím dokončení z PDF vytvoří odpovídající výřez ve formátu PNG. Vazba mezi otázkou a obrázkem tedy není převzata z metadat dokumentu, ale odvozena z polohy prvků na stránce.

Součástí zpracování je také čištění textu. Skript normalizuje mezery, odstraňuje problematické řídicí znaky, nahrazuje vybrané ligatury a opravuje některé chyby vzniklé při mapování znaků z PDF fontů. Po dokončení extrakce provádí základní validaci výsledku. Kontroluje se zejména souvislá řada otázek, přítomnost všech tří odpovědí a jednoznačné určení správné odpovědi.

Výstupem procesu byla strukturovaná data obsahující text otázky, jednotlivé odpovědi, označení správné odpovědi, tematickou oblast a případnou obrazovou přílohu. Tato data byla následně použita pro naplnění databáze aplikace.

Použitý skript není možné považovat za obecný převodník testových otázek z PDF. Je navázán na konkrétní podobu zdrojového dokumentu a na jeho vizuální formátování. Pokud Ministerstvo vnitra v budoucnu zveřejní dokument s jinou sazbou, pořadím objektů nebo způsobem zvýraznění správných odpovědí, skript nemusí být bez úprav znovu použitelný. U menšího rozsahu změn proto může být efektivnější provést konverzi ručně, případně využít jazykový model jako asistenta při převodu a kontrole strukturovaných dat.

4.11.2 Import legislativních podkladů

Kromě testových otázek byly do aplikace importovány také relevantní právní předpisy. Zdrojová data byla získána z portálu e-Sbírka.cz [32]. Pro import byly využity strojově čitelné soubory ve formátu JSON, což umožnilo spolehlivé zpracování a zachování struktury dokumentu.

Import zákonů je realizován pomocí Laravel seederu `LawSeeder`. V adresáři `database/seeders/data/laws/` jsou vstupní JSON soubory, přičemž každý odpovídá jednomu právnímu předpisu. Název sou-

boru bez přípony se používá jako jednoznačný identifikátor zákona v aplikaci.

Každý vstupní soubor obsahuje část `metadata` a pole `fragmenty`. Metadata obsahují základní informace o předpisu, například číslo předpisu, ročník, název a datum účinnosti. Tyto údaje se ukládají do tabulky `laws`. Samotný text předpisu je uložen jako posloupnost fragmentů, které se ukládají do tabulky `law_fragments`.

Fragment nepředstavuje vždy celý paragraf nebo odstavec, ale obecnou část struktury právního předpisu. Může jít například o nadpis, část, paragraf, odstavec, písmeno nebo bod. Každý fragment obsahuje vlastní identifikátor, typ, hloubku zanoření a obsah ve formátu XHTML. Text zákona se proto v aplikaci neskládá z jednoho souvislého textového pole, ale postupným vykreslením fragmentů v pořadí, v jakém jsou uvedeny ve vstupním JSON souboru.

Tento způsob uložení umožňuje zachovat původní strukturu právního předpisu a zároveň odkazovat na konkrétní části zákona. Frontend při zobrazení zákona prochází jednotlivé fragmenty, vykresluje jejich XHTML obsah a podle typu fragmentu vytváří kotvy pro přímé odkazy. Díky tomu je možné odkazovat například na konkrétní paragraf nebo odstavec:

```
/legislativa/Sb_2024_90_2026-01-01_IZ#par_1-odst_1
```

Pokud není možné vytvořit sémantickou kotvu podle struktury zákona, použije se záložní kotva založená na identifikátoru fragmentu. Tento mechanismus je důležitý zejména pro vysvětlení testových otázek, která mohou odkazovat přímo na relevantní ustanovení právního předpisu.

Práce s fragmenty má však i určitá omezení. Význam konkrétní části zákona nemusí být obsažen pouze v jednom fragmentu, ale může vyplývat z kombinace nadpisů, paragrafu, odstavce a dalších navazujících částí. Při vytváření odkazů je proto nutné respektovat pořadí fragmentů a jejich hierarchii. Aplikace fragmenty neinterpretuje jako vlastní právní model, ale používá je především k věrnému zobrazení textu předpisu a k vytvoření stabilních odkazů na jeho části.

4.11.3 Import vysvětlení otázek

Součástí řešení jsou kromě testových otázek a správných odpovědí také stručná vysvětlení. Jejich cílem je, aby uživatel při procvičování pouze nememoroval správnou možnost, ale pochopil, proč je daná odpověď správná a z jakého právního nebo věcného základu vychází.

Pro vytvoření vysvětlení byly využity jazykové modely ChatGPT a Gemini. Jako vstup jim byly poskytnuty strojově čitelné testové otázky a relevantní právní předpisy importované z portálu e-Sbírka. Výstupem mělo být krátké vysvětlení každé otázky, včetně odkazu na konkrétní ustanovení zákona tam, kde takové ustanovení existuje.

Tvorba vysvětlení probíhala iterativně. Výstupy byly opakovaně kontrolovány a upravovány tak, aby odpovídaly znění otázky, správné odpovědi a příslušné části právního předpisu. U otázek, které se nevážou přímo ke konkrétnímu zákonnému ustanovení, například u zdravotnického minima nebo praktických zásad první pomoci, bylo doplněno věcné vysvětlení správné odpovědi bez odkazu na zákon.

Seeder `QuestionExplanationSeeder` zajišťuje import vysvětlení do databáze. Při importu přiřazuje vysvětlení ke konkrétním otázkám a legislativní odkazy propojuje s již importovanými právními předpisy. Systém tak může u otázky zobrazit komentář ke správné odpovědi i odkaz na související část zákona. Po jeho otevření je uživatel přesměrován přímo na relevantní ustanovení v legislativní části aplikace.

Tato část řešení má zároveň prostor pro další zpřesňování. Přestože byla vysvětlení při přípravě kontrolována, u některých otázek může být vhodné doplnit přesnější formulaci, lepší citaci nebo srozumitelnější věcné odůvodnění. Do budoucna proto lze vysvětlení dále upravovat na základě uživatelské zpětné vazby a postupně zvyšovat jejich odbornou přesnost i praktickou užitečnost při studiu.

5 Ověření, nasazení a zhodnocení

5.1 Ověření funkčnosti

Po dokončení implementace byla aplikace ověřena kombinací automatizovaných testů, kontroly produkčního sestavení frontendové části a ručního ověření hlavních uživatelských scénářů.

Při posledním spuštění testovací sady prošlo všech 67 testů s celkovým počtem 288 asercí. Naměřené řádkové pokrytí sledovaných PHP tříd backendové části dosáhlo 100 %.

```
Tests: 67 passed (288 assertions)
Total coverage: 100.0 %
```

Ručně byly ověřeny hlavní uživatelské scénáře aplikace: načítání otázek, procvičování, generování simulace zkoušky, uložení rozpracované zkoušky, fungování offline režimu, synchronizace dat po přihlášení a přístup do administrační části.

Součástí ověření byla také kontrola produkčního sestavení frontendové části. Tím bylo potvrzeno, že klientskou část aplikace lze úspěšně sestavit a nasadit společně s backendem.

5.2 Měření pomocí Lighthouse

K ověření kvality nasazené frontendové části aplikace byl použit nástroj Lighthouse v prohlížeči Google Chrome. Měření bylo provedeno nad produkčním buildem aplikace a zahrnovalo výkon, přístupnost, dodržení doporučených postupů, SEO a podporu PWA.

Nástroj byl využit také při ladění aplikace. Během testování upozornil na nedostatečný kontrast textu vůči pozadí jednoho z tlačítek v uživatelském rozhraní. Tento nedostatek byl odstraněn úpravou barev daného prvku, čímž se zlepšila jeho čitelnost i přístupnost.

Lighthouse dále upozornil na možnost zmenšení velikosti statických obrázků. Původně použité soubory ve formátu PNG byly proto pomocí terminálového nástroje `ImageMagick` převedeny do formátu WebP, který při zachování dostatečné vizuální kvality výrazně snížil velikost přenášených dat. Celková velikost obrázkových podkladů se z původních přibližně 55 MB snížila na 2,9 MB. Díky této optimalizaci

uživatel při první návštěvě webu a stažení všech podkladů pro offline použití, včetně instalace aplikace jako PWA, přenese celkem přibližně 3,6 MB dat.

5.3 Nasazení

Aplikace je nasazována jako jedna Laravel aplikace obsahující backendovou i frontendovou část. Nasazení je řízeno pomocí GitLab CI a automatizováno nástrojem Laravel Envoy. Konfigurace CI pipeline je uložena v souboru `.gitlab-ci.yml` a jednotlivé kroky nasazení jsou definovány v souboru `Envoy.blade.php`.

Deploy proces pracuje s konkrétní verzí zdrojového kódu z CI pipeline. Na serveru připraví novou verzi aplikace, nainstaluje PHP závislosti pomocí Composeru, nainstaluje JavaScriptové závislosti pomocí npm a vytvoří produkční sestavení frontendových souborů. Následně se připojí sdílená konfigurace aplikace a perzistentní úložiště souborů.

Po přípravě nové verze je aplikace přepnuta na aktuální release. V závěru nasazení jsou spuštěny databázové migrace a propojení veřejně dostupného úložiště souborů. Frontend a backend jsou nasazovány společně, takže verze klientské části odpovídá aktuální verzi serverové části aplikace.

Produkční verze aplikace představuje finální nasazené řešení, které je veřejně dostupné na doméně `testynazbrojniopravneni.cz`.

5.3.1 Hosting

Aplikace je provozována na virtuálním serveru v prostředí Oracle Cloud. Pro nasazení byla zvolena instance `VM.Standard.E2.1.Micro` v režimu Always Free [52] s operačním systémem Ubuntu [53].

Server zajišťuje běh webového serveru, PHP runtime, databáze MariaDB a sestavených frontendových souborů vytvořených pomocí nástroje Vite. Backendová část zpracovává příchozí požadavky, komunikuje s databází a poskytuje data klientské části aplikace.

Nasazení na virtuálním privátním serveru umožňuje plnou kontrolu nad konfigurací prostředí, správou databáze, instalovanými závislostmi i procesem publikace nových verzí aplikace.

5.3.2 Zpřístupnění aplikace pomocí Cloudflare Tunnel

Aplikace je veřejně dostupná prostřednictvím služby Cloudflare Tunnel. Na serveru běží služba `cloudflared`, která vytváří odchozí šifrované spojení do infrastruktury Cloudflare. Veřejný provoz je následně směrován přes Cloudflare na lokální službu běžící na serveru [54].

Tento přístup eliminuje nutnost otevírat příchozí porty na serveru a omezuje jeho přímé vystavení internetu. Cloudflare zároveň zajišťuje správu HTTPS certifikátů pro veřejnou doménu aplikace a umožňuje konfigurovat směrování požadavků na konkrétní lokální službu.

5.4 Uživatelské testování

Kromě automatizovaných testů a ručního ověření hlavních scénářů bylo provedeno také uživatelské testování výsledné aplikace. Cílem testování bylo ověřit použitelnost aplikace na různých typech zařízení, získat zpětnou vazbu od reálných uživatelů a na základě zjištěných poznatků upravit vybrané části uživatelského rozhraní.

Testování mělo kvalitativní charakter a bylo provedeno se sedmi uživateli. Účastníci používali různá zařízení a operační systémy, konkrétně zařízení s Windows, Linux, Android a iOS. Při testování byly ověřovány zejména scénáře procvičování otázek, práce s režimem opakování, simulace zkoušky a celková srozumitelnost ovládání aplikace.

První testující používal aplikaci na počítači se systémem Linux. Při procvičování otázek upozornil na to, že opakované klikání myší je při delším používání zdlouhavé. Zároveň byl zvyklý z aplikace Anki na ovládání pomocí klávesnice. Na základě této připomínky byla do nastavení procvičování přidána možnost zapnout ovládání klávesnicí. Pro volbu a hodnocení odpovědí byly zvoleny klávesy `A`, `S`, `D`, `F` a pro pokračování mezerník. Tato volba vychází z přirozeného položení levé ruky při psaní všemi deseti, kdy se prsty nacházejí právě v jejich okolí. Klávesa `F` je navíc na většině klávesnic opatřena hmatovou značkou, což usnadňuje orientaci bez nutnosti sledovat klávesnici.

Druhý testující používal zařízení Apple iPhone a upozornil na to, že ho při procvičování zdržuje nutnost po každé odpovědi vybírat subjektivní hodnocení obtížnosti pro algoritmus FSRS. Na základě této zpětné vazby byla do nastavení přidána volba jednoduchého režimu. Po jeho zapnutí se uživateli po zodpovězení otázky nezobrazují

tlačítka pro hodnocení obtížnosti, ale pouze možnost pokračovat na další otázku. Tento režim snižuje množství rozhodování při procvičování a činí práci s aplikací jednodušší, i když za cenu méně přesného plánování opakování oproti plnému využití algoritmu FSRS.

Třetí testující používal zařízení se systémem Android a při testování nenarazil na zásadní problém. Ovládání aplikace, zobrazení otázek i průběh procvičování pro něj byly srozumitelné.

Čtvrtý a pátý testující používali mobilní zařízení se systémy Android a iOS. Oba byli s používáním aplikace spokojeni, ale zajímalo je, jakým způsobem aplikace doporučuje otázky k dalšímu opakování. Na základě této připomínky byla do nastavení přidána možnost zobrazit časy opakování. Pokud je tato volba aktivní, uživatel při hodnocení otázky vidí, za jak dlouho se mu otázka podle zvolené obtížnosti znovu zobrazí. Tato úprava zvyšuje transparentnost algoritmu a umožňuje uživateli lépe pochopit dopad jednotlivých voleb na další plán procvičování.

Šestý testující používal počítač se systémem Windows 10. S funkčností aplikace byl spokojen, ale uvedl, že by uvítal možnost volby jiného vzhledu aplikace. Tato připomínka nebyla v rámci práce implementována, ale byla zařazena mezi možné směry dalšího rozvoje, například ve formě podpory více vizuálních motivů.

Sedmý testující používal počítač se systémem Windows 11. Pozitivně hodnotil zejména režim simulace zkoušky, který podle něj dobře odpovídal očekávané podobě reálného testu. Tato zpětná vazba potvrdila význam snahy o dodržení oficiálních parametrů zkoušky, zejména počtu otázek, časového limitu a způsobu vyhodnocení.

Na základě uživatelského testování byly do aplikace doplněny tři praktické úpravy nastavení procvičování: ovládání pomocí klávesnice, jednoduchý režim procvičování a zobrazování časů opakování. Testování zároveň potvrdilo, že aplikace je použitelná na různých typech zařízení a že hlavní scénáře, zejména procvičování otázek a simulace zkoušky, jsou pro uživatele srozumitelné. Získané připomínky se týkaly především komfortu ovládání, míry zobrazených detailů a možností přizpůsobení uživatelského rozhraní.

5.5 Dokumentace a licence

Základní technická dokumentace projektu se nachází v souboru `README.md`. Tento soubor popisuje potřebné závislosti, nastavení vývojového prostředí a postup pro spuštění aplikace ze zdrojových kódů.

Zdrojové kódy aplikace jsou odevzdány pod licencí MIT. Licenční podmínky jsou uvedeny v souboru `LICENSE.md`, který je součástí projektu.

Závěr

Cílem bakalářské práce bylo navrhnout a implementovat webovou aplikaci pro přípravu na zkoušku odborné způsobilosti k získání zbrojního oprávnění. V teoretické části byla provedena rešerše současných řešení, která se zaměřila především na dostupné funkce, práci s testovými otázkami, způsob vyhodnocování a uživatelské rozhraní. Na základě této rešerše a analýzy potřeb cílové skupiny byly stanoveny požadavky na novou aplikaci.

V praktické části byla vytvořena aplikace umožňující procvičování oficiálních testových otázek. Uživatel může otázky procházet a filtrovat podle tematických oblastí, využít cvičný režim, spustit simulaci reálné zkoušky a získat automatické vyhodnocení výsledků. Významnou součástí aplikace je režim chytrého opakování, který využívá algoritmus FSRS. Ten umožňuje plánovat opakování otázek podle předpokládané míry zapamatování a předchozích odpovědí uživatele.

Aplikace byla navržena jako progresivní webová aplikace s responzivním uživatelským rozhraním. Podporuje používání na různých typech zařízení, vybrané funkce v offline režimu, ukládání postupu uživatele a synchronizaci dat mezi zařízeními. Součástí řešení je také uživatelský účet a administrační rozhraní pro správu uživatelů a testových otázek. Vybrané části systému byly otestovány, zejména backendové API, autentizace, synchronizace dat a práce s uživatelským profilem.

Výsledkem práce je funkční aplikace, která představuje moderní nástroj pro systematickou přípravu na zkoušku. Oproti běžně dostupným řešením klade důraz zejména na přehledné rozhraní, personalizované opakování otázek, možnost offline použití a synchronizaci napříč zařízeními.

Do budoucna by bylo možné aplikaci dále rozšířit například o využití anonymizovaných dat od většího počtu uživatelů. Na jejich základě by bylo možné přesněji určovat reálnou obtížnost jednotlivých otázek a tyto informace využít jako vstupní parametr pro algoritmus FSRS. Aplikace by se tak mohla lépe přizpůsobit i novým uživatelům, kteří zatím nemají vytvořenou vlastní historii odpovědí.

Na základě provedené rešerše, návrhu, implementace, testování a dokumentace lze konstatovat, že výsledné řešení splňuje stanovené cíle bakalářské práce.

Bibliografie

1. *Zákon č. 90/2024 Sb., o zbraních a střelivu* [online]. Parlament České republiky, 2024 [cit. 2026-03-22]. Dostupné z: https://www.e-sbirka.cz/sb/2024/90. účinnost od 1. ledna 2026.
2. *Nařízení vlády č. 238/2025 Sb., o provádění zkoušek podle zákona o zbraních a střelivu* [online]. Vláda České republiky, 2025 [cit. 2026-03-22]. Dostupné z: https://www.e-sbirka.cz/sb/2025/238. účinnost od 1. ledna 2026.
3. *Hlavní změny v právní úpravě na úseku zbraní, střeliva a munice od 1. ledna 2026* [online]. Ministerstvo vnitra České republiky, 2026 [cit. 2026-03-22]. Dostupné z: https://mv.gov.cz/clanek/hlavni-zmeny-v-pravni-uprave-na-useku-zbrani-streliva-a-munice-od-1-ledna-2026.aspx. sekce rozcestníku k nové právní úpravě.
4. *Nejčastější otázky k nové právní úpravě nakládání se zbraněmi, střelivem a municí* [online]. Ministerstvo vnitra České republiky, 2026 [cit. 2026-03-22]. Dostupné z: https://mv.gov.cz/clanek/nejcastejsi-otazky-k-nove-pravni-uprave-nakladani-se-zbranemi-strelivem-a-munici.aspx. sekce rozcestníku k nové právní úpravě.
5. *Zbraně, střelivo, munice a bezpečnostní materiál* [online]. Ministerstvo vnitra České republiky, 2026 [cit. 2026-03-22]. Dostupné z: https://mv.gov.cz/clanek/zbrane-strelivo-bezpecnostni-material.aspx. rozcestník k nové právní úpravě a souvisejícím dokumentům.
6. *Zbrojní oprávnění* [online]. Ministerstvo vnitra České republiky, 2026 [cit. 2026-03-22]. Dostupné z: https://mv.gov.cz/clanek/zbrojni-opravneni.aspx.
7. *Soubor testových otázek pro teoretickou část zkoušky odborné způsobilosti a komisionální zkoušky* [online]. Ministerstvo vnitra České republiky, 2025 [cit. 2026-03-22]. Dostupné z: https://mv.gov.cz/soubor/mv-soubor-testovych-otazek-pro-teoretickou-cast-zkousky-odborne-zpusobilosti-a-komisionalni-

zkousku.aspx. platné od 1. ledna 2026, aktualizováno 12. 11. 2025.

8. *Trénink testových otázek ke zkoušce odborné způsobilosti* [online]. ZbraněKvalitně.cz, 2026 [cit. 2026-03-22]. Dostupné z: https://zbranekvalitne.cz/zbrojni-prukaz/testove-otazky/trenink. verze 2026/1/1, platná od 1. ledna 2026.
9. *Zkoušky na zbrojní průkaz on-line – včetně otázek od 2026* [online]. eTesty.cz, 2026 [cit. 2026-03-22]. Dostupné z: https://zbrojni-prukaz.etesty.cz/.
10. *Zkoušky na zbrojní průkaz – prohlížení otázek* [online]. eTesty.cz, 2026 [cit. 2026-03-22]. Dostupné z: https://zbrojni-prukaz.etesty.cz/prohlizeni_otazek.php?otazka=2503-v_trestnim_zakoniku_je_o_nutne_obrane_stanoveno_ze. sekce otázek pro zbrojní oprávnění od 2026.
11. *Zbrojní průkaz – testy 2026* [online]. Google Play, 2026 [cit. 2026-03-22]. Dostupné z: https://play.google.com/store/apps/details?hl=cs&id=cz.etesty.zbrojni_prukaz. NEPLUS, s.r.o.
12. *Zbrojní průkaz 2026 Prémium* [online]. Google Play, 2026 [cit. 2026-03-22]. Dostupné z: https://play.google.com/store/apps/details?hl=cs&id=cz.jamesdeer.guntest. James Deer.
13. *Zbrojní průkaz 2026* [online]. Apple App Store, 2026 [cit. 2026-03-22]. Dostupné z: https://apps.apple.com/rs/app/id1435343141. Jakub Jelínek.
14. EBBINGHAUS, Hermann. *Memory: A Contribution to Experimental Psychology*. New York: Teachers College, Columbia University, 1913. Translated by Henry A. Ruger and Clara E. Bussenius. Original work published 1885.
15. CEPEDA, Nicholas J.; PASHLER, Harold; VUL, Edward; WIXTED, John T.; ROHRER, Doug. Distributed Practice in Verbal Recall Tasks: A Review and Quantitative Synthesis. *Psychological Bulletin*. 2006, roč. 132, č. 3, s. 354–380. Dostupné z DOI: 10.1037/0033-2909.132.3.354.

16. DUNLOSKY, John; RAWSON, Katherine A.; MARSH, Elizabeth J.; NATHAN, Mitchell J.; WILLINGHAM, Daniel T. Improving Students' Learning With Effective Learning Techniques: Promising Directions From Cognitive and Educational Psychology. *Psychological Science in the Public Interest*. 2013, roč. 14, č. 1, s. 4–58. Dostupné z DOI: 10.1177/1529100612453266.

17. KARPICKE, Jeffrey D.; ROEDIGER, Henry L. The Critical Importance of Retrieval for Learning. *Science*. 2008, roč. 319, č. 5865, s. 966–968. Dostupné z DOI: 10.1126/science.1152408.

18. KARPICKE, Jeffrey D.; BLUNT, Janell R. Retrieval Practice Produces More Learning than Elaborative Studying with Concept Mapping. *Science*. 2011, roč. 331, č. 6018, s. 772–775. Dostupné z DOI: 10.1126/science.1199327.

19. SETTLES, Burr; MEEDER, Brendan. A Trainable Spaced Repetition Model for Language Learning. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2016, s. 1848–1858. Dostupné z DOI: 10.18653/v1/P16-1174.

20. WOŹNIAK, Piotr A.; GORZELAŃCZYK, Edward J. Optimization of Repetition Spacing in the Practice of Learning. *Acta Neurobiologiae Experimentalis*. 1994, roč. 54, s. 59–62.

21. WOŹNIAK, Piotr A. *SuperMemo 2: Algorithm* [online]. 1998. [cit. 2026-05-08]. Dostupné z: https://super-memory.com/english/ol/sm2.htm.

22. ANKI. *Background* [online]. 2026. [cit. 2026-05-08]. Dostupné z: https://docs.ankiweb.net/background.html.

23. REDDY, Siddharth; LABUTOV, Igor; BANERJEE, Siddhartha; JOACHIMS, Thorsten. Unbounded Human Learning: Optimal Scheduling for Spaced Repetition. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 2016, s. 1815–1824. Dostupné z DOI: 10.1145/2939672.2939850.

24. TABIBIAN, Behzad; UPADHYAY, Utkarsh; DE, Abir; ZAREZADE, Ali; SCHÖLKOPF, Bernhard; GOMEZ-RODRIGUEZ, Manuel. Enhancing Human Learning via Spaced Repetition Optimization. *Proceedings of the National Academy of Sciences*. 2019, roč. 116, č. 10, s. 3988–3993. Dostupné z DOI: 10.1073/pnas.1815156116.

25. OPEN SPACED REPETITION. *The Algorithm* [online]. 2026. [cit. 2026-05-08]. Dostupné z: https://github.com/open-spaced-repetition/awesome-fsrs/wiki/The-Algorithm.

26. APPLE APP STORE. *Zbrojní Průkaz Pro 2026* [online]. 2026. [cit. 2026-05-08]. Dostupné z: https://apps.apple.com/cz/app/zbrojn%C3%AD-pr%C5%AFkaz-pro-2026/id1629078695.

27. TESTY NA ZBROJNÍ PRŮKAZ. *Jak funguje náš kartičkový systém* [online]. 2026. [cit. 2026-03-04]. Dostupné z: https://www.testy-zbrojni-prukaz.cz/informace/jak-to-funguje.

28. *Making PWAs installable* [online]. MDN Web Docs, 2026 [cit. 2026-03-22]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Guides/Making_PWAs_installable.

29. *Service Worker API* [online]. MDN Web Docs, 2026 [cit. 2026-03-22]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API.

30. *Getting started* [online]. web.dev, 2026 [cit. 2026-03-22]. Dostupné z: https://web.dev/learn/pwa/getting-started. část Learn PWA; obsahuje doporučené charakteristiky kvalitní PWA.

31. *Responsive web design basics* [online]. web.dev, 2019 [cit. 2026-04-12]. Dostupné z: https://web.dev/articles/responsive-web-design-basics.

32. E-SBÍRKA. *e-Sbírka - Elektronická Sbírka zákonů a mezinárodních smluv* [online]. 2026. [cit. 2026-05-08]. Dostupné z: https://e-sbirka.gov.cz/.

33. BALSAMIQ. *Balsamiq Wireframes* [online]. 2026. [cit. 2026-05-08]. Dostupné z: https://balsamiq.com/.

34. *React Documentation* [online]. Meta Open Source [cit. 2026-04-20]. Dostupné z: https://react.dev/.

35. *TypeScript Documentation* [online]. Microsoft [cit. 2026-04-23]. Dostupné z: https://www.typescriptlang.org/docs/.

36. OPEN SPACED REPETITION. *ts-fsrs: TypeScript toolkit for building spaced repetition systems with FSRS* [online]. [cit. 2026-04-22]. Dostupné z: https://github.com/open-spaced-repetition/ts-fsrs.

37. *Tailwind CSS Documentation* [online]. Tailwind Labs [cit. 2026-04-20]. Dostupné z: https://tailwindcss.com/.

38. *Laravel Documentation* [online]. Laravel [cit. 2026-04-20]. Dostupné z: https://laravel.com/docs.

39. *Model-View-Controller* [online]. Laravel [cit. 2026-04-21]. Dostupné z: https://laravel.com/learn/getting-started-with-laravel/what-is-mvc.

40. *Asset Bundling with Vite* [online]. Laravel [cit. 2026-04-21]. Dostupné z: https://laravel.com/docs/vite.

41. *MariaDB Documentation* [online]. MariaDB Foundation [cit. 2026-04-23]. Dostupné z: https://mariadb.org/documentation/.

42. *Laravel Sanctum* [online]. Laravel [cit. 2026-05-02]. Dostupné z: https://laravel.com/docs/sanctum.

43. *Laravel Socialite* [online]. Laravel [cit. 2026-05-02]. Dostupné z: https://laravel.com/docs/socialite.

44. *Microsoft Provider for Laravel Socialite* [online]. Socialite Providers [cit. 2026-05-02]. Dostupné z: https://github.com/SocialiteProviders/Microsoft.

45. *Laravel Sail* [online]. Laravel [cit. 2026-04-21]. Dostupné z: https://laravel.com/docs/sail.

46. *Vite Documentation* [online]. Vite [cit. 2026-04-21]. Dostupné z: https://vite.dev/.

47. *Web app manifests* [online]. MDN Web Docs [cit. 2026-05-07]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Manifest.

48. *Make it installable* [online]. web.dev [cit. 2026-05-07]. Dostupné z: https://web.dev/learn/pwa/installation.

49. *Service Worker API* [online]. MDN Web Docs [cit. 2026-05-07]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API.

50. *IndexedDB API* [online]. MDN Web Docs [cit. 2026-05-07]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API.

51. *Cache API* [online]. MDN Web Docs [cit. 2026-05-07]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/API/Cache.

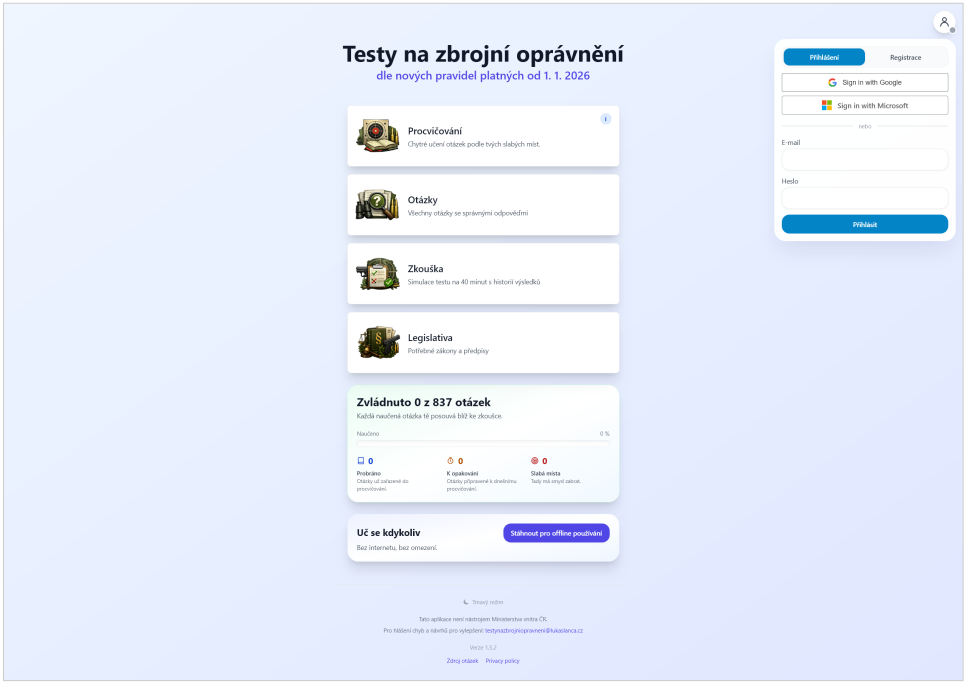
52. *Oracle Cloud Free Tier* [online]. Oracle [cit. 2026-04-21]. Dostupné z: https://www.oracle.com/cloud/free/.

53. *Ubuntu server* [online]. Ubuntu [cit. 2026-04-23]. Dostupné z: https://ubuntu.com/download/server.

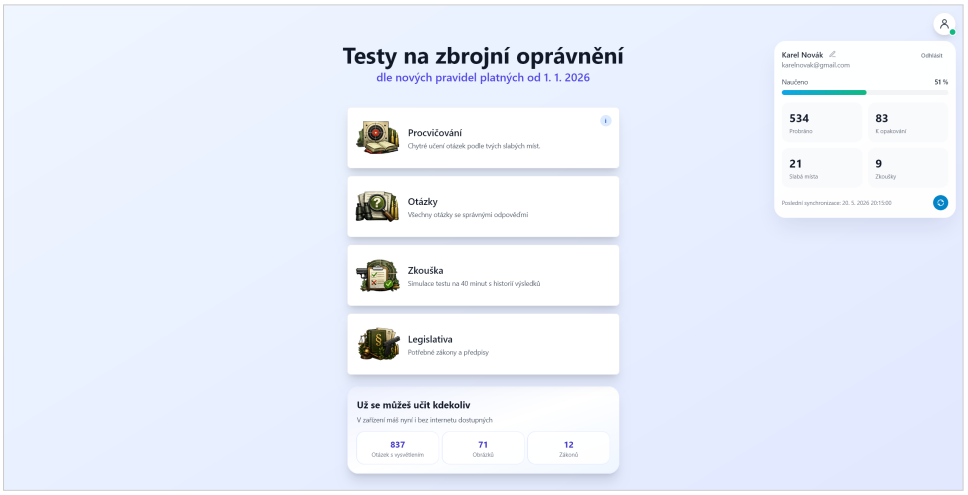
54. *Cloudflare Tunnel* [online]. Cloudflare [cit. 2026-04-23]. Dostupné z: https://developers.cloudflare.com/tunnel/.

A Přílohy

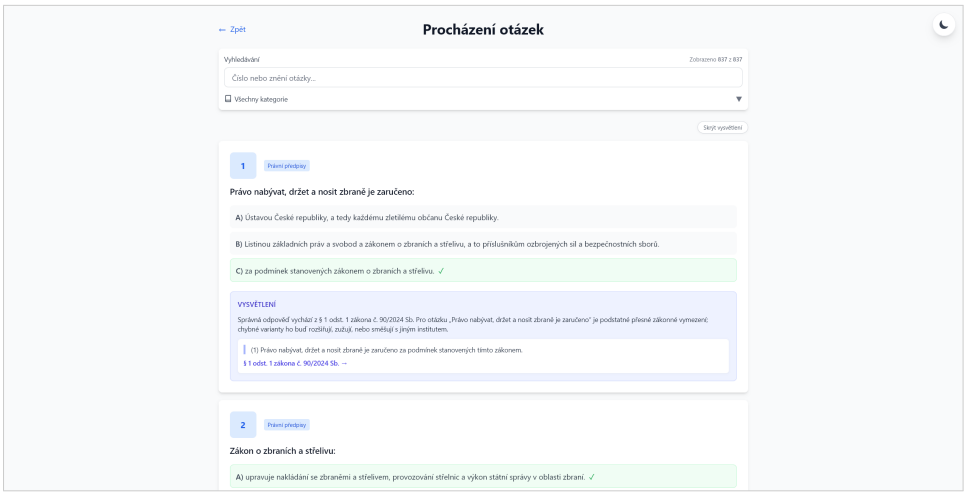
Součástí této práce je ZIP archiv obsahující zdrojový kód aplikace. Kromě toho jsou níže uvedeny také vybrané ukázky základních obrazovek aplikace, které slouží k rychlé vizuální představě o jejím uživatelském rozhraní.



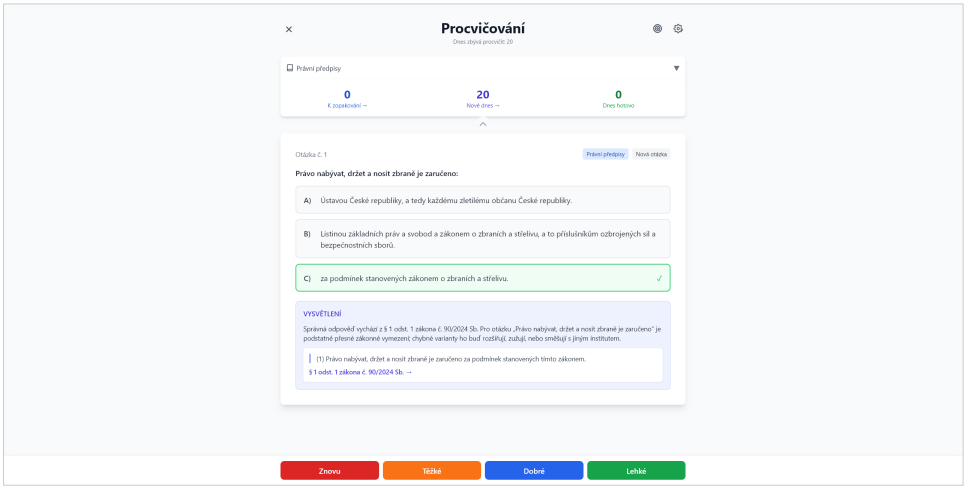
Obrázek A.1: Úvodní obrazovka při první návštěvě.



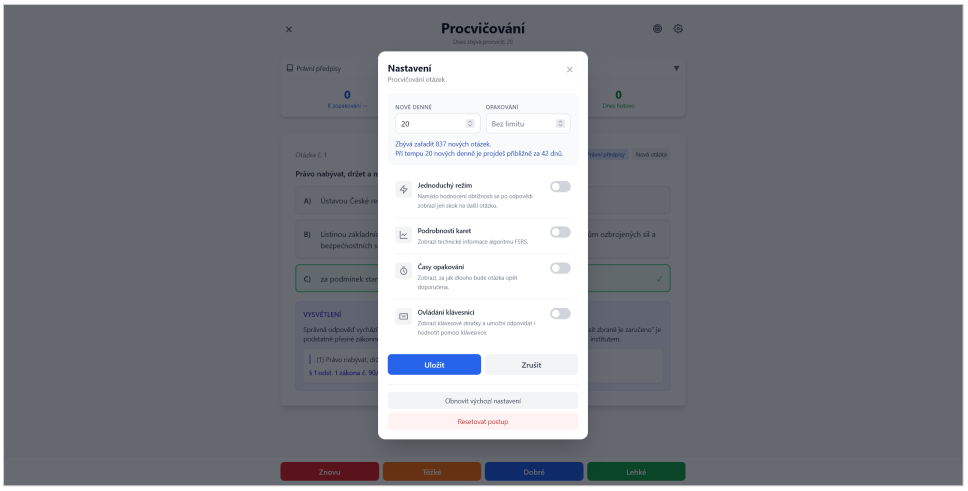
Obrázek A.2: Úvodní obrazovka přihlášeného uživatele se staženými daty pro offline funkcionalitu.



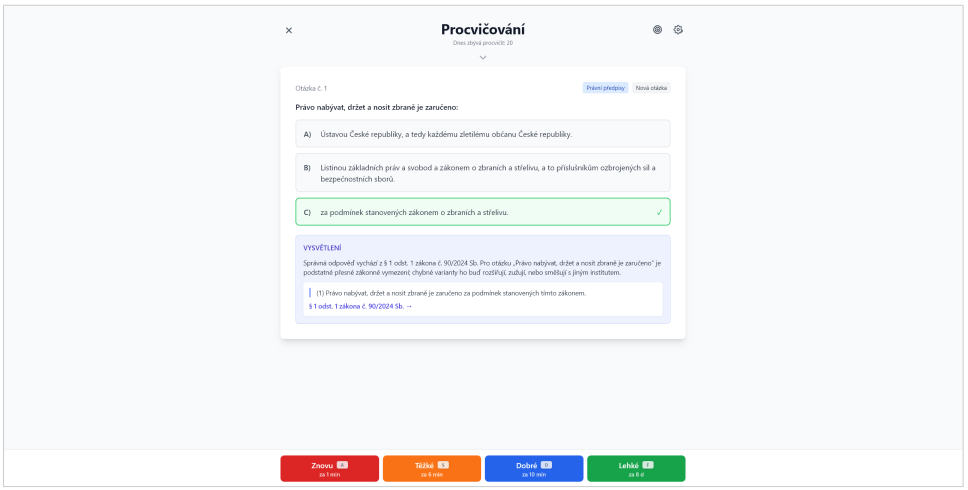
Obrázek A.3: Procházení databáze testových otázek.



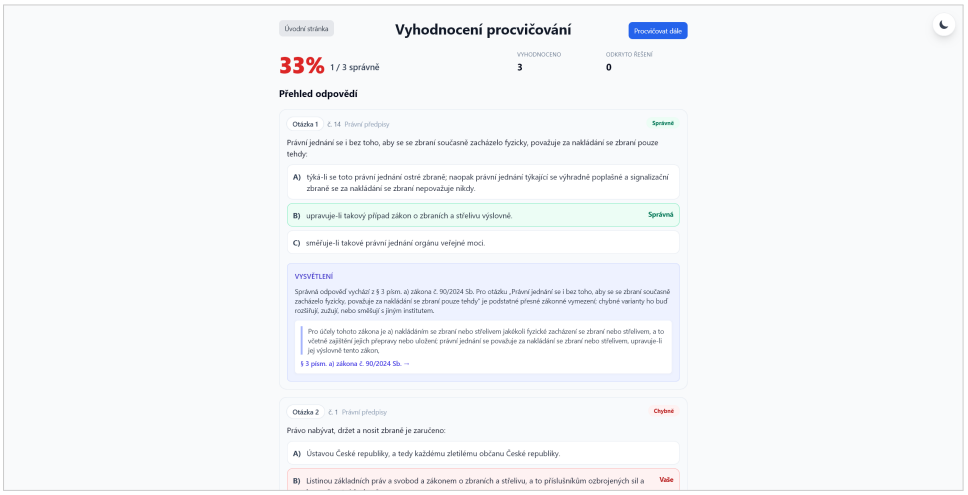
Obrázek A.4: Procvičování testových otázek.



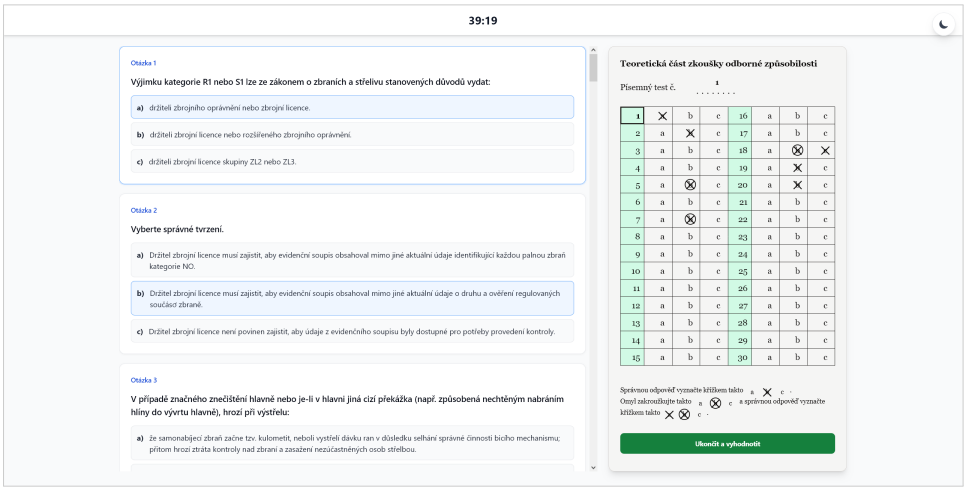
Obrázek A.5: Volitelná nastavení procvičování.



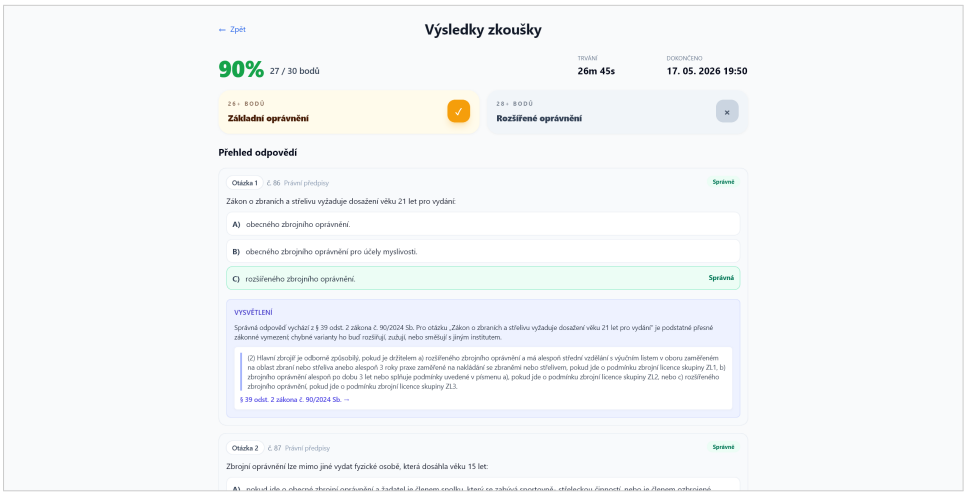
Obrázek A.6: Úprava nastavení procvičování podle zpětné vazby od uživatelů. Je zapnuté ovládání klávesnicí a zobrazení časů dalších opakování.



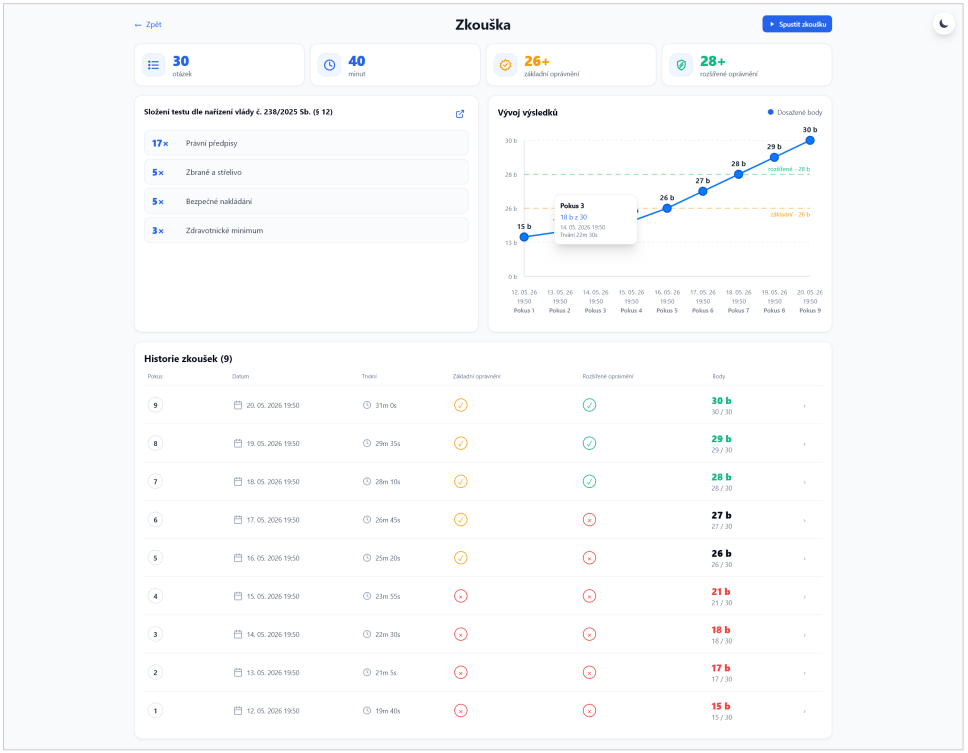
Obrázek A.7: Vyhodnocení relace procvičování.



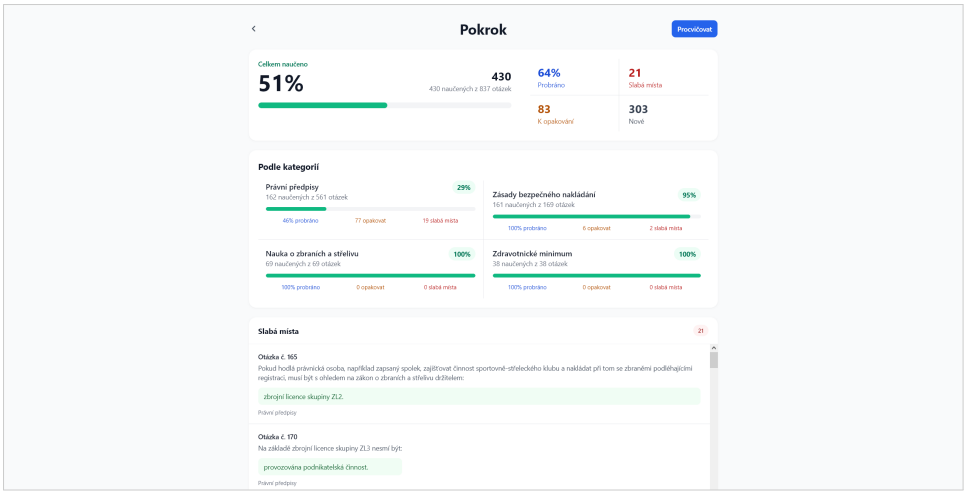
Obrázek A.8: Průběh simulace zkoušky.



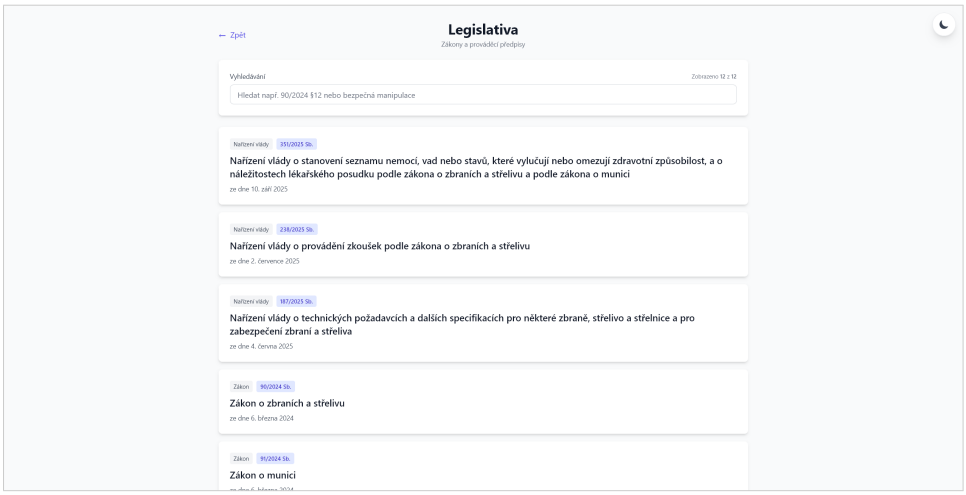
Obrázek A.9: Vyhodnocení simulace zkoušky.



Obrázek A.10: Přehled pokroku a historie zkoušek.



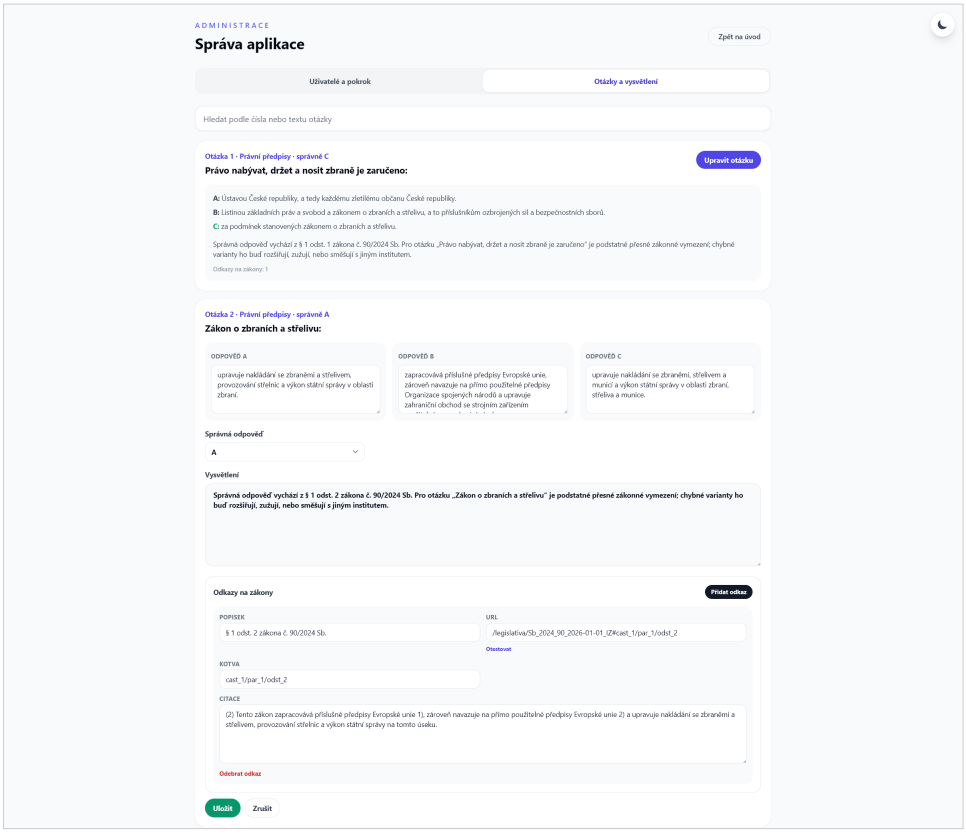
Obrázek A.11: Přehled studijního postupu uživatele.



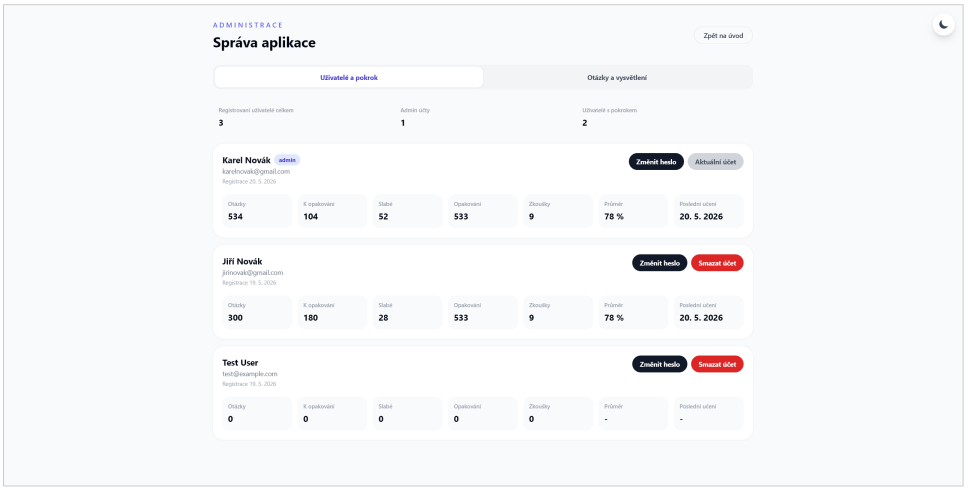
Obrázek A.12: Přehled legislativních podkladů.



Obrázek A.13: Zobrazení podrobnosti vyhlášky a možnost kopírovat odkaz na danou část.



Obrázek A.14: Administrační rozhraní správy otázek.



Obrázek A.15: Administrační rozhraní správy uživatelů.