

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

**TÍMOVÝ PROJEKT
SEMINÁRNA PRÁCA**

2025

Vedúci práce:
Ing. Stanislav Marochok
Študenti:
Bc. Bence Both
Bc. Matyas Horvath
Bc. Jozef Nyitrai
Bc. Vincent Pálfy
Bc. Aron Tükör

Obsah

Úvod	1
1 Požiadavky (Zadanie)	2
2 Teória o historických technikách šifrovania	8
2.1 Substitučné šifry	8
2.2 Transpozičné šifry	9
2.3 Kódové systémy	9
2.4 Nomenklátor:	10
3 Analýza trhu	11
3.1 Transkribus	11
3.2 READ-COOP	12
3.3 De-Crypt	13
3.4 Porovnanie riešení	14
3.5 Záver analýzy trhu	14
4 Návrh implementacie	15
4.1 Use case diagram	18
4.2 Diagramy aktivít	20
4.3 Class diagram	23
5 Nastavenie Webového Servera pre zápisnicu	27
5.1 Úvod	27
5.2 Inštalácia Operačného Systému	27
5.3 Inštalácia Potrebných Komponentov	28
6 Implementácia	31
6.1 Hlavná stránka projektu	31
6.1.1 Podstránka projektových detajlov	33
6.1.2 Podstránka tímu	34
6.1.3 Podstránka FAQ	35
6.1.4 Podstránka swagger API	36
6.2 Registrácia/Prihlásenie	38
6.2.1 Implementácia Bearer Token v Aplikácii	41
6.2.2 Implementácia prihlásenia cez Google účet	42

6.3	Uvítacia stránka po prihlásení	43
6.4	Podstránka profilu prihláseného používateľa	44
6.5	Implementácia funkcie načítania obrázkov	45
6.6	Nástroje	48
6.6.1	Sekcia segmentácie	49
6.6.2	Sekcia analýzy	51
6.6.3	Sekcia segmentácia symbolov	53
6.6.4	Sekcia úpravy JSON výstupov	55
6.6.5	Sekcia dešifrovania	57
6.7	Knižnica dokumentov	59
6.7.1	Úprava dokumentov	61
6.7.2	Vymazanie dokumentu	63
6.8	PHP-Python Komunikácia v Webovej Aplikácii	64
6.9	Priebeh vývoja	65
6.10	Nasadenie a verejný prístup	66
6.11	Testovanie	67
7	Zápisnice	68
7.1	1. semester - 1. týždeň	68
7.2	1. semester - 2. týždeň	70
7.3	1. semester - 3. týždeň	74
7.4	1. semester - 4. týždeň	78
7.5	1. semester - 5. týždeň	81
7.6	1. semester - 6. týždeň	84
7.7	1. semester - 7. týždeň	87
7.8	1. semester - 8. týždeň	90
7.9	1. semester - 9. týždeň	93
7.10	1. semester - 10. týždeň	96
7.11	2. semester - 1. týždeň	98
7.12	2. semester - 2. týždeň	100
7.13	2. semester - 3. týždeň	103
7.14	2. semester - 4. týždeň	104
7.15	2. semester - 5. týždeň	107
7.16	2. semester - 6. týždeň	110
7.17	2. semester - 7. týždeň	111

7.18 2. semester - 8. týždeň	114
7.19 2. semester - 9. týždeň	117
7.20 2. semester - 10. týždeň	120
8 Zoznam hotových stránok	124
Záver	126
Zoznam použitej literatúry	127
Prílohy	I
A Štruktúra elektronického nosiča	II

Zoznam obrázkov a tabuliek

Obrázok 1	Diagram pre spracovanie klúčov	15
Obrázok 2	Diagram pre spracovanie textov	16
Obrázok 3	Use case diagram	19
Obrázok 4	Activity diagram	22
Obrázok 5	Class diagram	26
Obrázok 6	Inštalácia Raspberry OS Lite	27
Obrázok 7	Nastavenie hostiteľského mena a cielovej IP adresy na webovej stránke NO-IP	29
Obrázok 8	Hlavná webová stránka projektu	32
Obrázok 9	Notifikácia o použití cookies	32
Obrázok 10	Podstránka s detajlmí projektu	33
Obrázok 11	Podstránka tímu	34
Obrázok 12	Podstránka často kladených otázok	35
Obrázok 13	Registračná stránka webovej aplikácie	39
Obrázok 14	Formulár pre verifikáciu e-mailovej adresy	39
Obrázok 15	Prihlasovacia stránka webovej aplikácie	40
Obrázok 16	Uvítacia stránka webovej aplikácie	43
Obrázok 17	Podstránka profilu prihláseného používateľa	44
Obrázok 18	Funkcionalita nahranie obrázka na hlavnej stránke	45
Obrázok 19	Nahraný obrázok, pred uložením dokumenta	46
Obrázok 20	Uložený dokument, s presmerovacím tlačidlom	47
Obrázok 21	Sekcia nástrojov v navigačnom paneli	48
Obrázok 22	Pomôcka pre používateľov počas segmentácie	49
Obrázok 23	Nástroj pre segmentáciu textu	50
Obrázok 24	Nástroj pre analýzu dokumentu	52
Obrázok 25	Nástroj pre segmentáciu symbolov	54
Obrázok 26	Pomôcka pre úpravu JSON výstupu	55
Obrázok 27	Nástroj pre úpravu a uloženia JSON výstupu	56
Obrázok 28	Výsledok dešifrovania	57
Obrázok 29	Nástroj desifrovania pomocou už predpracovaného klíča	58
Obrázok 30	Podstránky pre jednotlivé typy dokumentov v navigačnom paneli	59

Obrázok 31	Podstránka zobrazujúca uložené klúče	60
Obrázok 32	Podstránka zobrazujúca uložené šifrované texty	60
Obrázok 33	Podstránka na úpravu jednotlivých klúčov	62
Obrázok 34	Nahraný obrázok podstránka	63
Obrázok 35	Testovanie pomocou Postmanu	67
Obrázok 36	Rozvrh študentov v zimnom semestri	69
Obrázok 37	Inštalácia Raspberry OS Lite	71
Obrázok 38	Nastavenie mena hostu a cieľovej IP adresy na webovej stránke NO-IP.	73
Obrázok 39	Ovládací panel XAMPP	75
Obrázok 40	Hlavná stránka projektu	76
Obrázok 41	Podstránka registrácie	77
Obrázok 42	Podstránka prihlásenia	77
Obrázok 43	Hlavná stránka po prihlásovaní	79
Obrázok 44	Ukážka nahrania obrázka na hlavnej stránke	80
Obrázok 45	Integrácia brány Cloudflare a prístupu	82
Obrázok 46	Menu aplikácie CloudFlare	83
Obrázok 47	CloudFlare public hostname menu	83
Obrázok 48	Ukážka python verzie na serveri	85
Obrázok 49	Ukážka spusteného python skrptu na webstránke	85
Obrázok 50	Rozhranie aplikácie Overleaf	88
Obrázok 51	Rozhranie aplikácie Figma	89
Obrázok 52	Vytvorené tabulky prepojenia používateľov s obrázkami	91
Obrázok 53	Výber fotografie na úpravu	91
Obrázok 54	Ukážka úpravy snímkov	92
Obrázok 55	Zdrojový kód JWT tokenov vo VSC	94
Obrázok 56	Ukážka tokenu na webovej stránke	94
Obrázok 57	Poznámky z konzultácií	97
Obrázok 58	Rozvrh študentov v letnom semestri	99
Obrázok 59	Poznámky z konzultácie	101
Obrázok 60	Prihlasovacia stránka	105
Obrázok 61	Webová aplikácia s ukážkou s nahraným obrazom	105
Obrázok 62	Zmena vo vizualizácii vo webovej aplikácii	108

Obrázok 63	Zmena v databáze	108
Obrázok 64	Nový dizajn hlavnej stránky	112
Obrázok 65	Nový dizajn prihlásovacej stránky	112
Obrázok 66	Aktualizovaná stránka dokumentov	113
Obrázok 67	Overenie e-mailovej adresy	114
Obrázok 68	Podstránka upravenia informácií o používateľovi	115
Obrázok 69	Príklad operácie CRUD	115
Obrázok 70	Podstránka zobrazujúce upravenia dokumentov	118
Obrázok 71	Dokumenty používateľa rozdelené podľa typu - klúče a šifrované texty	119
Obrázok 72	Podstránka často kladených otázok	121
Obrázok 73	Akceptácia uloženie cookies	121
Obrázok 74	Nová vylepšená verzia podstránky profilu	122

Zoznam výpisov

1	Inštalácia Apache webového servera	28
2	Inštalácia PHP Modulu	28
3	Inštalácia MariaDB a PHP modulov	28
4	Nastavenie databázy pre WordPress	28
5	Stiahnutie a príprava WordPress súborov na serveri	29
6	SQL príkaz na vytvorenie tabuľky users pre uchovávanie údajov o registrovaných používateľoch.	38
7	Inštalácia Apache webového servera	71
8	Inštalácia PHP Modulu	71
9	Nastavenie databázy pre WordPress	72
10	Stiahnutie a príprava WordPress súborov na serveri	72

Úvod

Historické šifrované dokumenty, konkrétnie rukopisné šifrovacie kľúče a šifrované texty, sú významným zdrojom informácií o dobovej komunikácii a jej zabezpečení. Tieto dokumenty, známe aj ako nomenklátory, predstavovali v minulosti základný nástroj na šifrovanie a dešifrovanie tajných správ. Pre ich ručne písaný charakter a často zložitú štruktúru je však ich spracovanie časovo náročné a vyžaduje vysokú úroveň odbornosti.

Cieľom tejto aplikácie je automatizovať a urýchliť spracovanie rukopisných šifrovacích kľúčov a šifrovaných textov, čím sa zefektívni práca odborníkov a sprístupní tento proces širšiemu publiku. Digitalizované šifrovacie kľúče výrazne urýchlia dešifrovanie týchto textov a umožnia systematické uchovávanie historických údajov. Aplikácia zároveň poskytne priestor na organizáciu a kategorizáciu týchto dokumentov, čo uľahčí ich analýzu a ďalší výskum.

Navrhovaný systém bude využívať kombináciu manuálnych a poloautomatických nástrojov, pričom umožní používateľom lokalizovať a kategorizovať texty, identifikovať šifrovacie mechanizmy a dešifrovať správy s využitím digitalizovaných kľúčov. Interaktívne užívateľské rozhranie bude navrhnuté tak, aby uľahčilo prácu s aplikáciou, a systém bude pripravený na integráciu pokročilých algoritmov alebo AI modulov, ktoré ešte viac zefektívnia celý proces.

Hlavným prínosom tejto aplikácie je skrátenie času potrebného na spracovanie rukopisných dokumentov a ich dešifrovanie. Tým sa otvára priestor na hlbšie pochopenie dobovej komunikácie, sprístupnenie cenných historických informácií a rozšírenie možností výskumu v oblasti kryptológie. Tento projekt spája moderné technologické riešenia s potrebou zachovania kultúrneho dedičstva a zvyšuje efektivitu práce s historickými šifrovanými textami.

1 Požiadavky (Zadanie)

Your task is to develop a web application for processing historical handwritten documents—encrypted texts and substitution cipher keys—using simulated AI modules. The application should be built on the .NET platform, with Python used for AI simulations, allowing future integration of real AI models.

The application must classify uploaded documents as either encrypted texts or substitution cipher keys using a simulated AI module. For encrypted texts, users should localize the text, transcribe it, and decrypt it using brute-force algorithms or uploaded substitution cipher keys. For cipher keys, the application should localize and classify substitution sub-systems, detect substitution items, and map plain text to cipher text. The system must allow users to apply a new cipher key to existing encrypted texts to check compatibility and vice versa. The application must also include an interactive tutorial to guide new users on how to use the system effectively.

Students must develop their own solutions for various challenges, thinking critically about design, integration, and functionality. This project requires creativity and independent problem-solving.

The front-end should be responsive and visually appealing, enhancing user experience. The back-end must communicate with Python-based AI modules and implement JWT Bearer authorization for security. Users should also be able to save their progress and resume it later.

Deliverables include a functional web application, source code, technical documentation, a user manual, and a presentation. Collaborate effectively using version control, follow coding standards, and ensure thorough testing.

Ponuka pre tímový projekt

Web Application for Historical Handwritten Documents Processing

1. Tím

Zadávateľ tímového projektu:

Ing. Stanislav Marochok (*email: stanislav.marochok@stuba.sk*)

Téma:

Web Application for Historical Handwritten Documents Processing

Anotácia:

Your task is to develop a web application for processing historical handwritten documents, encrypted texts and substitution cipher keys - using simulated AI modules. The application should be built on the .NET platform, with Python used for AI simulations, allowing future integration of real AI models.

The application must classify uploaded documents as either encrypted texts or substitution cipher keys using a simulated AI module. For encrypted texts, users should localize the text, transcribe it, and decrypt it using brute-force algorithms or uploaded substitution cipher keys. For cipher keys, the application should localize and classify substitution sub-systems, detect substitution items, and map plain text to cipher text. The system must allow users to apply a new cipher key to existing encrypted texts to check compatibility and vice versa. The application must also include an interactive tutorial to guide new users on how to use the system effectively.

Students must develop their own solutions for various challenges, thinking critically about design, integration, and functionality. This project requires creativity and independent problem-solving.

The front-end should be responsive and visually appealing, enhancing user experience. The back-end must communicate with Python-based AI modules and implement JWT Bearer authorization for security. Users should also be able to save their progress and resume it later.

Deliverables include a functional web application, source code, technical documentation, a user manual, and a presentation. Collaborate effectively using version control, follow coding standards, and ensure thorough testing.

Členovia tímu:

Bc. Bence Both

- *Email:* xbothb@stuba.sk
- *Skills:*
 - Expert: Python, Java, Linux
 - Pokročilý: JavaScript, C, C++, SQL
 - Skúsenosť: Html, CSS, WordPress

Bc. Matyas Horváth

- *Email:* xhorvathm2@stuba.sk
- *Skills:*
 - Expert: Linux, Python, C#, .NET
 - Pokročilý: Html, Css, JavaScript, Java
 - Skúsenosť: JQuery, Bootstrap, C, C++

Bc. Jozef Nyitrai

- *Email:* xnyitrai@stuba.sk
- *Skills:*
 - Expert: Html, Css, JavaScript, Linux, PHP
 - Pokročilý: SQL, C, C++, Java, Python
 - Skúsenosť: WordPress, JQuery, BootStrap

Bc. Áron Tükör

- *Email:* xtukor@stuba.sk
- *Skills:*
 - Expert: Python, HTML, JavaScript, PHP
 - Pokročilý: CSS, Java, C, C++, Linux
 - Skúsenosť: WordPress, JQuery, BootStrap

Bc. Vincent Pálfi

- *Email:* xpalfy@stuba.sk
- *Skills:*
 - Expert: Html, Css, JavaScript, SQL, PHP
 - Pokročilý: Linux, C, C++, Java, Python
 - Skúsenosť: WordPress, JQuery, BootStrap

2. Motivácia

Všetci členovia tímu absolvovali bakalárské štúdium aplikovanej informatiky na Slovenskej technickej univerzite v Bratislave na Fakulte Elektrotechniky a Informatiky.

Pri výbere témy projektu sme zohľadnili zručnosti členov nášho tímu a oblasti záujmu v rámci oddelenia IT. Sme presvedčení, že rozdelenie tímu a rozdelenie úloh zodpovedá požiadavkám projektu.

Dvaja členovia nášho tímu, Bc. Bence Both a Bc. Matyas Horváth, sa počas svojho predchádzajúceho štúdia špecializovali na bezpečnosť informačných systémov (BIS) a v súčasnosti pokračujú v štúdiu v tejto oblasti. Vďaka tomu už ovládajú techniky potrebné na zvýšenie bezpečnosti projektu, najmä v oblasti vývoja back-endu.

Ďalší členovia tímu, Bc. Jozef Nyitrai, Bc. Áron Tükör a Bc. Vincent Pálfy, sa špecializujú na modelovanie a simuláciu udalostných systémov (MSUS) a v súčasnosti pokračujú v štúdiu v oblasti inteligentných softvérových systémov (ISS). Boli by zodpovední za front-end dizajn projektu s osobitným dôrazom na implementáciu užívateľsky výhodného rozhrania.

Každý člen nášho tímu by sa okrem rôznych úloh projektu aktívne podieľal na jeho celkovom vývoji, čím by sa zabezpečila vysoká kvalita realizácie úloh. Našou hlavnou motiváciou pri práci je získavať ďalšie vedomosti a skúsenosti a dodať projekt, ktorý splní očakávania a bude vysoko kvalitný. Snažíme sa dodať projekt v kvalite, ktorá môže slúžiť ako cenná a užitočná referencia pre ostatných.

3. Čo môžeme poskytnúť

Prvým krokom vo vývoji projektu je vytvorenie prezentačnej webovej stránky, ktorá umožní postupnú aktualizáciu dokumentácie a obsahu webovej stránky v priebehu projektu. Na sprístupnenie webovej stránky prostredníctvom internetu je potrebné nakonfigurovať webový server, na čo náš tím plánuje použiť dostupný mikropočítač Raspberry Pi 3.

Ako alternatívne riešenie sa zvažoval vývoj webovej stránky založenej na systéme WordPress, ktorej realizácia podlieha schváleniu vedúceho témy projektu. Táto platforma by ponúkla možnosť zjednodušiť správu obsahu a uľahčiť údržbu webovej stránky. Na zabezpečenie prístupu na internet by bol systém pripojený k celosvetovej sieti prostredníctvom domáceho Wi-Fi routera, kde by bol webový server sprístupnený verejnosti zapnutím funkcie presmerovania portov. Toto riešenie by zabezpečilo správne pripojenie a bezpečný tok údajov medzi používateľmi a serverom. Okrem týchto dvoch riešení sú k dispozícii aj platené doménové služby, ktoré by ušetrili čas strávený nastavovaním servera, ktorý by sa potom mohol využiť na vývoj projektu.

Druhá fáza projektu sa zameriava na vývoj aplikácie, ktorá zahŕňa aplikáciu založenú na platforme .NET. Počas vývoja sa bude venovať osobitná pozornosť používateľskému rozhraniu, ktoré by malo byť nielen estetické, ale aj citlivé a používateľsky prívetivé, čím sa zabezpečí jednoduchá a intuitívna používateľská skúsenosť. Okrem toho v rámci projektu vytvoríme aj výukové video, ktoré bude krok za krokom poskytovať návod na používanie webovej aplikácie.

Počas vývoja backendu bude prioritou integrácia a analýza modulov umelej inteligencie založených na jazyku Python, ktoré budú tvoriť základ algoritmu rozpoznávania obrazu. V tejto súvislosti bude kľúčové dôkladné štúdium dokumentácie a praktické používanie modulov.

Po stránke bezpečnosti sa bude osobitná pozornosť venovať implementácii autorizačného mechanizmu založeného na nosiči JWT, ktorý bude posilnený ďalšími bezpečnostnými opatreniami. To je nevyhnutné na zabezpečenie toho, aby používatelia mohli bezpečne uložiť svoju reláciu a následne v nej bez problémov pokračovať. Tieto aspekty projektu zabezpečujú spoľahlivosť a vysokú úroveň bezpečnosti aplikácie.

Na efektívnu a účinnú prácu je nevyhnutné podrobne rozdelenie práce, ktoré umožňuje členom tímu špecializovať sa na vývoj back-endových aj front-endových komponentov. Pri vývoji back-endu sa zameriavame na spoľahlivú správu údajov a efektívne spracovanie požiadaviek používateľov, zatiaľ čo pri vývoji front-endu sa sústredíme na vytvorenie používateľsky prívetivého rozhrania a implementáciu responzívneho dizajnu. Kľúčovým aspektom je jednoduchosť logiky úloh, ale aj efektívnosť, čo prispieva k dlhodobej udržateľnosti a škálovateľnosti projektu.

Na zabezpečenie transparentnosti a vysokej úrovne spolupráce v procese vývoja budeme používať systém správy verzií GitHub. Ten umožní sledovanie zmien v kóde, spájanie rôznych vývojových vetiev a porovnanie verzíí, čo výrazne uľahčí komunikáciu a spoluprácu medzi tímom.

Nepretržitá a úzka spolupráca medzi členmi tímu je nevyhnutná pre úspešnú realizáciu projektu. Okrem realizácie projektu sa osobitná pozornosť venuje príprave podrobnej a aktuálnej dokumentácie s cieľom zabezpečiť transparentnosť a odbornú kvalitu. Ústrednou úlohou je aj neustále vylepšovanie a optimalizácia prezentačnej webovej stránky, ktorá podporuje prezentáciu projektu a jeho hodnotu pre používateľov.

Na zabezpečenie plynulého postupu prác sa konajú pravidelné stretnutia a konzultácie tímu s cieľom preskúmať aktuálny stav projektu, vyriešiť všetky vzniknuté problémy a rýchlo reagovať na nové potreby, ktoré sa objavia. Súčasťou práce je aj vedenie potrebných zápisov z konzultácií, ktoré by tím tiež uchovával na webovej stránke, ktorá slúži na dokumentáciu. Okrem toho sme v neustálom kontakte s naším

vedúcim, aby sme projekt prispôsobili vašim potrebám a zabezpečili, že konečný výsledok splní očakávania v každom ohľade. Náš tím sa snaží zachovať efektívnosť spolupráce, ktorá je kľúčovým faktorom úspešného a včasného dodania projektu.

4. Predpokladané zdroje

- *Jazyky:* Python, Html, Css, JavaScript, PHP, SQL, UML
- *Software:* JetBrains (PhpStorm, WebStorm, PyCharm), Visual Studio Code, WordPress, .NET, Azure
- *Hardware:* Raspberry Pi 3 bude fungovať ako webový server pre WordPress, okrem toho budeme používať vlastné počítače. Prípadne bude používané iné platné webové služby, napríklad Azure.

5. Aktuálny rozvrh všetkých členov tímu

	8:00 - 8:50	9:00 - 9:50	10:00 - 10:50	11:00 - 11:50	12:00 - 12:50	13:00 - 13:50	14:00 - 14:50	15:00 - 15:50	16:00 - 16:50	17:00 - 17:50
Pondelok						I-UPB (1,2,3,4,5)		I-UPB (1,2,3,4,5)		
Úterok			I-RZZ / I-ZKRY (1,2,3,4,5)							
Streda	I-LOG (1,2,3,4,5)		I-LOG (1,2,3,4,5)			I-ZKRY (1,2)				
Štvrtok	I-SUNS (1,2,3,4,5)		I-RZZ (3,4,5)							
Piatok		I-SUNS (1,3,4,5)		I-SUNS (2)						

Prednáška	Cvičenie	<table border="1"><tr><td>1</td><td>Bc. Bence Both</td></tr><tr><td>2</td><td>Bc. Matyas Horváth</td></tr><tr><td>3</td><td>Bc. Jozef Nyitrai</td></tr><tr><td>4</td><td>Bc. Áron Tükör</td></tr><tr><td>5</td><td>Bc. Vincent Pálfi</td></tr></table>	1	Bc. Bence Both	2	Bc. Matyas Horváth	3	Bc. Jozef Nyitrai	4	Bc. Áron Tükör	5	Bc. Vincent Pálfi
1	Bc. Bence Both											
2	Bc. Matyas Horváth											
3	Bc. Jozef Nyitrai											
4	Bc. Áron Tükör											
5	Bc. Vincent Pálfi											

6. Konštruktívne návrhy zmien organizácie predmetu

Tím nemá žiadne ďalšie pripomienky ani konštruktívnu kritiku k tomuto projektu. Na základe opisu ide o veľmi zaujímavý projekt, ktorému sa všetci členovia tímu venujú s rovnakým nasadením a nadšením.

2 Teória o historických technikách šifrovania

Predtým, ako sa pustíme do implementácie nášho projektu, je nevyhnutné pochopiť základné koncepty historických šifier. Táto sekcia ponúka stručný prehľad klúčových techník.

Prehľad kryptografie v histórii:

Kryptografia, umenie a veda zabezpečenia komunikácie, zohrala klúčovú úlohu v histórii ľudstva. Od starovekých civilizácií až po modernú dobu sa kryptografické techniky používali na ochranu citlivých informácií, zabezpečenie vojenských stratégii a umožnenie dôvernej komunikácie. Skoré kryptografické metódy boli často jednoduché, no zároveň geniálne.

Tri hlavné kategórie historických šifier:

- Substitučné šifry: Nahrádzajú prvky otvoreného textu zodpovedajúcimi prvkami šifrovaného textu.
- Transpozičné šifry: Preusporiadajú poradie znakov otvoreného textu podľa definovaných pravidiel.
- Kódové systémy: Nahrádzajú celé slová alebo frázy preddefinovanými symbolmi alebo číslami.

2.1 Substitučné šifry

Substitučné šifry patria medzi najstaršie a najjednoduchšie formy kryptografie. V týchto systémoch je každý prvak otvoreného textu zvyčajne jednotlivé písmená alebo symboly nahradený iným prvkom na základe preddefinovaného pravidla alebo klúča.

Otvorený text (plaintext) označuje pôvodnú, nezašifrovanú správu alebo text, ktorý má byť chránený pred neželaným prístupom. Po aplikácii šifrovacieho procesu vzniká šifrovaný text (ciphertext), ktorý je nečitateľný bez správneho klúča. Štruktúra otvoreného textu v substitučných šifrach zostáva nezmenená, no identita jednotlivých znakov je pozmenená.

Hlavné charakteristiky

- Jednoznačné mapovanie otvoreného textu na šifrovaný text: Každý znak v otvorenom teste je nahradený zodpovedajúcim znakom v šifrovanom teste. Napríklad v Caesarovej šifre s posunom o 3 sa A stáva D, B sa stáva E, a tak ďalej.

2.2 Transpozičné šifry

Transpozičné šifry šifrujú otvorený text preusporiadáním jeho znakov podľa špecifického vzoru alebo pravidla, bez toho, aby menili samotné znaky. Na rozdiel od substitučných šifier, ktoré nahradzajú znaky inými, transpozičné šifry sa zameriavajú na zmenu poradia znakov v správe.

Hlavná charakteristika

- Znaky zostávajú nezmenené, mení sa však ich poradie: Identita každého znaku v otvorenom teste je zachovaná, ale ich pozicie sú v šifrovanom teste pozmenené.
Napríklad: HELLO sa môže stať OLLEH alebo HLELO.

2.3 Kódové systémy

Kódové systémy šifrujú informácie tým, že nahradzajú celé slová, frázy alebo vety preddefinovanými symbolmi, číslami alebo skupinami znakov. Na rozdiel od šifier, ktoré pracujú na individuálnych znakoch alebo menších jednotkách, kódy fungujú na vyššej jazykovej úrovni, čo ich robí vhodnými na prenos dôležitých kľúčových slov alebo pojmov.

Hlavné charakteristiky

- Použitie kódovej knihy: Kódová kniha obsahuje mapovanie otvorených slov alebo fráz na ich zodpovedajúce kódované reprezentácie.
Príklad: attack → 777, retreat → 999.
- Flexibilita a využiteľnosť: Kódové systémy sú obzvlášť užitočné na kódovanie opakujúcich sa fráz alebo kritických pojmov, ako sú mená, miesta alebo vojenské akcie, čo zjednodušuje komunikáciu a znižuje dĺžku správy.

2.4 Nomenklátor:

Hybridný kryptografický systém, ktorý kombinuje substitučné šifry s kódovými prvkami. Často sa používal medzi 15. a 18. storočím v diplomatickej korešpondencii.

- Klúčové vlastnosti:
 - Zložka substitučnej šifry: Nahrádza jednotlivé písmená alebo malé skupiny písmen.
 - Kódová zložka: Preddefinované symboly alebo čísla predstavujú konkrétné slová alebo frázy.
 - Obrovský počet kombinácií: Substitučná časť už sama o sebe obsahuje veľké množstvo permutácií, ku ktorým sa pridáva aj množstvo kódov v kódovej knihe. Chýbajúca kódová kniha: Ak kódová kniha nebola zackytená, dešifrovanie vyžadovalo obrovské množstvo času a pokusov.
- Zraniteľnosti: Zachytená kódová kniha by mohla celkový systém urobiť nebezpečným.

Takéto šifrovania sú relatívne ľahko dešifrovateľné, ak je k dispozícii správna kódová kniha alebo klúč. Naša platforma presne túto možnosť ponúka: ktokoľvek, kto má šifrovaný text alebo klúče, ich môže nahrať na stránku, kde systém porovnáva tieto údaje so všetkými už uloženými klúčmi a pokúsi sa ich použiť na dešifrovanie textu. Touto funkciou sa snažíme zjednodušiť proces dešifrovania a efektívne využitie klúčov.

3 Analýza trhu

V rámci analýzy trhu sme preskúmali existujúce riešenia na spracovanie dokumentov a optické rozpoznávanie znakov (OCR). Zamerali sme sa na porovnanie ich funkcionálit, technologických možností a vhodnosti pre náš systém. Medzi hlavné konkurenčné produkty patria **Transkribus** [1], **READ-COOP** [2] a **De-Crypt** [3], ktoré poskytujú pokročilé nástroje na digitalizáciu a analýzu textu z historických a moderných dokumentov.

3.1 Transkribus

Transkribus je platforma na báze umelej inteligencie, ktorá sa špecializuje na automatisovanú transkripciu a rozpoznávanie rukopisu (Handwritten Text Recognition – HTR). Vyuvíjaná v rámci projektu READ (Recognition and Enrichment of Archival Documents), je široko využívaná archívmi, knižnicami a výskumnými inštitúciami.

- **Funkcionalita:**

- Automatické rozpoznávanie tlačeného aj písaného textu.
- Možnosť trénovať vlastné modely na špecifické typy dokumentov.
- Export výsledkov do formátov ako PDF, XML, alebo ALTO.

- **Technológia:**

- Využíva HTR+ modely založené na hlbokom učení (Deep Learning).
- Cloudová platforma s možnosťou lokálnej inštalácie pre veľké organizácie.

- **Výhody:**

- Vysoká presnosť rozpoznávania aj pri historických dokumentoch.
- Flexibilita v prispôsobovaní modelov pre špecifické typy písma.

- **Nevýhody:**

- Plná funkcionalita je dostupná len za poplatok.
- Vyššia náročnosť na výpočtový výkon pri trénovaní vlastných modelov.

3.2 READ-COOP

READ-COOP je kooperatívna platforma zameraná na kolaboratívne spracovanie dokumentov. Vychádza z projektu Transkribus a ponúka kommerčné aj open-source riešenia pre organizácie aj jednotlivcov.

- **Funkcionalita:**

- Podpora strojového učenia na rozpoznávanie textu.
- Možnosť spolupráce viacerých používateľov pri anotácii a korektúre.
- Integrácia s archívnymi systémami na dlhodobú archiváciu.

- **Technológia:**

- Používa pokročilé neuronové siete na HTR a OCR.
- Hybridný prístup – cloudové a on-premise riešenia.

- **Výhody:**

- Podpora kolaborácie a crowdsourcingu pri spracovaní dokumentov.
- Rozšíriteľnosť a podpora rôznych jazykov a typov písma.

- **Nevýhody:**

- Implementácia vlastných riešení vyžaduje technické znalosti.
- Niektoré pokročilé funkcie sú dostupné len v platenej verzii.

3.3 De-Crypt

De-Crypt je moderný nástroj určený na dešifrovanie historických rukopisov a rozpoznávanie textu. Zameriava sa na automatizované rozpoznávanie zložitých textových štruktúr a ich prepis.

- **Funkcionalita:**

- Automatická transkripcia historických textov vrátane skratiek.
- Analýza štruktúrovaných dokumentov s rozpoznaním textových polí.
- Možnosť kombinácie OCR s manuálnou korekciou.

- **Technológia:**

- Využíva Convolutional Neural Networks (CNN) a Recurrent Neural Networks (RNN).
- API integrácia na priamu implementáciu do iných systémov.

- **Výhody:**

- Presnosť pri rozpoznávaní historických a komplexných dokumentov.
- Podpora prispôsobenia modelov pre špecifické typy rukopisov.

- **Nevýhody:**

- Vyžaduje výkonnú infraštruktúru pre veľké datasety.
- Obmedzená podpora pre moderné jazyky mimo historických textov.

3.4 Porovnanie riešení

Nasledujúca tabuľka poskytuje prehľadné porovnanie troch riešení na rozpoznávanie textu: Transkribus, READ-COOP a De-Crypt. Porovnávame kľúčové vlastnosti, ako je schopnosť rozpoznať rukopisné a tlačené texty, možnosti prispôsobenia modelov, podpora kolaborácie, formáty exportu dát, náročnosť implementácie a licenčný model.

Vlastnosť	Transkribus	READ-COOP	De-Crypt
Rozpoznávanie písma	Tlačené a rukopisné	Tlačené a rukopisné	Rukopisné (hlavne historické)
Prispôsobenie modelov	Áno	Áno	Áno
Podpora kolaborácie	Obmedzená	Plná podpora	Obmedzená
Export dát	PDF, XML, ALTO	PDF, XML, JSON	XML, JSON
Implementačná náročnosť	Stredná	Vysoká	Vysoká
Licenčný model	Freemium	Open-source a komerčný	Komerčný

3.5 Záver analýzy trhu

Na základe analýzy dostupných riešení je zrejmé, že moderné OCR systémy poskytujú vysokú presnosť pri spracovaní dokumentov a možnosť prispôsobiť modely pre špecifické potreby. Transkribus a READ-COOP ponúkajú robustné nástroje na rozpoznávanie a spoluprácu, zatiaľ čo De-Crypt sa špecializuje na dešifrovanie komplexných historických rukopisov.

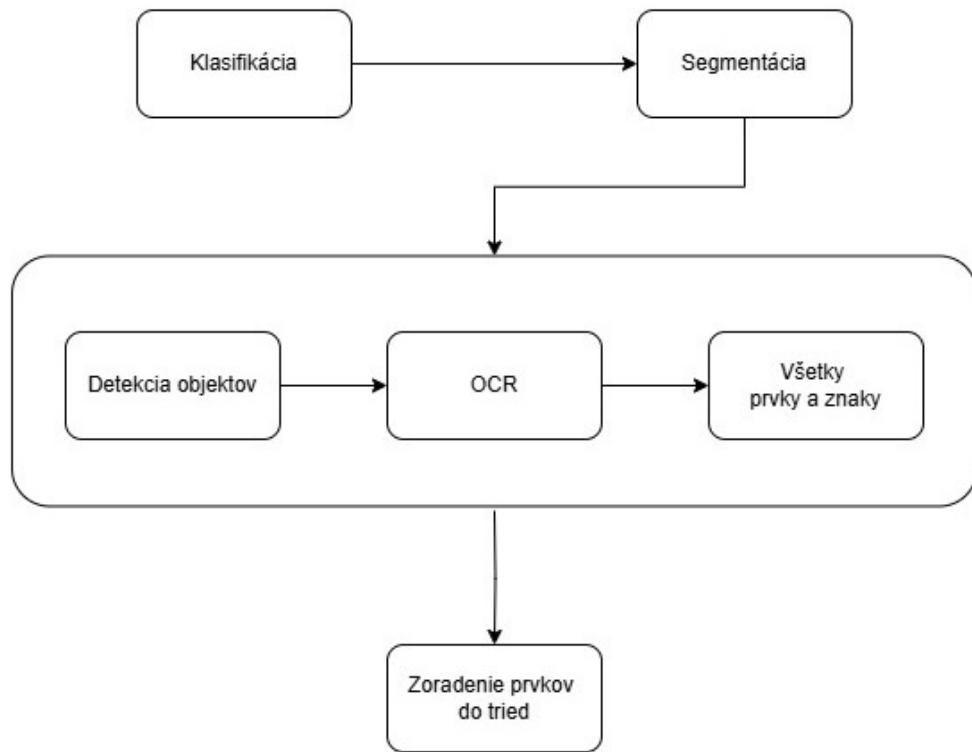
Pre náš systém je kľúčové vybrať riešenie, ktoré kombinuje vysokú presnosť, prispôsobiteľnosť modelov a jednoduchú integráciu. Vzhľadom na tieto kritériá je READ-COOP vhodnou voľbou pre prípady, kde je potrebná rozsiahla kolaborácia a prispôsobenie, zatiaľ čo Transkribus je efektívnejší pre štandardné OCR riešenia s možnosťou trénovania vlastných modelov.

4 Návrh implementacie

Projekt je zameraný na vývoj algoritmu na spracovanie historických dokumentov a záznamov, pričom jeho hlavným cieľom je vytvorenie webovej aplikácie, ktorá umožní spracovanie a dešifrovanie zašifrovaných dokumentov a kľúčov. Táto aplikácia pozostáva z viacerých modulov, ktoré spolupracujú na spracovaní vstupných obrazových súborov a ich následnom dešifrovaní.

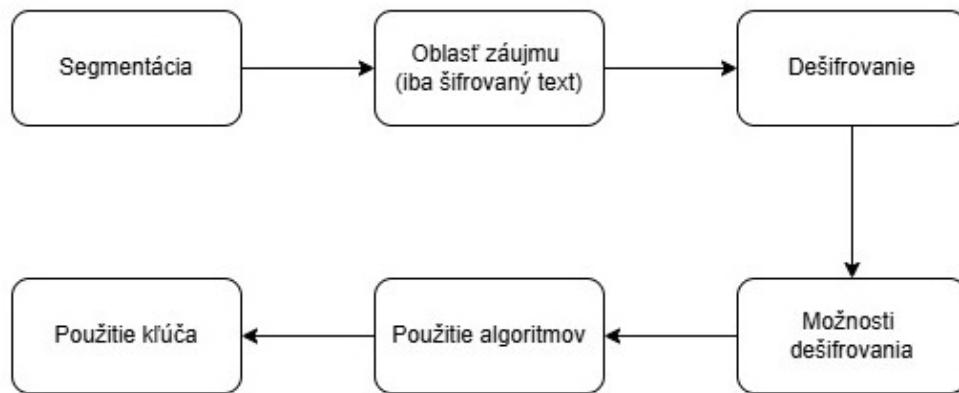
Algoritmus bude analyzovať dva hlavné typy dokumentov:

- Obrazové súbory obsahujúce šifrovacie kľúče.



Obr. 1: Diagram pre spracovanie kľúčov

- Obrazové súbory obsahujúce zašifrované správy, ktoré systém po spracovaní dešifruje.



Obr. 2: Diagram pre spracovanie textov

Proces spracovania je navrhnutý nasledovne:

1. Nahrávanie a segmentácia obrázkov:

- Aplikácia umožní používateľom nahráť obrazové súbory, ktoré obsahujú šifrovanie klúče alebo zašifrované texty.
- V prvom module bude k dispozícii nástroj na segmentáciu obrázkov, ktorý umožní identifikovať a vystrihnúť jednotlivé časti obrazu. Tieto časti predstavujú buď klúče, alebo segmenty zašifrovaného textu.

2. Uloženie a organizácia údajov:

- Segmentované obrázky budú uložené v aplikácii, pričom budú kategorizované do samostatných sekcií pre šifrovacie klúče a zašifrované texty. Používateľ tak získa prehľad o dostupných zdrojoch pre ďalšie spracovanie.

3. Nástroj na mapovanie klúčov:

- Používateľ bude môcť vybrať obrazové segmenty obsahujúce šifrovacie klúče a pomocou nástroja na mapovanie priradiť znaky zo šifrovacieho textu k ich otvoreným textovým ekvivalentom. Tento proces umožní vytvorenie mapy znakov alebo slov, ktorá sa následne použije na dešifrovanie.

4. Dešifrovanie textu:

- Po namapovaní šifrovacích kľúčov bude môcť používateľ použiť tieto mapy v ďalšom module na dešifrovanie zašifrovaných textov. Výsledkom bude otvorený text, ktorý bude možné zobraziť alebo uložiť.

Dôležitým aspektom projektu je, že cieľom nie je implementácia samotných spracovateľských alebo dešifrovacích modulov. Naším cieľom je navrhnúť a vytvoriť webový rámec, ktorý umožní jednoduchú integráciu týchto modulov. Tento rámec poskytne flexibilitu a škálovateľnosť, aby mohli byť jednotlivé moduly pridávané, upravované alebo vylepšované podľa potrieb projektu.

Výsledná aplikácia ponúkne používateľsky prívetivé rozhranie, ktoré uľahčí prácu s historic-kými dokumentmi a umožní efektívne dešifrovanie pomocou kombinácie poloautomatických a manuálnych nástrojov.

4.1 Use case diagram

Aby sme lepšie pochopili a efektívne navrhli jednotlivé funkcionality systému, rozhodli sme sa vizualizovať jeho základné procesy a interakcie pomocou diagramov. Tieto diagramy slúžia na znázornenie prepojení medzi používateľmi a modulmi aplikácie, ako aj na detailné zobrazenie jednotlivých krokov v spracovaní dát.

Autentifikácia a správa používateľov

Na úvod sme sa zamerali na procesy autentifikácie a správy používateľov. Diagram znázorňuje, ako používateľ prechádza jednotlivými krokmi od registrácie až po prihlásenie, pričom systém generuje autentifikačný token, ktorý zabezpečuje bezpečnosť a kontrolu prístupu. Táto vizualizácia nám pomohla jasne definovať vzťah medzi jednotlivými operáciami, ako sú registrácia, prihlásenie a odhlásenie.

Zobrazovanie a kategorizácia údajov

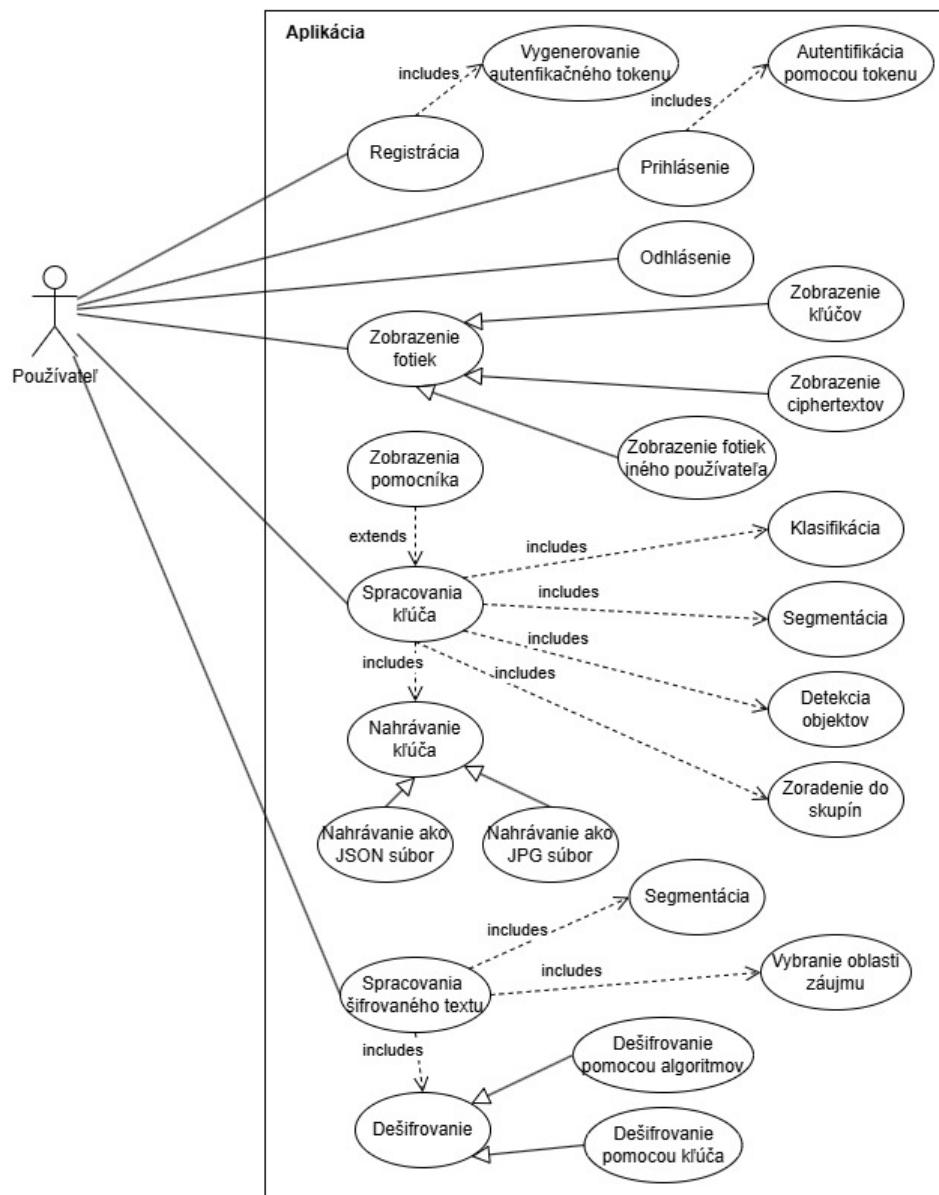
Na lepsie pochopenie spracovania nahraných súborov sme vytvorili diagram znázorňujúci interakcie používateľa s modulmi na zobrazovanie fotografií. Používateľ môže prezerať vlastné aj cudzie nahrané súbory, pričom tieto sú spracované modulmi na klasifikáciu, segmentáciu a detekciu objektov. Tieto kroky zabezpečujú presné rozdelenie šifrovacích klúčov a šifrovaných textov do vhodných kategórií, čím sa uľahčuje ich ďalšie spracovanie.

Spracovanie šifrovacích klúčov a textov

Pre moduly zamerané na spracovanie šifrovacích klúčov a šifrovaných textov sme vytvorili podrobnejšie diagramy, ktoré ukazujú, ako používateľ interaguje so systémom pri nahrávaní a spracovaní dát. Šifrovacie klúče je možné nahrávať vo formáte JSON alebo JPG a následne sú spracované segmentačnými nástrojmi na identifikáciu a prípravu na použitie. Pre šifrované texty sme definovali kroky lokalizácie oblastí záujmu, ich spracovanie a následné dešifrovanie pomocou algoritmov alebo nahratých klúčov.

Interaktívny pomocník

Pre lepšie pochopenie navigácie používateľov v systéme sme znázornili, ako pomocník rozširuje možnosti základného zobrazenia fotografií a spracovania údajov. Diagram zdôrazňuje, ako interaktívny pomocník poskytuje rady a podporu pri práci s jednotlivými modulmi aplikácie, čím uľahčuje používanie systému aj pre začiatočníkov.



Obr. 3: Use case diagram

4.2 Diagramy aktivít

Procesy spracovania v systéme Po vytvorení *use case diagramu*, ktorý nám poskytol prehľad o funkciách a interakciách používateľov so systémom, sme sa rozhodli navrhnúť *diagramy aktivít*, ktoré detailne zobrazujú sekvenčiu krokov jednotlivých procesov. Tieto diagramy nám pomohli lepšie pochopiť, ako systém spracováva šifrovacie klúče a šifrované texty, a slúžia ako základ pre implementáciu jednotlivých modulov.

Spracovanie šifrovacieho klúča

Proces spracovania šifrovacieho klúča je znázornený v diagrame aktivít a pozostáva z nasledovných krokov:

- *Nahranie klúča* Používateľ začína nahraním JPG súboru obsahujúceho šifrovací klúč. Tento súbor je vstupom pre ďalšie spracovanie.
- *Klasifikácia* Po nahraní systému prebieha klasifikácia, ktorá identifikuje základné vlastnosti šifrovacieho klúča, ako je jeho formát alebo typ.
- *Segmentácia* Systém následne vykonáva segmentáciu obsahu súboru, čím rozdeľuje klúč na menšie časti, ktoré môžu byť šifrovacie znaky alebo ich skupiny.
- *Detekcia objektov* Detekčné algoritmy identifikujú jednotlivé prvky v segmentovanom obsahu, ako sú symboly, znaky alebo iné relevantné časti šifrovacieho klúča.
- *Zoradenie do tried* Detekované prvky sú automaticky zaradené do tried na základe ich vlastností a funkcií. Tento krok umožňuje efektívnejšiu prácu s klúčom v ďalších častiach procesu.
- *Uloženie výsledkov* Spracované dátá môžu byť uložené:
 - **Ako JSON:** Tento formát je vhodný na ďalšiu analýzu alebo automatizované spracovanie.
 - **Ako JPG:** Výsledný obraz môže byť použitý na vizuálnu kontrolu alebo prezentáciu.

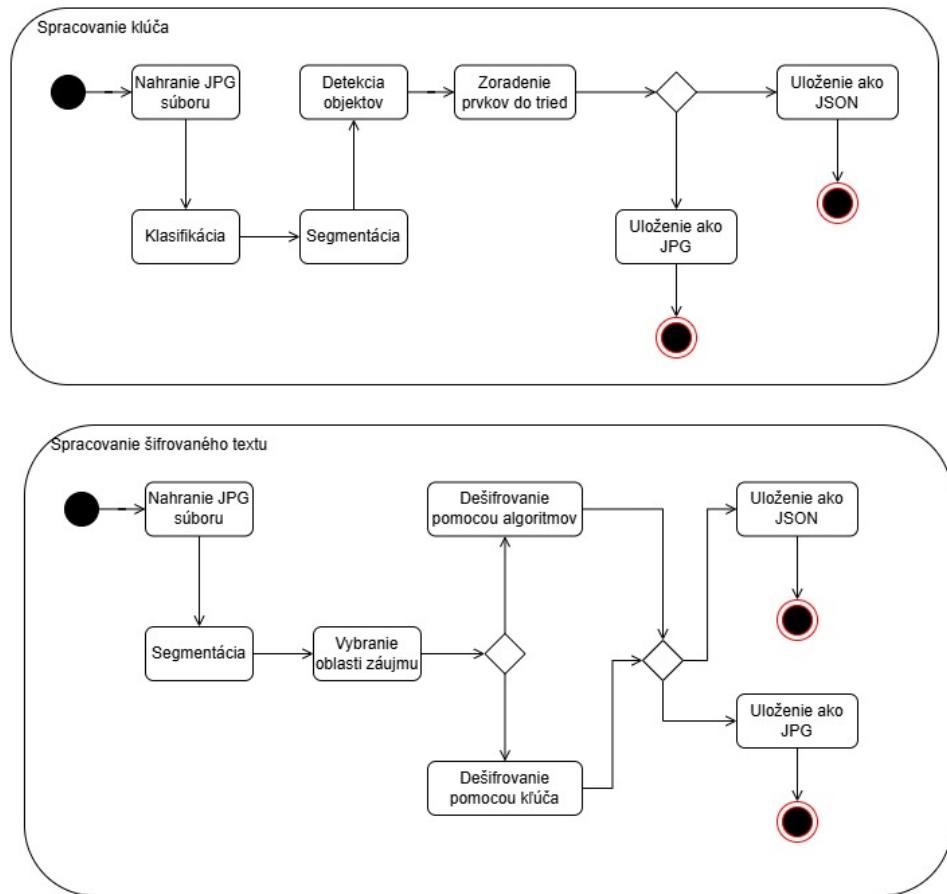
Spracovanie šifrovaného textu

Diagram aktivít pre spracovanie šifrovaného textu znázorňuje kroky, ktoré systém vykonáva na rozlúštenie zašifrovaného obsahu:

- *Nahranie textu* Proces začína nahraním JPG súboru obsahujúceho šifrovaný text. Tento súbor je vstupom pre následnú analýzu.
- *Segmentácia* Systém segmentuje text na menšie časti, čím identifikuje jednotlivé oblasti záujmu vhodné na dešifrovanie.
- *Výber oblasti záujmu* Používateľ alebo algoritmus označí konkrétnu časť šifrovaného textu, ktorá bude dešifrovaná.
- *Dešifrovanie* Vybraná oblasť textu je dešifrovaná jedným z nasledujúcich spôsobov:
 - **Pomocou algoritmov:** Automatizované dešifrovanie na základe implementovaných metód, ako je brute-force alebo iné špecifické algoritmy.
 - **Pomocou šifrovacieho klúča:** Použitie predtým spracovaného a nahraného šifrovacieho klúča na rozlúštenie textu.
- *Uloženie výsledkov* Dešifrované údaje môžu byť uložené v dvoch formátoch:
 - **Ako JSON:** Na ďalšiu analýzu alebo integráciu.
 - **Ako JPG:** Na vizuálne uloženie výsledkov pre používateľov.

Význam diagramov aktivít

Tieto diagramy aktivít nám umožnili vizualizovať jednotlivé kroky spracovania a prepojenia medzi nimi. Poskytujú jasné štruktúry procesu, ktorá je dôležitá nielen pre implementáciu, ale aj pre testovanie a dokumentáciu systému. Týmto spôsobom sme mohli identifikovať klúčové body každého procesu a optimalizovať ich pre efektívne spracovanie dát v rámci aplikácie.



Obr. 4: Activity diagram

4.3 Class diagram

Pre navrhovaný systém sme vytvorili triedny diagram, ktorý popisuje základné entity a ich vzájomné vzťahy. Tento diagram definuje triedy User, Document, Item, Tag a ResultOfProcessing, spolu s ich atribútmi, metódami a vzťahmi.

Trieda User

Trieda **User** reprezentuje entitu používateľa v systéme. Obsahuje základné informácie o používateľovi a metódy, ktoré mu umožňujú interagovať so systémom.

- **Atribúty:**

- – `id: int` – Unikátne identifikačné číslo používateľa.
- – `username: String` – Užívateľské meno.
- – `email: String` – E-mailová adresa používateľa.
- – `password: String` – Heslo používateľa, ktoré je uložené bezpečne.

- **Metódy:**

- + `register()` – Registrácia nového používateľa do systému.
- + `login()` – Prihlásenie používateľa.
- + `logout()` – Odhlásenie používateľa zo systému.
- + `createDocument()` – Vytvorenie nového dokumentu.
- + `shareDocument()` – Zdieľanie dokumentu s inými používateľmi.
- + `processDocument()` – Spracovanie dokumentu v systéme.

Trieda Document

Trieda **Document** reprezentuje dokument vytvorený používateľom. Obsahuje informácie o dokumente a umožňuje jeho manipuláciu.

- **Atribúty:**

- – `id: int` – Unikátne identifikačné číslo dokumentu.
- – `author_id: int` – Identifikátor autora dokumentu.
- – `type: Enum` – Typ dokumentu: (Key, Ciphertext)
- – `title: String` – Názov dokumentu.

- – `description: String` – Popis dokumentu.
- – `shared_with: List<int>` - Zoznam identifikátorov používateľov, s ktorými je dokument zdieľaný. Null znamená, že dokument je private.

- **Metódy:**

- + `uploadItem()` - Nahranie položky do dokumentu.
- + `deleteItem()` - Odstránenie položky z dokumentu.
- + `openItem()` - Otvorenie položky v dokumente.

Trieda Item

Trieda **Item** reprezentuje položku (obrázku) pridanú do dokumentu. Obsahuje informácie o stave a vlastnostiach položky.

- **Atribúty:**

- – `id: int` – Unikátne identifikačné číslo položky.
- – `status: Enum` – Stav položky (napr. koncept, publikovaný).
- – `title: String` – Názov položky.
- – `description: String` – Popis položky.
- – `image_path: String` – Cesta k obrázku položky.
- – `published: Date` – Dátum publikovania položky.
- – `modified: Date` – Dátum poslednej úpravy položky.

Trieda Tag

Trieda **Tag** reprezentuje značku (tag), ktorou je možné kategorizovať položky.

- **Atribúty:**

- – `id: int` – Unikátne identifikačné číslo značky.
- – `tag_name: String` – Názov značky.
- – `description: String` – Popis značky.

Trieda ResultsOfProcessing

Trieda **ResultsOfProcessing** Táto trieda uchováva výsledky spracovania položiek (obrázok) v dokumente.

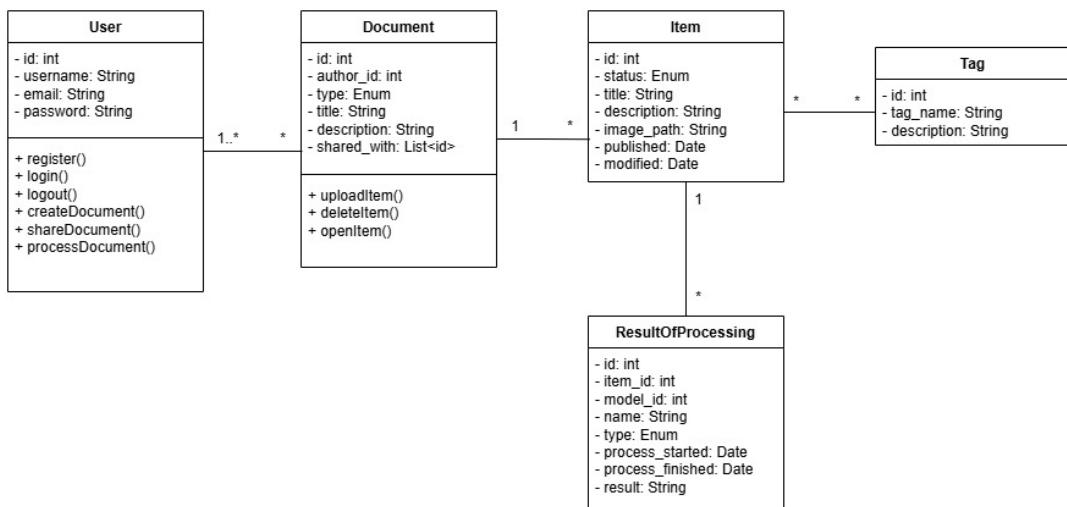
- **Atribúty:**

- – `id: int` – Unikátne identifikačné číslo výsledku.
- – `item_id: int` – Identifikátor položky, ktorá bola spracovaná.
- – `model_id: int` – Identifikátor modelu použitý na spracovanie.
- – `name: String` – Názov spracovania.
- – `type: Enum` – Typ spracovania.
- – `process_started: Date` – Dátum začiatku spracovania.
- – `process_finished: Date` – Dátum ukončenia spracovania.
- – `result: String` – Výsledok spracovania.

Vzťahy medzi triedami

- **User a Document:** Vzťah M:N (Jeden používateľ môže vlastniť viaceré dokumenty, ale aj dokument môže byť vzdialenosť s rôznymi používateľmi). Na opačnej strane, každý dokument má práve jedného autora, identifikovaného atribútom `author_id`.
- **Document a Item:** Vzťah 1:N (Jeden dokument môže obsahovať viaceré položiek).
- **Item a Tag:** Vzťah M:N (Každá položka môže mať pridelenú viacerých značiek, ale aj značky môžu byť pridelené pre rôzne položky).
- **Item a ResultOfProcessing:** Vzťah 1:N (Každá položka môže mať viaceré výsledkov spracovania).

Tento diagram podrobne popisuje hlavné entity systému, ich atribúty, metódy a spôsob ich vzájomnej interakcie.



Obr. 5: Class diagram

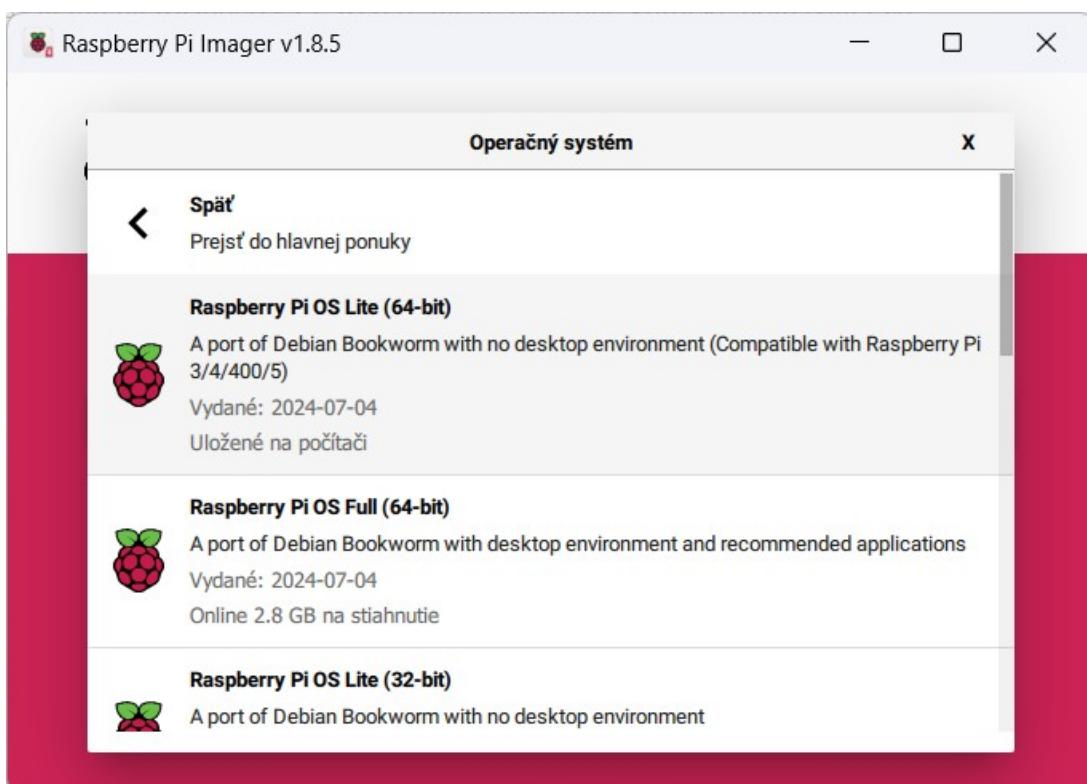
5 Nastavenie Webového Servera pre zápisnicu

5.1 Úvod

Cieľom tohto kroku v projekte bolo nastavenie webového servera, ktorý bude slúžiť ako platforma pre dokumentáciu a protokol projektu. Na tento účel bol zvolený mikropočítač **Raspberry Pi 3** [4], na ktorom bude bežať systém **WordPress** [5]. WordPress bude použitý ako systém na správu obsahu, ktorý umožní vytváranie a správu dokumentácie a ďalších informácií v blogovom formáte.

5.2 Inštalácia Operačného Systému

Prvým krokom bolo inštalovanie operačného systému na Raspberry Pi. Zvolená bola verzia **Raspberry Pi OS Lite** [6], ktorá neobsahuje grafické užívateľské rozhranie, čo je ideálne pre náš projekt, keďže pre správu servera sa využíva SSH. Na inštaláciu operačného systému bola použitá aplikácia **Raspberry Pi Imager** [7].



Obr. 6: Inštalácia Raspberry OS Lite

5.3 Inštalácia Potrebných Komponentov

Apache Webový Server [8] : Na inštaláciu webového servera bol použitý Apache, populárny open-source webový server, ktorý umožňuje hostovať WordPress. Inštalácia prebieha príkazom

```
sudo apt-get install apache2 -y
```

Výpis 1: Inštalácia Apache webového servera

Po inštalácii Apache bol Raspberry Pi pripravený na hostovanie WordPress stránky.

PHP Modul [9] : PHP je skriptovací jazyk, ktorý WordPress používa na generovanie dynamického obsahu webovej stránky. Na inštaláciu PHP sa použil príkaz:

```
sudo apt-get install php -y
```

Výpis 2: Inštalácia PHP Modulu

PHP tiež umožňuje komunikáciu medzi WordPressom a databázou.

Databázový Server MariaDB [10] : Na správu databázy pre WordPress bol použitý MariaDB, open-source databázový server. Po inštalácii bolo potrebné vytvoriť databázu pre WordPress a priradiť potrebné privilegium pre prístup. Inštalácia prebiehala nasledovne:

```
sudo apt-get install mariadb-server php-mysql -y  
sudo mysql_secure_installation  
sudo mysql -uroot -p
```

Výpis 3: Inštalácia MariaDB a PHP modulov

```
CREATE wordpress DATABASE;  
GRANT ALL PRIVILEGES ON wordpress.* TO 'root'@'localhost' IDENTIFIED BY 'password';  
FLUSH PRIVILEGES;
```

Výpis 4: Nastavenie databázy pre WordPress

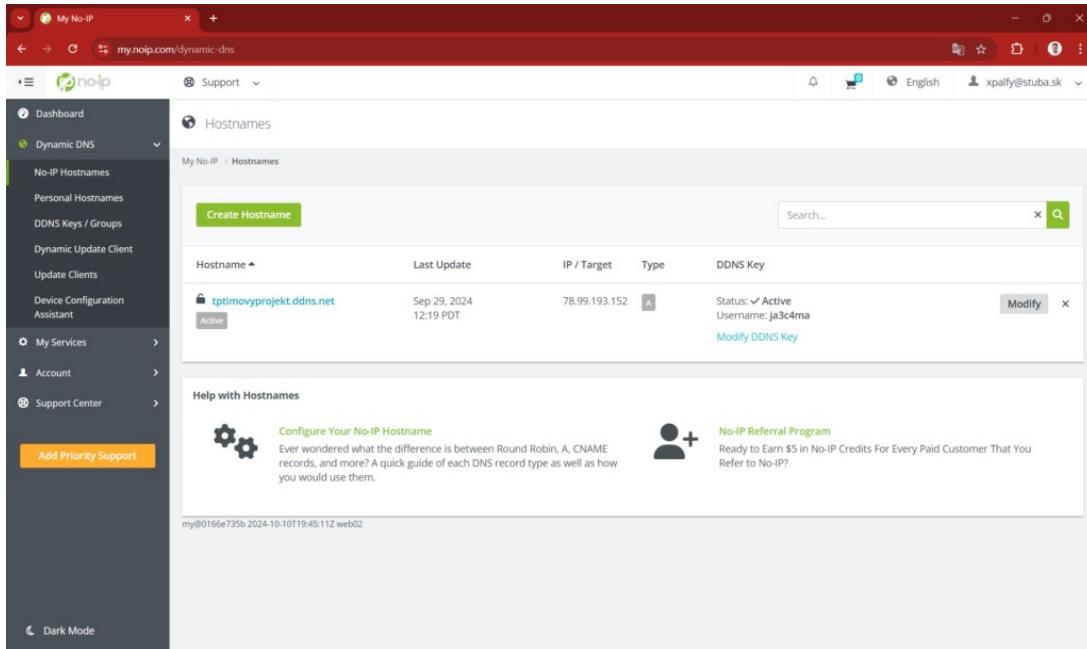
Inštalácia a Konfigurácia WordPress: Po inštalovaní Apache, PHP a MariaDB bol ďalším krokom inštalovať najnovšiu verziu WordPressu. Tento proces zahŕňal stiahnutie a rozbalenie súborov do adresára webového servera. Inštalácia prebiehala príkazmi:

```
sudo wget http://wordpress.org/latest.tar.gz
sudor tar xzf latest.tar.gz
sudo mv wordpress/* .
sudo chown -R www-data: .
```

Výpis 5: Stiahnutie a príprava WordPress súborov na serveri

Po dokončení inštalácie bolo možné pristúpiť k inicializácii WordPressu cez webový prehliadač zadáním IP adresy Raspberry Pi.

Konfigurácia Domény a HTTPS: Webová stránka bola spočiatku dostupná len v miestnej sieti. Na to, aby bola prístupná cez internet, bolo potrebné nastaviť presmerovanie portov na smerovači (porty 80 a 443 pre HTTP a HTTPS). Pomocou služby NO-IP.com bola pridelená bezplatná doména, ktorá bola spojená so službou, čím bola stránka dostupná aj mimo lokálnej siete.



Obr. 7: Nastavenie hostiteľského mena a cielovej IP adresy na webovej stránke NO-IP.

Na zabezpečenie pripojenia cez HTTPS bol použitý nástroj Certbot, ktorý umožnil získať SSL certifikáty pre šifrovanie pripojenia.

Po dokončení týchto krovok bola webová stránka plne funkčná a prístupná cez internet s podporou HTTPS. Stránka bude slúžiť ako platforma pre dokumentáciu projektu, kde budú zaznamenané všetky kroky a pokroky pri vývoji nástroja na dešifrovanie historických dokumentov. Odkaz na webovú stránku: tptimovuprojekt.ddns.net

6 Implementácia

6.1 Hlavná stránka projektu

Hlavná stránka webovej aplikácie je úvodným bodom pre používateľov, ktorí chcú využívať naše nástroje na **prácu so stredovekými textmi**. Stránka je navrhnutá prehľadne a intuitívne, aby umožnila jednoduchú orientáciu.

Ked' používateľ navštívi webovú stránku prvýkrát, dostane notifikáciu o tom, že naša stránka používa **cookies**.

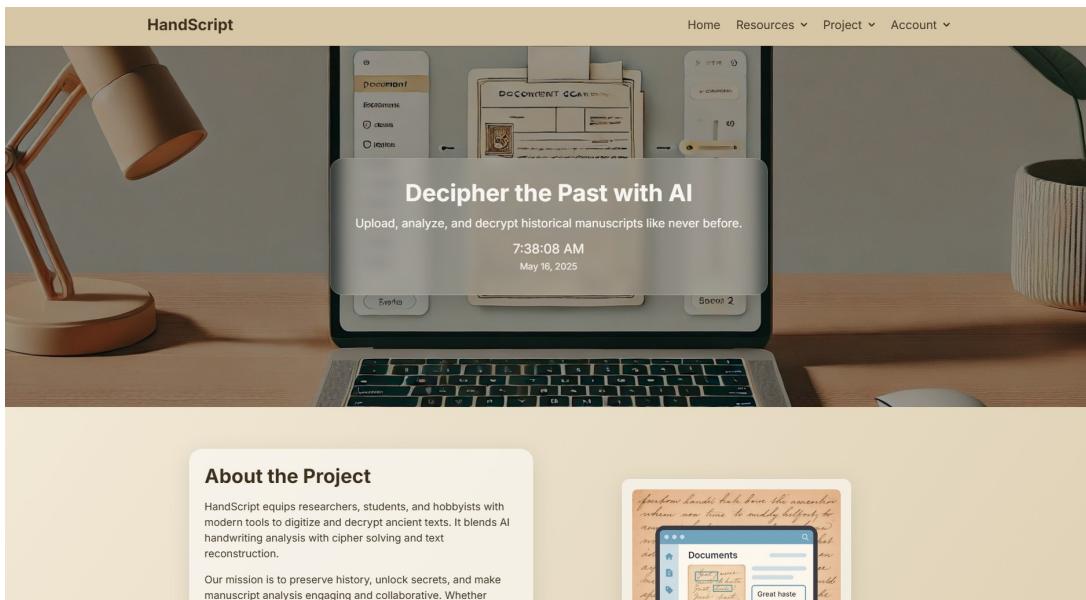
V hornej časti stránky sa nachádza navigačný panel, ktorý obsahuje odkazy na podstránky s **detajlmi o projekte**, o **členoch tímu** a o **často kladených otázok**. Tieto sekcie poskytujú používateľom prehľad o cieľoch projektu, jeho vývoji a ľudoch, ktorí sa na ňom podielali.

Súčasťou navigačného panela sú aj odkazy na **registráciu** a **prihlásenie**. Používateľ sa tak môže jednoducho zaregistrovať a vytvoriť si účet alebo sa prihlásiť do existujúceho účtu.

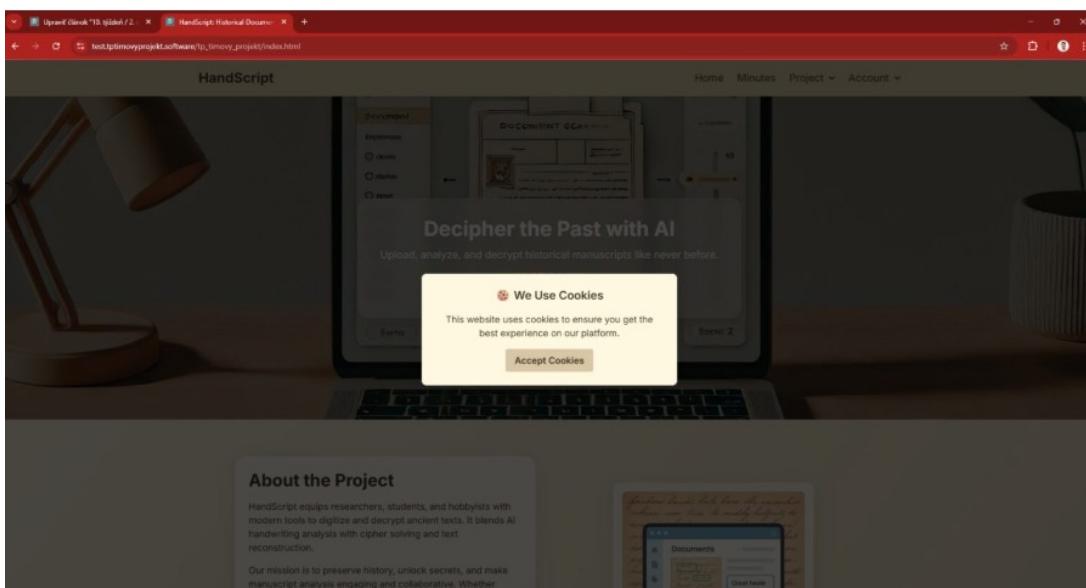
Pod navigačným panelom sa nachádza hlavná sekcia stránky, kde sú uvedené menšie detaily o projekte a jeho využití. V tejto časti sa nachádzajú aj digitálne hodiny, ktoré zobrazujú aktuálny čas v reálnom čase a dodávajú stránke dynamický prvok.

Na konci sú opäťovne umiestnené odkazy na registráciu a prihlásenie, aby boli tieto možnosti ľahko dostupné aj po prechode na spodok stránky.

Na boku webovej stránky sa nachádza pätička, kde môžeme nájsť odkazy na stránku projektu a na stránku s najčastejšími otázkami (FAQ).



Obr. 8: Hlavná webová stránka projektu



Obr. 9: Notifikácia o použití cookies

6.1.1 Podstránka projektových detajov

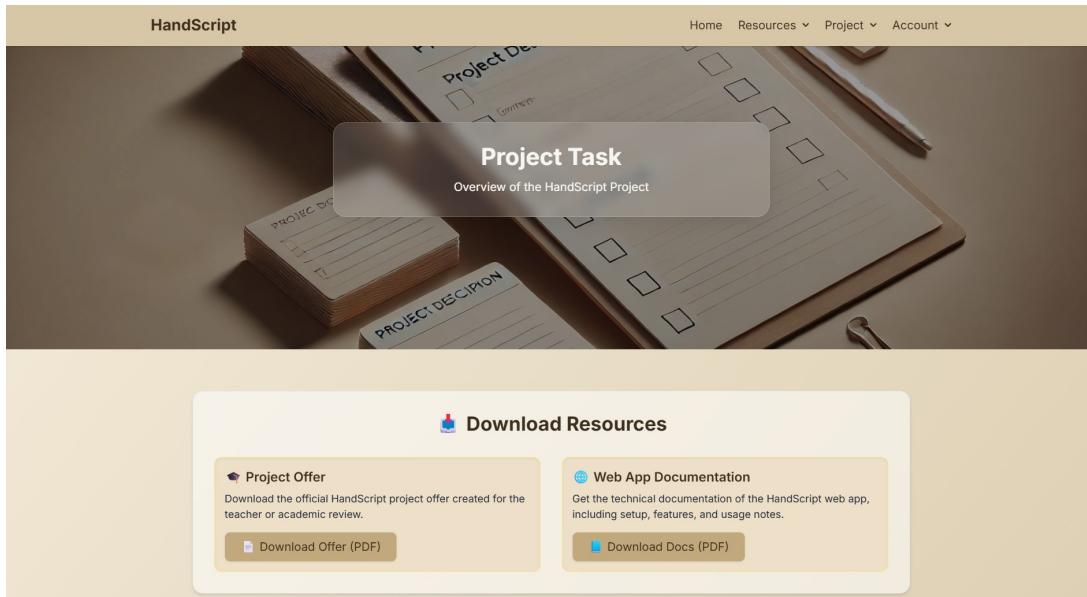
Podstránka s detailmi projektu poskytuje návštěvníkom podrobný prehľad o projekte a jeho cieľoch.

V úvode stránky sa nachádzajú odkazy na stiahnutie dôležitých dokumentov, konkrétnie **ponuky projektu vo formáte PDF** a **dokumentácia**, ktoré popisujú technické a procesné aspekty vývoja. Tieto súbory sú dostupné pre všetkých návštěvníkov a slúžia ako oficiálny sprievodca projektom.

Po tejto úvodnej časti nasleduje **textové vysvetlenie samotného projektu**, kde je podrobne opísaná téma, ktorej sa projekt venuje. Táto časť objasňuje hlavný cieľ projektu, jeho význam a prístup, ktorý bol použitý pri jeho realizácii.

V ďalšej sekcií sú uvedené všetky **programovacie jazyky**, **použitý softvér** a prípadne aj **hardvérové nároky**, ktoré sú potrebné na úspešnú implementáciu. Tento prehľad slúži najmä na lepšie pochopenie technického zázemia a rozsahu riešenia.

Celá podstránka je navrhnutá prehľadne, aby návštěvník rýchlo získal prístup k dokumentom a zároveň sa mohol podrobne oboznámiť s obsahom a cieľom projektu.



Obr. 10: Podstránka s detajlm projektu

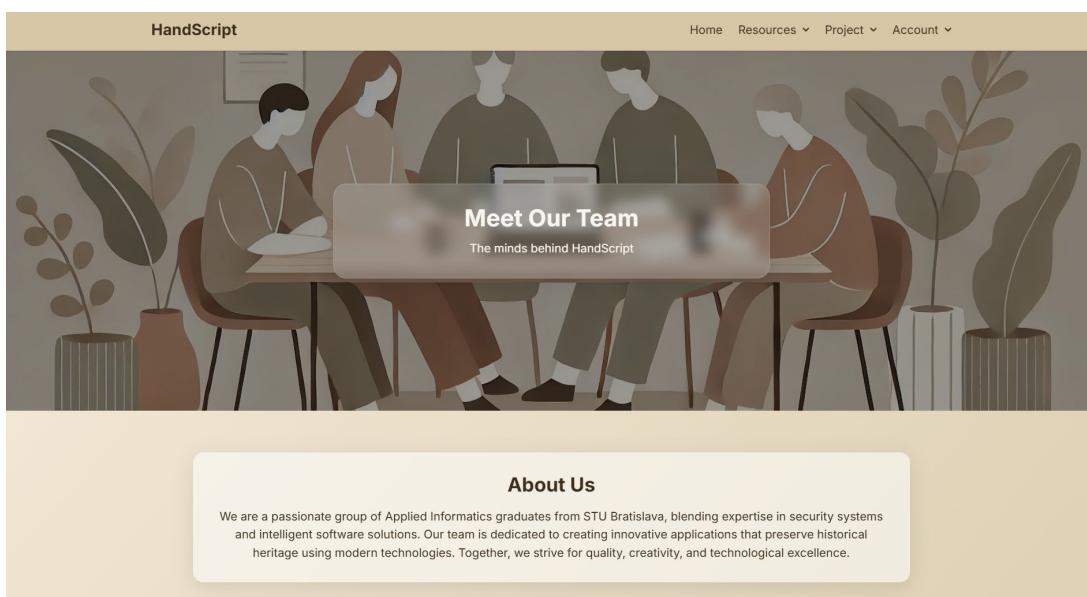
6.1.2 Podstránka tímu

Podstránka tímu predstavuje ľudí, ktorí **stoja za projektom**. V úvode je sekcia, kde je v niekoľkých vetách stručne opísané, kto sme, aké sú naše hodnoty a čo nás motivovalo pracovať na tomto projekte. Text je formulovaný tak, aby pôsobil priateľsky a zároveň profesionálne.

Nasleduje časť, kde sú vymenované **hlavné oblasti odbornosti** členov tímu. Táto sekcia poukazuje na technické, výskumné a iné relevantné schopnosti, ktoré prispeli k realizácii projektu.

V závere podstránky je vyzdvihnutý **projektový vedúci**. Táto časť zdôrazňuje, kto bol hlavným koordinátorom a hnacou silou tímu.

Celá podstránka je navrhnutá tak, aby návštěvníkovi poskytla rýchly, ale komplexný pohľad na tím stojaci za projektom.



Obr. 11: Podstránka tímu

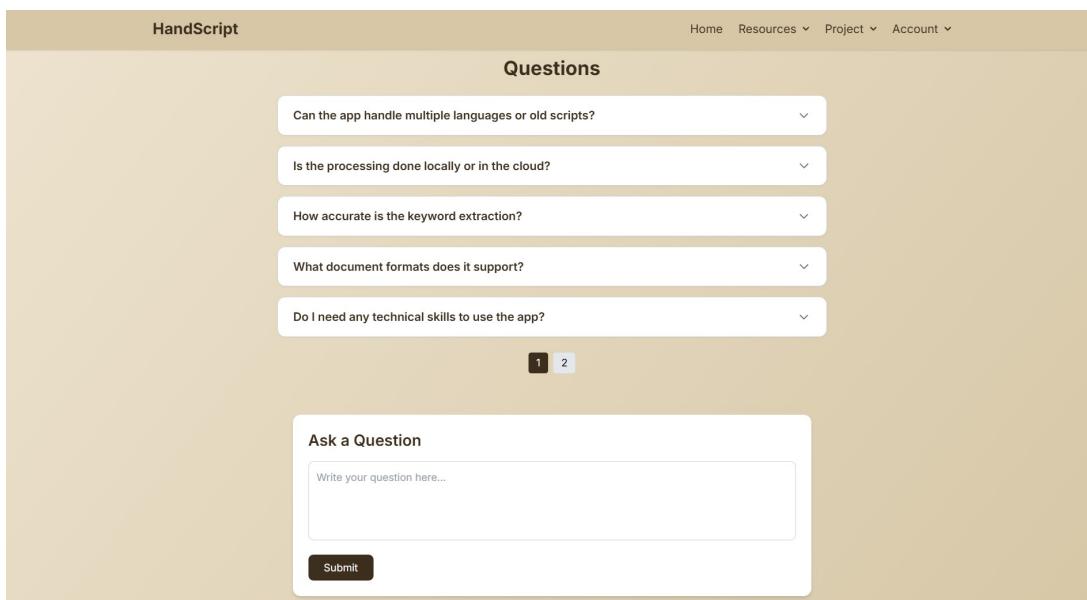
6.1.3 Podstránka FAQ

Podstránka FAQ je určená na to, aby používateľom poskytla **rýchle odpovede a pomoc pri práci** s webovou aplikáciou. Hned na začiatku stránky sa nachádza **videonávod**, ktorý názorne vysvetluje, ako používať jednotlivé funkcie aplikácie. Video slúži ako praktický sprievodca pre nových aj existujúcich používateľov a pomáha im rýchlo sa zorientovať.

Pod videom sa nachádza sekcia **často kladených otázok**, kde sú zoradené najbežnejšie otázky používateľov spolu s odpoveďami. Tieto otázky pokrývajú rôzne témy – od **technických problémov**, cez **registráciu**, až po fungovanie samotného systému na **dešifrovanie textov**.

Potom máme **interaktívne pole** na zadávanie otázok, kde môžu návštěvníci napísat svoju vlastnú otázku, ak v zozname nenájdu odpoveď na to, čo hľadajú. Tento formulár umožňuje jednoduché a priame spojenie s tímom podpory.

Cieľom tejto stránky je poskytnúť používateľovi podporu jednoducho, rýchlo a efektívne, bez potreby kontaktovať technickú podporu pri každom probléme.



Obr. 12: Podstránka často kladených otázok

6.1.4 Podstránka swagger API

Tento projekt používa Swagger (špecifikáciu OpenAPI) na dokumentáciu a interakciu s RESTful API. Swagger poskytuje štandardizovaný spôsob popisu API endpointov, ich vstupných parametrov, výstupov a očakávaných chýb, čím vývojárom uľahčuje testovanie a integráciu s backendom.

Na integráciu Swaggeru do našej Flask aplikácie využívame knižnicu Flasgger, ktorá umožňuje definovať endpointy priamo v docstringoch v kóde. Výhodou je automaticky generované interaktívne rozhranie Swagger UI, ktoré slúži ako dokumentácia aj testovací nástroj.

Endpointy sú prehľadne rozdelené podľa RESTovskej štruktúry do modulov:

- **Modules**

- POST `/crop_polygon` – Oreže polygonálnu oblasť z obrázka.
- POST `/modules/classify` – Vykoná klasifikáciu dokumentu na základe jeho obsahu.
- POST `/modules/decrypt_cipher_with_key` – Dešifruje dokument pomocou iného dokumentu ako kľúča.
- POST `/modules/encode_letters` – Zakóduje znaky dokumentu.
- POST `/modules/get_cipher_json` – Získa zakódované údaje vo formáte JSON.
- POST `/modules/get_processing_result_status` – Získa stav spracovania pre daný dokument.
- POST `/modules/segmentate_page` – Segmentuje celú stránku dokumentu.
- POST `/modules/segmentate_sections` – Rozdelí dokument do sekcií.

- **Zdieľanie dokumentov**

- POST `/documents/add_shared_user` – Pridá zdieľaného používateľa k dokumentu.
- POST `/documents/remove_shared_user` – Odstráni zdieľaného používateľa z dokumentu.

- **Správa dokumentov**
 - POST `/documents/get_document` – Načíta detailné údaje o dokumente vrátane metadát a obrázkov.
 - POST `/documents/get_documents_by_user_and_status` – Získava dokumenty používateľa podľa stavu (napr. návrh, publikovaný).
 - POST `/documents/get_items_by_doc_and_status` – Získava položky dokumentu podľa stavu a typu.
 - DELETE `/documents/delete_document` – Vymaže dokument z databázy a súborového systému.
 - POST `/documents/get_json` – Načíta uložené JSON dátá dokumentu.
 - POST `/documents/save_json` – Uloží upravený obsah JSON dokumentu.
 - POST `/documents/save_processing_result` – Uloží výsledok spracovania dokumentovej položky (napr. segmentácia, klasifikácia).
 - POST `/documents/update_doc_public` – Nastaví dokument ako verejný alebo súkromný.
 - POST `/documents/update_document_title` – Zmení názov dokumentu s kontrolou duplicit.

- **Používateľia**

- DELETE `/users/delete_user` – Vymaže používateľa a všetky jeho dokumenty.

- **Segmentácia**

- POST `/segmentate_text` – Vykoná detailnú segmentáciu textu (napr. riadky, bloky).

6.2 Registrácia/Prihlásenie

Prvým krokom implementácie bolo vytvorenie **databázu** pre projekt. V databáze sme pomocou SQL príkazov navrhli tabuľku s názvom **users**, ktorá je určená na uchovávanie údajov o **registrovaných používateľoch**. Táto tabuľka obsahuje atribúty ako **ID**, **používateľské meno**, **e-mail** a **heslo**. Na zabezpečenie integrity údajov a predchádzanie duplikáciám sme stanovili, že identifikátory, používateľské mená a e-maily musia byť **jedinečné**. SQL príkaz použitý na vytvorenie tabuľky bol nasledovný:

```
CREATE TABLE users
(
    id          INT AUTO_INCREMENT PRIMARY KEY,
    username    VARCHAR(255) NOT NULL UNIQUE,
    email       VARCHAR(255) NOT NULL UNIQUE,
    password    VARCHAR(255) NOT NULL,
    is_verified TINYINT(1)   DEFAULT 0
);
```

Výpis 6: SQL príkaz na vytvorenie tabuľky **users** pre uchovávanie údajov o registrovaných používateľoch.

Na podstránke registrácie si používateľ zadáva svoje **používateľské meno, e-mail a heslo**. Z bezpečnostných dôvodov je heslo vyžadované **dvakrát**, aby sa minimalizovala pravdepodobnosť preklepov a zabezpečilo, že používateľ zadal správne údaje. V databáze sú ukladané **šifrované heslá**, čo zaručuje vyššiu úroveň ochrany údajov.

The screenshot shows a web browser window for the 'HandScript' application. At the top, there is a navigation bar with links for Home, Project, Minutes, Login, and Register. The main content area has a light beige background. A central modal dialog box is titled 'Create an Account'. It contains four input fields: 'Username' (containing 'xtukor'), 'Email' (containing 'tukoraron@gmail.com'), 'Password' (containing '*****'), and 'Confirm Password' (containing '*****'). Below these fields is a brown 'Register' button. At the bottom of the page, there is a small copyright notice: '© 2025 HandScript — [Visit Project Page](#)'.

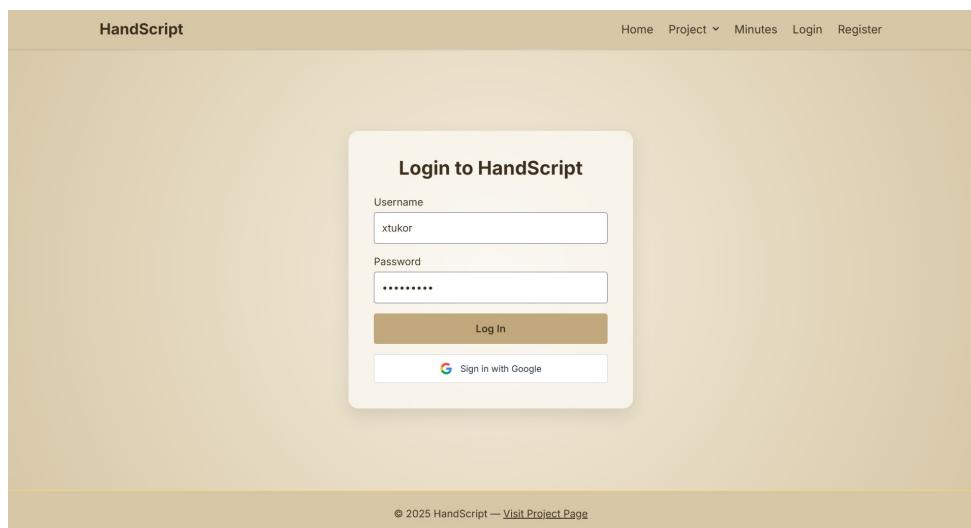
Obr. 13: Registračná stránka webovej aplikácie

Ked používateľ vyplní registračný formulár, na obrazovke sa zobrazí druhý formulár. Pred prvým prihlásením bude používateľ požiadany o **overenie svojej e-mailovej adresy**. Systém pošle šestciferný kód, ktorý musí používateľ zadat do formulára. Ak sa kód **zhoduje**, adresa je **overená**.

The screenshot shows a dark-themed version of the 'HandScript' application. The registration form is displayed in a light-colored modal dialog box. The 'Email Verification' section asks the user to enter a code sent to their email. The input field contains '770875' and below it is a brown 'Verify' button. Other fields for 'Username' ('xtukor') and 'Password' ('*****') are visible at the bottom of the dialog. The footer of the page includes the copyright notice: '© 2025 HandScript — [Visit Project Page](#)'.

Obr. 14: Formulár pre verifikáciu e-mailovej adresy

Na podstránke prihlásenia sa používateľ môže po úspešnej verifikácii e-mailovej adresy **prihlásiť** prostredníctvom prihlasovacieho formulára. Systém kontroluje **správnosť údajov**, pričom v prípade chyby, ako je nesprávne heslo, už existujúce používateľské meno alebo nesprávny formát údajov, používateľ dostane chybové hlásenie. Na zobrazovanie týchto hlásení sme použili knižnicu **TOAST**, ktorá ponúka prehľadné a moderné notifikácie. Ak je prihlásenie úspešné, používateľ je automaticky presmerovaný na stránku určenú iba pre overených používateľov. Neoverení používatelia na túto stránku **nemajú prístup**, čo zabezpečuje bezpečnosť a súkromie.



Obr. 15: Prihlasovacia stránka webovej aplikácie

Dizajn webovej aplikácie bol navrhnutý s využitím frameworkov jQuery a Bootstrap, ktoré zabezpečili responzívne a moderné používateľské rozhranie. Tieto nástroje umožnili vytvoriť intuitívny a esteticky príjemný vzhľad stránky, čím sa zlepšila celková používateľská skúsenosť. Týmto spôsobom sme úspešne implementovali základné funkcie registrácie a prihlásenia, čím sme položili pevné základy pre ďalší vývoj aplikácie.

6.2.1 Implementácia Bearer Token v Aplikácii

Ako súčasť zabezpečenia našej aplikácie sme implementovali systém autentifikácie založený na ***JSON Web Token (JWT)*** [11]. Tento mechanizmus využíva ***Bearer Token*** [12], ktorý umožňuje bezpečné overovanie používateľov bez potreby opakovanej autentifikácie. Pri úspešnom prihlásení server vygeneruje token obsahujúci základné identifikačné údaje používateľa, ako je jeho **ID** alebo **meno**, doplnkové informácie ako **čas vytvorenia** a **čas vypršania**, a podpis šifrovaný algoritmom **HS256** pomocou **tajného kľúča**. Token je následne odoslaný klientovi, ktorý ho uchováva vo svojom úložisku, napríklad v **LocalStorage** alebo **Cookies**.

Token sa používa pri každej ďalšej požiadavke klienta na server. Klient ho zahrnie do hlavičky HTTP požiadavky v tvare **Authorization: Bearer <token>**. Server po prijatí požiadavky token **overí** – skontroluje jeho **platnosť**, **podpis** a **správnosť údajov**. Na tento účel sme využili knižnicu ***Firebase PHP-JWT*** [13], ktorú sme nainštalovali pomocou príkazu `composer require firebase/php-jwt`. Ak je token platný, server **povolí prístup** k chráneným zdrojom a spracuje požiadavku. Ak je token **neplatný**, **expirovaný** alebo **chýba**, server odpovie chybou **401 Unauthorized**, čím sa zabezpečí, že k citlivým dátam majú prístup len overení používateľia.

Tento systém výrazne zvyšuje **bezpečnosť aplikácie**, pretože eliminuje potrebu opakovane odosielat prihlásovacie údaje a minimalizuje riziko neoprávneného prístupu. V prípade potreby môžeme tokeny obohatiť o ďalšie informácie, ktoré môžu byť užitočné pri **autorizácii**, ako napríklad **používateľské role** alebo **špecifické oprávnenia**.

Implementáciou **Bearer Token** sme zabezpečili **moderný a efektívny spôsob autentifikácie a autorizácie používateľov**. Tento krok dopĺňa už existujúce funkcie **registrácie a prihlásovania**, čím aplikácia splňa **vysoké štandardy bezpečnosti** a poskytuje používateľom **pohodlný a bezpečný prístup** k chráneným zdrojom. Týmto spôsobom sme položili základy pre ďalší vývoj aplikácie, kde bude jednoduché rozšíriť tento systém o **po-kročilé funkcie**, ako je **expirácia relácií** alebo podpora **viacfaktorovej autentifikácie**.

6.2.2 Implementácia prihlásenia cez Google účet

Ako súčasť zjednodušenia autentifikácie a zvýšenia bezpečnosti našej aplikácie sme implementovali možnosť **prihlásenia používateľov cez Google účet** pomocou protokolu **OAuth 2.0** [14]. Tento mechanizmus umožňuje používateľom **rýchle a bezpečné prihlásenie** bez potreby vytvárania nového hesla, čím znižuje **riziko neoprávneného prístupu** a zlepšuje **celkovú používateľskú skúsenosť**.

Pri pokuse o prihlásenie je používateľ **presmerovaný na prihlásovaciu stránku Google**, kde sa overí jeho identita. Po úspešnom prihlásení Google vygeneruje **OAuth 2.0 Access Token**, ktorý obsahuje informácie o používateľovi, ako napríklad:

- **Unikátne ID používateľa** (sub – subject),
- **E-mailovú adresu** (email),
- **Meno a priezvisko** (name),

Tento token je následne **odoslaný do našej aplikácie**, kde ho server **overí pomocou verejného klíča Google** a skontroluje jeho platnosť.

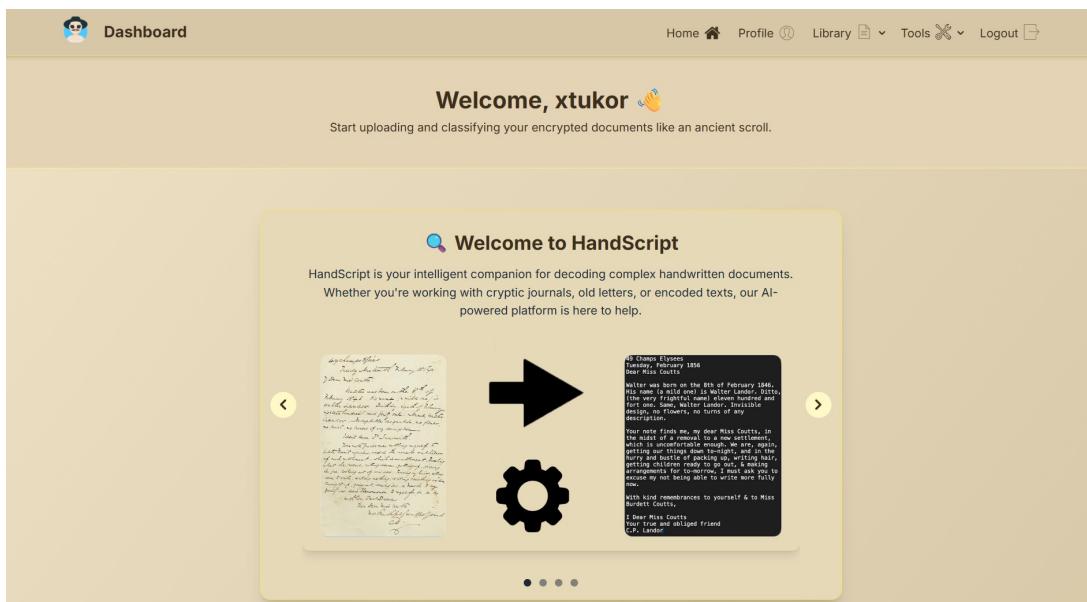
Po úspešnom overení tokenu sa **údaje používateľa uložia do databázy** (ak ešte neexistujú) a server vygeneruje **vlastný JWT token** pre ďalšiu komunikáciu s aplikáciou. Tento token sa následne **odosielá klientovi**, ktorý ho **uchováva v LocalStorage alebo Cookies** a používa ho na **autentifikáciu pri ďalších požiadavkách na server**.

Implementácia prihlásenia cez Google účet dopĺňa existujúce možnosti autentifikácie a poskytuje **moderný a bezpečný spôsob prístupu** do aplikácie. V budúcnosti môžeme tento systém rozšíriť o ďalšie funkcie, ako je **podpora viacfaktorovej autentifikácie** alebo **automatické obnovenie tokenu** pre ešte vyššiu úroveň bezpečnosti a pohodlia.

6.3 Uvítacia stránka po prihlásení

Ak sa používateľ úspešne prihlási, **hlavná stránka** bude obsahovať interaktívne výukové programy. Tu sa používateľ môže oboznámiť s funkciami webovej stránky, ako:

- Základné informácie o aplikácii,
- Informácie o profilu,
- Nahrávanie dokumentov,
- Modely na analýzu obrázkov,
- Ukladanie a zdieľanie dokumentov.



Obr. 16: Uvítacia stránka webovej aplikácie

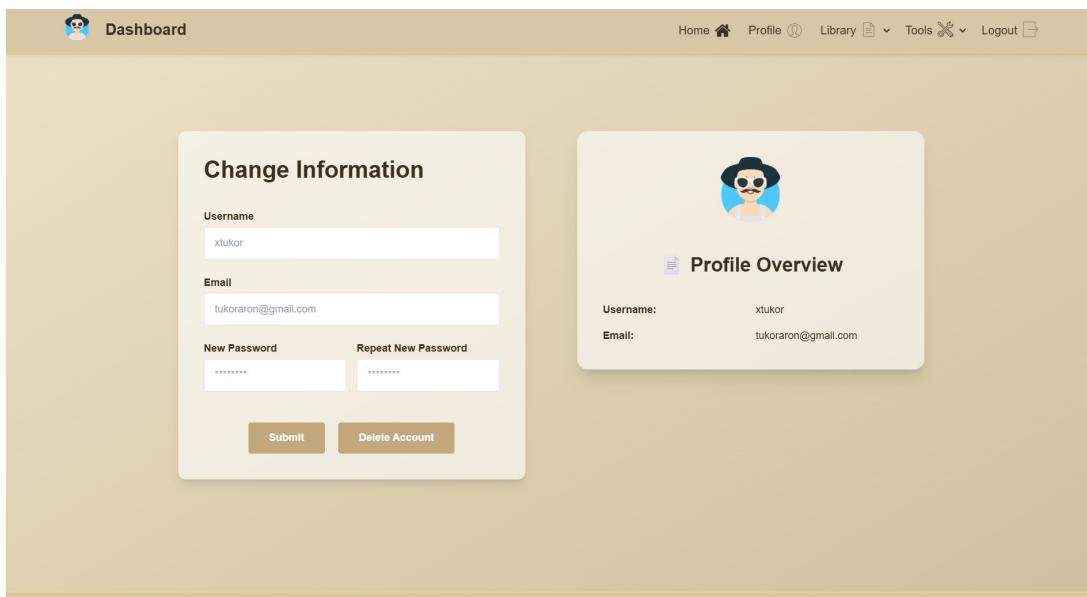
6.4 Podstránka profilu prihláseného používateľa

Podstránka profilu slúži na správu osobného profilu a zobrazenie základných informácií o účte. Stránka je rozdelená na dve hlavné časti – ľavú a pravú.

Na ľavej strane sa nachádza **sekcia určená na úpravu používateľských údajov**. Používateľ tu môže zmeniť svoje **meno**, **e-mailovú adresu** alebo **heslo**. Súčasťou tejto časti je aj možnosť **vymazať používateľský účet**, pričom táto funkcia je zretelne označená a vyžaduje potvrdenie pred vykonaním zmazania, aby sa predišlo náhodnej strate údajov.

Pravá strana slúži **na zobrazenie používateľských informácií** v prehľadnej forme. Používateľ tu vidí svoj **avatar**, spolu so základnými údajmi ako **meno**, **e-mail**, **ID avatara** a **používateľské ID**. Táto časť poskytuje rýchly náhľad na profil bez potreby preklikávania do ďalších nastavení.

Celá podstránka je navrhnutá s dôrazom na jednoduchosť a prehľadnosť, aby si používateľ mohol ľahko spravovať svoj účet a mať všetky dôležité informácie na dosah.

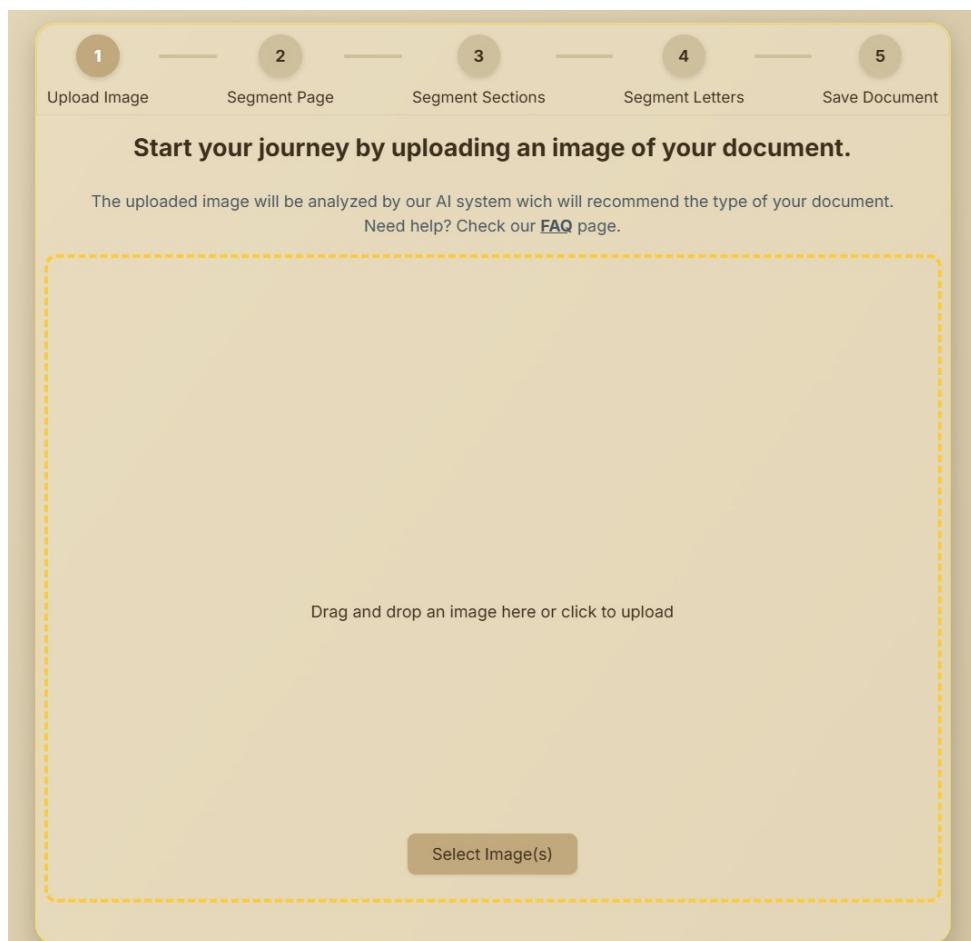


Obr. 17: Podstránka profilu prihláseného používateľa

6.5 Implementácia funkcie načítania obrázkov

Ako súčasť rozšírenia funkcionality našej aplikácie sme úspešne implementovali systém na **nahrávanie obrázkov**. Táto funkcionalita je prístupná iba pre overených používateľov, čím sa zabezpečuje ochrana dát a zabráňuje neoprávnenému prístupu. Pred vstupom do tejto časti aplikácie systém overí používateľskú identitu, čím zaručuje **bezpečnosť**.

Používateľské rozhranie bolo navrhnuté s dôrazom na **jednoduchosť** a **efektívnosť**. Na výber obrázkov je možné použiť dve metódy: **pretiahnutie súboru do vyhradeného pola** alebo jeho **výber prostredníctvom správcu súborov**. Po nahraní sa obrázok uloží na server, kde je spracovaný. Namiesto ukladania súboru priamo do databázy sa na server uloží samotný obrázok a v databáze sa zaznamená len **cesta k súboru**. Tento prístup minimalizuje zátaž na databázu a zvyšuje výkon aplikácie.



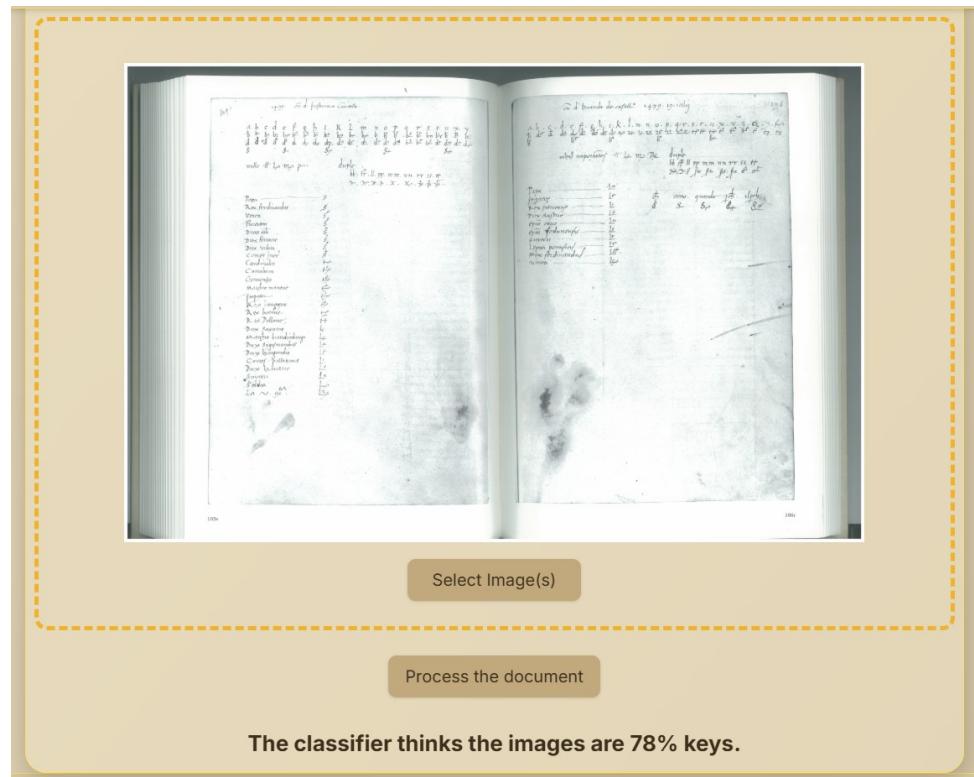
Obr. 18: Funkcionalita nahranie obrázka na hlavnej stránke

Po nahratí súboru a zadaní názvu používateľ vyberie spôsob uloženia súboru. Na pomoc stránka zobrazí používateľovi, čo si myslí o súbore v percentách. Podľa toho, čo je vybrané, vytvorí sa document, ktorý je základom spracovania. Každý document je vytvorený s jedným obrázkom, ktorý je v databáze uložený ako jeden item.



Obr. 19: Nahraný obrázok, pred uložením dokumenta

Po úspešnom uložení súboru sa zobrazí nové tlačidlo na **spustenie spracovania**. Toto tlačidlo **presmeruje** používateľa na samostatné nástroje, ktoré budú predstavené v nasledujúcej kapitole.



Obr. 20: Uložený dokument, s presmerovacím tlačidlom

Týmto sme úspešne dokončili základnú funkciu nahrávania obrázkov, ktorá je dôležitým krokom v našom projekte a vytvára **pevný základ** pre ďalšie rozšírenia.

Tento systém nielenže zlepšuje efektivitu správy súborov tým, že **eliminuje duplicitu**, ale tiež podporuje spoluprácu medzi používateľmi a zvyšuje celkovú flexibilitu aplikácie. Funkcionalita je plne integrovaná do existujúcej infraštruktúry, pričom sú rešpektované všetky pravidlá referenčnej integrity. Zároveň umožňuje jednoduché a bezpečné operácie ako nahrávanie, zdieľanie a správa obrázkov, čím zvyšuje celkovú hodnotu systému pre používateľov.

6.6 Nástroje

Webová aplikácia poskytuje používateľom špecializovanú **sekciu nástrojov**, ktorá umožňuje podrobne spracovanie historických textov v jednotlivých fázach. Používateľ si môže vyhľadať konkrétny document a následne si zvoliť, ktorú časť spracovateľského procesu chce aplikovať. K dispozícii sú tieto nástroje:

- **Segmentácia** – umožňuje výber časti textu pomocou polygonálneho nástroja priamo na dokumente. Táto fáza je základom pre ďalšie kroky spracovania.
- **Analýza** – pokročilé spracovanie segmentovaného textu, ktoré pripraví dátu pre rozpoznávanie alebo klasifikáciu.
- **Rozdelenie na symboly** – rozklad textu na jednotlivé symboly alebo slová, čo je klúčové pre dešifrovanie a následnú transliteráciu.
- **Úprava JSON výstupov** – manuálna alebo poloautomatická editácia výstupných dát vo formáte JSON, ktoré vznikli počas predchádzajúcich krov spracovania.
- **Dešifrovanie textu** - dešifrácia textov pomocou už pracovaných klúčov, t.j. klúče, na ktorých sú už urobené všetky predchádzajúce kroky.

Tieto nástroje sú navrhnuté tak, aby boli použiteľné **samostatne** alebo **v kombinácii**, podľa potrieb používateľa. Všetky sú prístupné cez navigačné menu v pravom hornom rohu webovej aplikácie, čo umožňuje rýchly prístup a plynulú prácu medzi jednotlivými fázami.



Obr. 21: Sekcia nástrojov v navigačnom paneli

6.6.1 Sekcia segmentácie

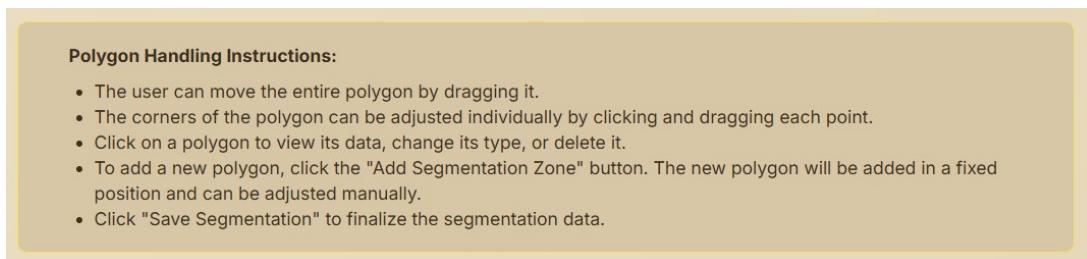
Podstránka segmentácie je súčasťou sady nástrojov, ktoré slúžia na spracovanie textov v jednotlivých fázach. Prvou z týchto fáz je segmentácia textu, teda **vybranie časti dokumentu, ktorá bude ďalej spracovávaná**.

Používateľ má na začiatku možnosť **vyhľadať dokument**, ktorý chce spracovať. Môže tak urobiť **priamo na tejto stránke** alebo sem môže byť **presmerovaný** cez tlačidlo na inej časti aplikácie, ako bolo spomenuté vyššie. Tento spôsob zabezpečuje prepojenosť medzi rôznymi časťami systému a plynulý prechod medzi jednotlivými krokmi spracovania.

Segmentácia samotná prebieha vizuálne – výberom oblasti textu pomocou **polygónov**. Používateľ môže myšou označiť konkrétny úsek textu na dokumente tým, že vytvorí mnohouholník okolo požadovanej časti. Tento výber následne slúži ako **vstup pre ďalšie spracovanie**, napríklad rozpoznávanie znakov alebo analýzu obsahu.

Stránka je navrhnutá s dôrazom na jednoduchosť a interaktivitu, aby proces segmentácie bol pre používateľa čo najintuitívnejší a presný.

Na každej podstránke, kde môže používateľ pracovať s polygónymi, sme umiestnili pomôcku:



Obr. 22: Pomôcka pre používateľov počas segmentácie

1 2 3 4 5

Upload Image Segment Page Segment Sections Segment Letters Save Document

Document position adjustment

The uploaded image has been analyzed, and the document's position is detected. Adjust the boundaries and position of the extracted document to ensure accurate content capture and further processing.

Need help? Click "Show Polygon Help" for detailed instructions or check our [FAQ](#) page.

key2

img4

Show Polygon Help

Add Segmentation Zone Save Segmentation

Obr. 23: Nástroj pre segmentáciu textu

6.6.2 Sekcia analýzy

Na podstránke analýzy sa používateľovi zobrazí dokument **rozdelený na menšie segmenty** podľa výstupu predchádzajúceho modulu. Tieto segmenty sú vizuálne ohraničené polygonálnymi líniami, ktoré znázorňujú, ako aplikácia interpretovala rozmiestnenie textu v dokumente. Tento prehľad umožňuje používateľovi ľahko skontrolovať, ako boli jednotlivé časti textu rozpoznané a spracované.

Každý vyznačený segment môže byť **označený inou farbou**, ktorá reprezentuje **obsah daného úseku**. Môže ísť o **jednotlivé písmená**, **celé slová**, prípadne **číselné údaje**, v závislosti od typu a štruktúry textu. Táto anotácia pomáha pri ďalšom spracovaní údajov a zvyšuje presnosť následných krokov.

Používateľ má navyše možnosť **manuálne pridávať** vlastné polygóny v prípadoch, keď model neoznačil niektoré oblasti správne alebo ich úplne vynechal. Tento zásah umožňuje dodatočné **doplnenie** alebo **opravu chýb**, čím sa zvyšuje flexibilita nástroja.

V súčasnosti nástroj pracuje s pevne definovanými vstupmi, ktoré boli predpripravené, no v budúcnosti je možná integrácia AI modulov v rámci iného projektu.

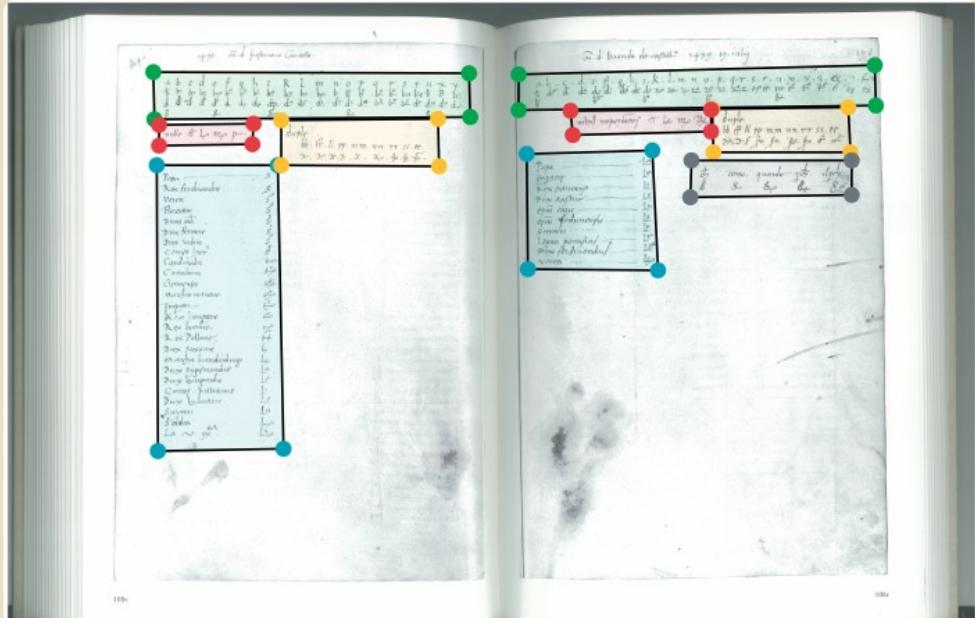
Section position adjustment

On this page, you can review and adjust the detected section boundaries of your uploaded document. Move or resize the highlighted areas to match the actual sections, and change their type if needed. When you are satisfied with the segmentation, save your changes to continue processing the document. Need help? Click "Show Polygon Help" for detailed instructions or check our [FAQ](#) page.

key2

img4

Show Polygon Help



Add Segmentation Zone Save Analysis

Obr. 24: Nástroj pre analýzu dokumentu

6.6.3 Sekcia segmentácia symbolov

Podstránka segmentácie symbolov slúži na **vizuálne rozdelenie textu na jednotlivé znaky/slová**, čo je kľúčový krok pri spracovaní šifrovaných alebo historických dokumentov. Na stránke sú zobrazené znaky klúča alebo šifrovaného textu, ktoré boli automaticky rozdelené a ohraničené **polygonálnymi obdlžníkmi**.

Tieto ohraničenia poskytujú používateľovi prehľad o tom, ako boli jednotlivé písmená **identifikované** a **oddelené**, a tým vytvárajú vizuálnu spätnú väzbu o kvalite segmentácie. Táto presnosť je dôležitá najmä pri ďalšom spracovaní a **dešifrovaní textu**.

Používateľ má zároveň možnosť **manuálne doplniť** alebo **opraviť ohraničenia**, ak niektoré znaky neboli rozpoznané správne alebo boli vynechané. Táto interaktívna možnosť zvyšuje kontrolu nad procesom a umožňuje prispôsobiť výstup konkrétnemu typu dokumentu.

Rovnako ako pri iných nástrojoch, aj v tomto prípade sú vstupy aktuálne pevne definované, avšak v budúcnosti je tu tiež možná integrácia AI modulov v rámci iného projektu.

Detect words and symbols

This tool allows you to detect and segment individual letters, symbols, or words from your uploaded document image. Select a document, then choose a section to work on. You can manually add, move, or adjust polygons to mark each letter or symbol.

Need help? Click "Show Polygon Help" for detailed instructions or check our [FAQ](#) page.

key2

img4

Show Polygon Help

Add Segmentation Zone

Save Letters

Obr. 25: Nástroj pre segmentáciu symbolov

6.6.4 Sekcia úpravy JSON výstupov

Na podstránke **úprav** môže používateľ uložiť výstup klúča alebo šifrovaného textu vo formáte **JSON**. Tento výstup predstavuje výsledok predchádzajúcich spracovateľských krokov a obsahuje všetky relevantné informácie o **rozdelení textu, anotáciách a identifikovaných znakoch**.

JSON výstup je v súčasnosti pevne definovaný podľa vopred stanoveného formátu. V budúcnosti sa však existuje možnosť jeho **automatizovaného generovania** pomocou **AI modulu**, čím by sa zjednodušila práca používateľa a zvýšila efektivita spracovania.

Používateľ má dve hlavné možnosti: stiahnuť JSON súbor do svojho zariadenia alebo ho **uložiť priamo v rámci webovej aplikácie**, kde môže byť ďalej spravovaný a upravovaný.

Okrem toho má používateľ k dispozícii aj **manuálnu úpravu** JSON výstupu. Táto možnosť je dôležitá najmä v prípadoch, keď výstup neobsahuje **všetky potrebné dátá**, alebo ak si používateľ praje prispôsobiť výsledok **podľa svojich požiadaviek**.

Na podstránke sme pridali pomôcku k spracovaniu JSON súborov:



Obr. 26: Pomôcka pre úpravu JSON výstupu

Edit Document JSON

Edit and validate the JSON structure of your selected document. Use the editor below to modify the document's JSON data. Make sure your changes follow proper JSON syntax. When finished, you can save your work or download the updated JSON file.

Need help? Click "Show Json Help" for detailed instructions or check our [FAQ](#) page.

key2

img4

Show Json Help

```
{  
  "polygons": [  
    [  
      {  
        "x": "16.24014705882353",  
        "y": "18.489545454545453"  
      },  
      {  
        "x": "16.24014705882353",  
        "y": "19.606363636363636"  
      },  
      {  
        "x": "29.381029411764708",  
        "y": "19.606363636363636"  
      },  
      {  
        "x": "29.381029411764708",  
        "y": "18.489545454545453"  
      },  
      {  
        "type": "word"  
      }  
    ]  
  ]  
}
```

Download JSON Save Work

Obr. 27: Nástroj pre úpravu a uloženia JSON výstupu

6.6.5 Sekcia dešifrovania

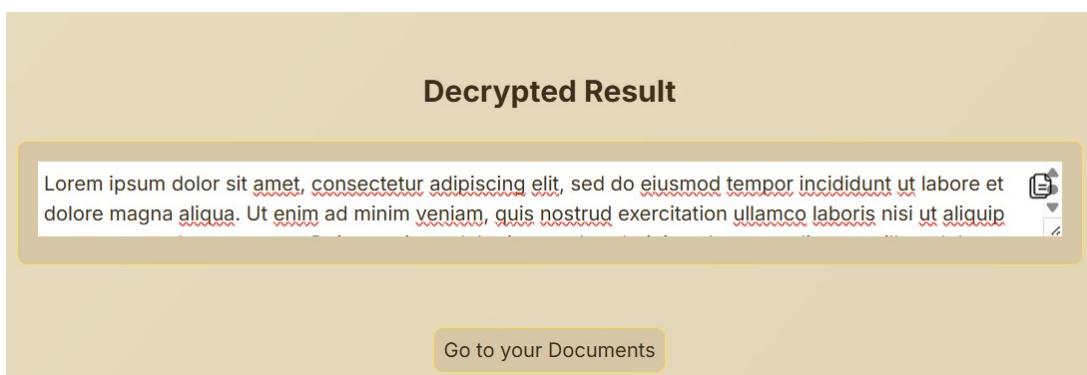
Modul na dešifrovanie predstavuje jednu z klúčových súčasťí webovej aplikácie, ktorá umožňuje používateľovi **rozlúštiť spracovaný šifrovaný text**. Používateľ si môže zvoliť jeden zo svojich dokumentov, ktoré už **prešli predchádzajúcimi fázami spracovania**, a použiť ho ako vstup pre tento nástroj.

Po výbere dokumentu je používateľovi automaticky zobrazený návrh **odporúčaných klúčov**, ktoré môžu slúžiť na dešifrovanie textu. Tieto klúče sú už uložené v databáze pomocou predchádzajúcich nástrojov.

Rovnako ako pri iných nástrojoch, aj na túto stránku môže byť používateľ **presmerovaný** automaticky z predchádzajúcich fáz spracovania dokumentu alebo môže prejsť priamo **cez navigačné menu** aplikácie.

Modul je navrhnutý tak, aby bol intuitívny, no zároveň flexibilný a pripravený na rozšírenia, ako napríklad automatické rozpoznávanie typu šifry alebo AI generovanie klúčov v budúcnosti.

Na konci podstránky je zobrazený Výsledok dešifrovania v textboxe:



Obr. 28: Výsledok dešifrovania

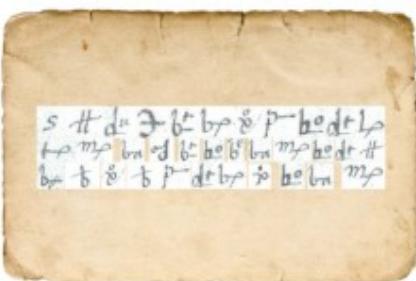
Decrypt your Cipher Documents

This module allows you to decrypt your previously processed Cipher documents using Key documents. To use it, first search for and select the encrypted Cipher document you want to decrypt, then preview the document to confirm your selection. Next, search for and select the appropriate Key document; our AI will recommend the best matching keys for your chosen Cipher document. Once both documents are selected, click the "Start Decipher" button to begin the decryption process. The decrypted result will be displayed below, and you can copy the text to your clipboard or navigate to your documents library.

Note: Only processed Cipher and Key documents will appear in the selectors. If you do not see your files, please upload and process them first.

Need help? Check our [FAQ](#) page.

Please choose the Document you want to decrypt



Please choose the Key you want to use for decryption

Search for a document

Our AI is recommending the following keys for your cipher document:

key3
Match Score: 68%

Select View

A thumbnail image of a document with a green border around the top half. Below the thumbnail, the text "key3" and "Match Score: 68%" is displayed. At the bottom are two buttons: "Select" and "View" followed by an eye icon.

key1
Match Score: 48%

Select View

A thumbnail image of a document with a grey border around the top half. Below the thumbnail, the text "key1" and "Match Score: 48%" is displayed. At the bottom are two buttons: "Select" and "View" followed by an eye icon.

key2
Match Score: 26%

Select View

A thumbnail image of a document with a grey border around the top half. Below the thumbnail, the text "key2" and "Match Score: 26%" is displayed. At the bottom are two buttons: "Select" and "View" followed by an eye icon.

Obr. 29: Nástroj dešifrovania pomocou už predpracovaného klúča

6.7 Knižnica dokumentov

Dokumenty uložené v aplikácii sú prístupné prostredníctvom navigačného panela v hornej časti aplikácie. Po kliknutí na **Documents** sa zobrazí malé rozbalovacie menu s dvoma možnosťami: **dokumenty s klúčmi** a **dokumenty so šifrovacími textmi**, ktoré predstavujú dva hlavné typy súborov, na ktoré sa aplikácia zameriava. Používateľ si môže vybrať, s ktorým typom dokumentu chce pracovať, a potom je používateľ v závislosti od toho, čo si vybral, presmerovaný na jednu z dvoch podstránok.



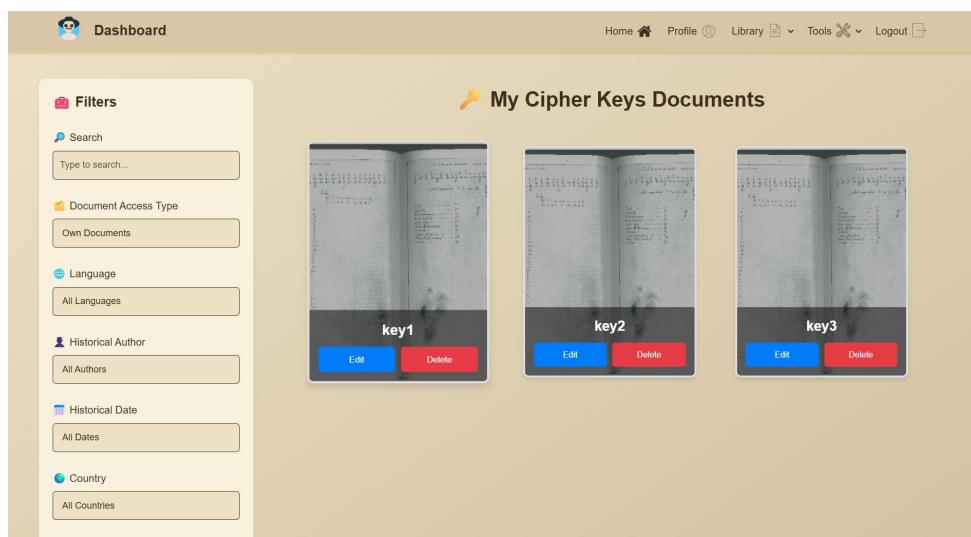
Obr. 30: Podstránky pre jednotlivé typy dokumentov v navigačnom paneli

Obe podstránky majú rovnaký tvar, sú len oddelené, aby používateľ nemusel filtrovať medzi klúčmi a šifrovými textmi:

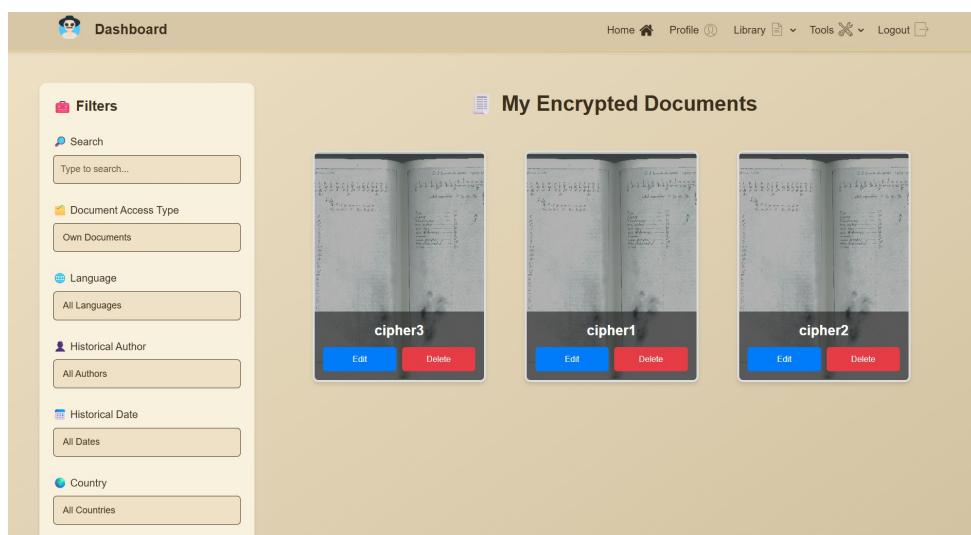
Vyhľadávacia a filtrovacia sekcia sa nachádza v ľavej časti podstránky. Používateľ môže pomocou textového poľa vyhľadať dokument podľa názvu. Súčasťou tejto časti je aj možnosť použiť viaceré filtre naraz – napríklad podľa **jazyka dokumentu**, **autora**, **dátumu vzniku**, alebo **krajiny pôvodu**. Tieto filtre výrazne zjednodušujú orientáciu v databáze a umožňujú rýchle zúženie výsledkov na relevantné položky.

Väčšinu podstránky tvorí samotná **knižnica dokumentov**, kde sú jednotlivé dokumenty zobrazené v prehľadnej mriežke. Každý dokument je prezentovaný ako samostatná karta s názvom, základnými informáciami a možnosťami na **úpravu** alebo **vymazanie**. Táto sekcia poskytuje používateľovi priamy prístup k správe súborov a je navrhnutá s dôrazom na jednoduchosť a prehľadnosť.

V spodnej časti stránky sa nachádza ovládací panel, kde si môže používateľ vybrať, koľko dokumentov sa má zobraziť na jednej strane – **6, 12 alebo 18 záznamov**. Okrem toho je dostupná aj funkcia **stránkovania**, ktorá umožňuje pohodlné prechádzanie medzi jednotlivými stranami databázy dokumentov. Táto kombinácia funkcionalít zabezpečuje, že aj pri väčšom množstve dát ostáva používateľské rozhranie rýchle, intuitívne a dobre ovládateľné.



Obr. 31: Podstránka zobrazujúca uložené klúče



Obr. 32: Podstránka zobrazujúca uložené šifrované texty

6.7.1 Úprava dokumentov

Po kliknutí na tlačidlo **Edit** v sekcií dokumentov sa otvorí podrobne okno úprav, ktoré je rozdelené na **štyri časti**:

- **Náhľad dokumentu** – Zobrazuje obrázok daného dokumentu.
- **Informácie o dokumente** – V tejto časti môže používateľ:
 - **Zobraziť si dátum publikácie,**
 - **Zobraziť si kľúč dešifrovania** - v prípade dokumentu so šifrovacím textom, ktorý bol kompletne spracovaný,
 - **Zobraziť metadáta o dokumente** - jazyk, krajina pôvodu, obdobie vzniku, autor,
 - **Zmeniť názov dokumentu,**
 - **Pridať používateľov s prístupom k dokumentu,**
 - **Sprístupniť dokument verejne** - dokument sa takto stane dostupný pre každého používateľa,
 - **Prezrieť si zoznam používateľov** - používateľia, ktorými je dokument zdieľaný
 - **odobrať prístup konkrétnym používateľom.**
- **JSON výstup** – Obsahuje výstup vo formáte JSON generovaný aplikáciou. Používateľ môže:
 - **upraviť JSON výstup a uložiť zmeny,**
 - **JSON výstup úplne odstrániť,**
 - **pokračovať v spracovaní dokumentu** - ak neboli naňho použitý každý nástroj.
- **Pomôcka pre úpravu JSON** – Táto časť poskytuje vysvetlenie, ako JSON funguje, aby noví používatelia pochopili, ako tento formát môžu upravovať.

Key JSON

```
{
  "classification": [
    18
  ]
}
```

How to Write a JSON Document

- Use double quotes “ for keys and string values
- Keys and values are separated by a colon :
- Separate entries with commas ,
- Wrap your structure in { }
- Use arrays [] for lists

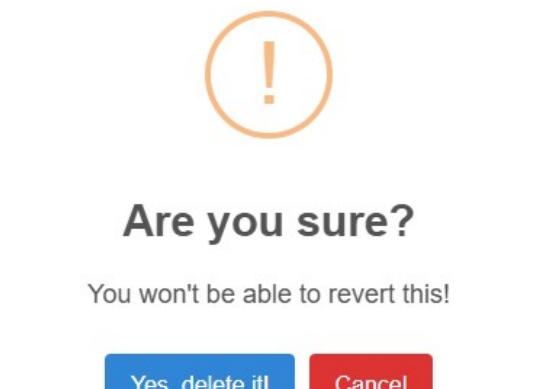
Example 1: Single Object

```
[
  {
    "key": "Batman",
    "code": "BAT-001"
  },
  {
    "key": "Wonder Woman",
    "code": "WW-002"
  }
]
```

Obr. 33: Podstránka na úpravu jednotlivých klúčov

6.7.2 Vymazanie dokumentu

V zobrazení dokumentov má používateľ možnosť dokument vymazať. Po kliknutí na tlačidlo **Delete** sa zobrazí vyskakovacie okno s potvrdením, ktoré upozorní, že tento krok je nezvratný.



Obr. 34: Nahraný obrázok podstránka

6.8 PHP-Python Komunikácia v Webovej Aplikácii

V priebehu posledného týždňa sme úspešne dokončili implementáciu nového backendového riešenia založeného na Flask, ľahkom a výkonnom mikroframeworku pre jazyk Python. Toto riešenie nahradilo pôvodnú architektúru, ktorá trpela problémami s výkonom a obmedzenou škálovateľnosťou.

Voľba Flaska bola motivovaná jeho jednoduchosťou, modulárnosťou a nízkymi nárokmi na systémové prostriedky, čo výrazne zjednodušilo vývoj a správu aplikácie. Flask nám umožňuje efektívne definovať a spracovávať API požiadavky, čím sa výrazne zrýchliла komunikácia medzi klientom a backendom.

Jedným z hlavných benefitov novej architektúry je lepšia kontrola nad spracovaním dát a pripravenosť na budúce rozšírenia systému. Konkrétnie, backend je teraz optimalizovaný pre integráciu modulov umelej inteligencie, ako je napríklad klasifikácia dokumentov, a poskytuje stabilné rozhranie na interakciu s ďalšími spracovateľskými komponentmi.

Celý systém tak získal na spoľahlivosti, rozšíriteľnosti a výkonnosti, čo predstavuje klúčový krok smerom k produkčnému nasadeniu aplikácie.

6.9 Priebeh vývoja

Náš tím používa centralizovaný prístup k vývoju, pri ktorom sa všetka práca vykonáva priamo na vzdialenom počítači spravovanom jedným z našich členov. Tento hostiteľský systém slúži ako vývojové prostredie aj ako produkčný server. Namiesto spúšťania aplikácie lokálne na jednotlivých zariadeniach sa každý vývojár pripája k tomuto hostiteľskému počítaču cez SSH. Najčastejšie používame Visual Studio Code s rozšírením Remote SSH, ktoré nám umožňuje otvárať súbory, používať integrovaný terminál a spúštať alebo ladíť aplikáciu, akoby bežala na našom vlastnom počítači.

Aplikácia je postavená pomocou PHP a MySQL a beží v rámci LAMPP balíka nainštaloванého na hostiteľskom počítači. Tento lokálny serverový balík obsahuje Apache, ktorý spracováva HTTP požiadavky, a MySQL, ktorý obsluhuje databázu na pozadí. Edže vývoj prebieha priamo na hostiteľskom zariadení, nie je potrebné, aby si každý člen tímu nastavoval vlastné lokálne prostredie. Tým sa minimalizujú rozdiely medzi individuálnymi konfiguráciami a zaručuje sa, že všetci pracujú v rovnakom prostredí.

Tento model je obzvlášť účinný pre malé tímy, kde je spolupráca častá a intenzívna. Zdrojový kód je uložený a spravovaný v repozitári na GitHube a každý člen tímu odosiela svoje zmeny do spoločnej vetvy.

6.10 Nasadenie a verejný prístup

Hoci aplikácia beží na lokálnom serveri, je potrebné, aby bola dostupná aj zvonka – napríklad na testovanie, spätnú väzbu alebo živé prezentácie. Na bezpečné sprístupnenie používame Cloudflare Tunnel v kombinácii s Cloudflare Zero Trust. Tunel je riadený ľahkým agentom (cloudflared), ktorý beží na hostiteľskom počítači. Po spustení vytvorí zabezpečené a šifrované spojenie medzi hostiteľom a globálnou sieťou Cloudflare. Vďaka tomu je možné získať verejne prístupnú URL adresu bez potreby otvárania portov na routeri alebo firewalli.

Cloudflare funguje ako reverzný proxy server – presmerováva externú prevádzku na hostiteľa, pričom skrýva jeho IP adresu. Keď je to potrebné, Cloudflare Zero Trust umožňuje nastaviť autentifikačné pravidlá, ktoré zabezpečujú, že aplikáciu môžu cez tunel používať iba autorizovaní používatelia. Táto kombinácia sprístupnenia a bezpečnosti nám umožňuje prevádzkovať aplikáciu na vlastnom serveri bez straty kontroly či bezpečnosti.

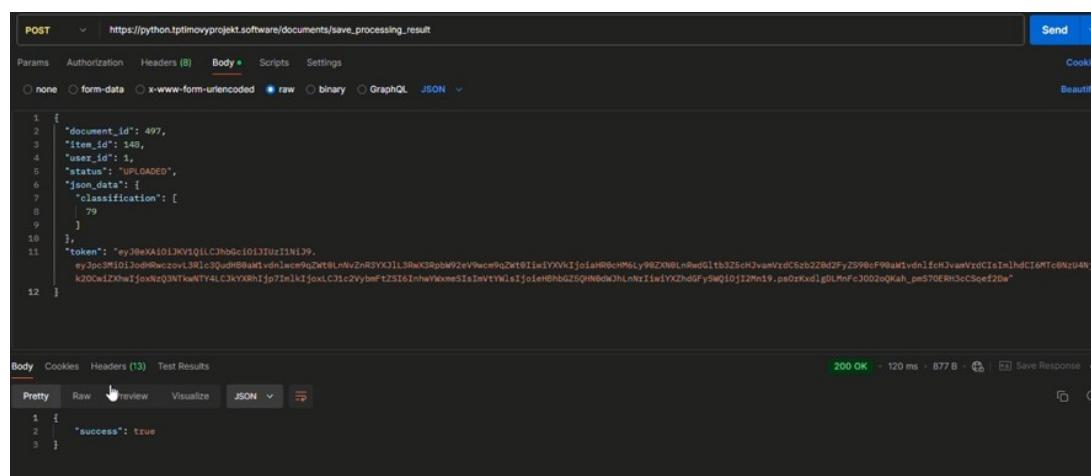
Nasadenie aplikácie je automatizované pomocou GitHub Actions. Kedykoľvek dôjde k zmene v repozitári – typicky v hlavnej vetve – spustí sa GitHub Action. Na hostiteľskom počítači beží proces, ktorý tieto zmeny sleduje. Po ich zistení stiahne najnovšiu verziu kódu z GitHubu, ukončí bežiaci server (ktorý môže byť napríklad vo Flasku alebo Apache), aktualizuje kód a znova spustí aplikáciu. Tento postup umožňuje nepretržité nasadzovanie s minimálnou manuálnou prácou a umožňuje tímu doručovať aktualizácie rýchlo a spoloahlivo.

6.11 Testovanie

V rámci nášho testovacieho procesu používame nástroj Postman na manuálne overovanie správania našich API endpointov. Postman je široko používaný nástroj na vývoj a testovanie API, ktorý poskytuje používateľsky prívetivé rozhranie na vytváranie, odosielanie a analyzovanie HTTP požiadaviek. Vďaka možnosti jednoducho nastavovať hlavičky, telá požiadaviek a parametre nám Postman umožňuje rýchlo a iteratívne testovať bez potreby písania ďalšieho kódu. Je to obzvlášť užitočné v počiatočných fázach vývoja endpointov, keď sa funkcia ešte vyvíja a rýchla spätná väzba je klúčová.

Aktuálne postupujeme tak, že posielame požiadavky na konkrétné endpointy s presne definovanými údajmi (zvyčajne vo formáte JSON), a následne kontrolujeme odpovede z hľadiska správnosti – overujeme HTTP status kódy, štruktúru odpovede a obsah dát. Taktiež pomocou Postmana testujeme rôzne hraničné prípady – ako sú neplatné vstupy alebo pokusy o neoprávnený prístup – aby sme sa uistili, že API zvláda chyby správnym a očakávaným spôsobom.

Manuálne testovanie pomocou Postmana je dôležité, pretože poskytuje istotu predtým, než pristúpime k tvorbe automatizovaných testov alebo integrácií s inými systémami. Umožňuje nám rýchlo odhaliť zjavné problémy, nezávisle overiť obchodnú logiku a lepšie pochopiť správanie systému.



```
POST https://python.tplimovskyprojekt.software/documents/save_processing_result
Params Authorization Headers (8) Body Scripts Settings
 none  form-data  x-www-form-urlencoded  raw  binary  GraphQL  JSON
1 {
2     "document_id": 497,
3     "item_id": 348,
4     "user_id": 1,
5     "status": "UPLOADED",
6     "json_data": [
7         {
8             "classification": [
9                 79
10            ]
11        },
12    ],
13    "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.
ey3jcSM101j3odRwcov13Rlc3Qjd8aaw1vdnlwq96Zwt8LnNvZnR3YXJlLSRMx3Rpbl92eV9cm9qZt0IiwiYXVkJjoiaH0oHM6Ly98ZXN0LnRwdGltb3Z6dH2vamVzdC5zb2Z8d2FyZS90cF90aM1vdnlfcHJvamVzdCisIe1hdCI6TcBNzU4Nj
k20Cc1ZKhwIjoxNzQNTxkwNTY4LC3kYXRhIjp7Ie1k1joxLC31c2VybemtzIiIehBhbdQzQhNbd2hlnNzIiIw1YXZhdiGy5WQ10f2Mn39.ps0rKvd1g0LmFe3002oQkah_pS70ERh3cCSqeF20w"
14 }
```

Body Cookies Headers (13) Test Results

Pretty Raw review Visualize JSON

```
1 {
2     "success": true
3 }
```

200 OK > 120 ms · 877 B Save Response

Obr. 35: Testovanie pomocou Postmanu

7 Zápisnice

7.1 1. semester - 1. týždeň

- **Dátum:** 30.09.2024, 08:00
- **Umiestnenie:** Slovenská technická univerzita v Bratislave, Fakulta elektrotechniky a informatiky
- **Miestnosť:** C614 (blok C, 6. poschodie)
- **Účastníci:**
 - Bc. Bence Both,
 - Bc. Matyas Horváth,
 - Bc. Jozef Nyitrai,
 - Bc. Áron Tükör,
 - Bc. Vincent Pálfy
- **Vedúci:** Ing. Stanislav Marochok

Bod programu č. 1: Predstavenie projektu

Počas konzultácie vedúci tímu podrobne predstavil projekt, ktorého cieľom je vyvinúť algoritmus na **spracovanie historických dokumentov a záznamov** a vytvoriť **webovú stránku**, ktorá umožní **spracovanie a dešifrovanie zašifrovaných dokumentov a klúčov**.

Algoritmus bude spracovávať dva typy dokumentov:

- **Obrazové súbory obsahujúce klúče** (Tieto sa analyzujú v niekoľkých krokoch pomocou algoritmu na spracovanie obrazu)
- **Obrazové súbory obsahujúce zašifrované správy** (Po analýze a spracovaní systém správy dešifruje)

Vedúci tímu zdôraznil historický význam šifrovania, ktoré zohralo klúčovú úlohu pri ochrane údajov v čase vojny, ako aj pri vývoji informačných technológií. V histórii ľudstva bolo vytvorených mnoho dokumentov, ktoré obsahujú kódy, ktoré sú stále nerozlúštitelné, a tento projekt sa zameriava na uspokojenie tejto potreby dešifrovania.

Systém ponúka **užívateľsky prívetivé rozhranie**, ktoré poskytuje **poloautomatický systém** spracovania obrazu. Technické podrobnosti a operačné kroky algoritmu budú vysvetlené v ďalších konzultáciách.

Bod programu č. 2: Organizačné otázky

Skupina organizuje týždenné konzultácie v pondelok o 8:00 hod. Čas sa môže v budúcnosti zmeniť, ale zatiaľ je to vopred dohodnutý čas. Vedúci skupiny bude priebežne k dispozícii na diskusiu o všetkých otázkach, ktoré sa vyskytnú počas procesu vývoja, prostredníctvom servera Discord skupiny.

Diskutovalo sa aj o vytvorení spoločného úložiska GitHub, kde bude zdrojový kód dostupný všetkým členom. To umožní rýchlejšiu a efektívnejšiu spoluprácu pri vývoji projektu.

	8:00 - 8:50	9:00 - 9:50	10:00 - 10:50	11:00 - 11:50	12:00 - 12:50	13:00 - 13:50	14:00 - 14:50	15:00 - 15:50	16:00 - 16:50	17:00 - 17:50
Pondelok						I-UPB (1,2,3,4,5)		I-UPB (1,2,3,4,5)		
Úterok				I-RZZ / I-ZKRY (1,2,3,4,5)						
Streda	I-LOG (1,2,3,4,5)		I-LOG (1,2,3,4,5)			I-ZKRY (1,2)				
Štvrtok	I-SUNS (1,2,3,4,5)		I-RZZ (3,4,5)							
Piatok		I-SUNS (1,3,4,5)		I-SUNS (2)						

Prednáška	Cvičenie	1 Bc. Bence Bohm
		2 Bc. Matúš Horváth
		3 Bc. Jozef Nytrai
		4 Bc. Áron Tókai
		5 Bc. Vincent Pálfi

Obr. 36: Rozvrh študentov v zimnom semestri

7.2 1. semester - 2. týždeň

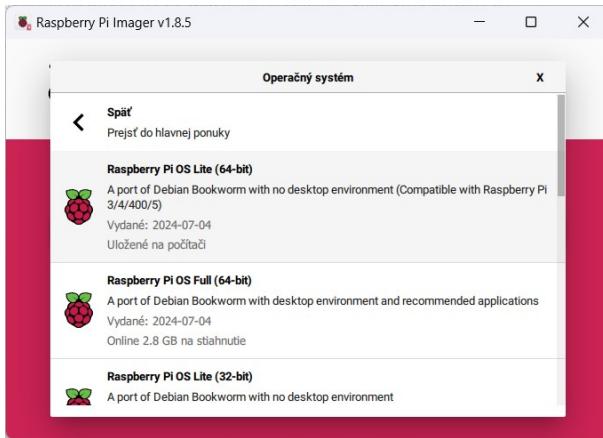
- **Dátum:** 07.10.2024, 08:00
- **Umiestnenie:** Slovenská technická univerzita v Bratislave, Fakulta elektrotechniky a informatiky
- **Miestnosť:** C614 (blok C, 6. poschodie)
- **Účastníci:**
 - Bc. Bence Both,
 - Bc. Matyas Horváth,
 - Bc. Jozef Nyitrai,
 - Bc. Áron Tükör,
 - Bc. Vincent Pálfy
- **Vedúci:** Ing. Stanislav Marochok

Bod programu č. 1: Nastavenie Web Server pre dokumentáciu tímového projektu

Počas konzultácií sa diskutovalo o špecifikáciách webovej stránky pre protokol. Webový server je založený na mikropočítači **Raspberry Pi 3** [4], na ktorom je nainštalovaná služba **WordPress** [5]. WordPress je open-source systém na správu obsahu, ktorý možno použiť na jednoduché vytváranie blogov alebo aj zložitejších webových stránok. V našom prípade sa použije na vytvorenie protokolu, takže bude fungovať ako blog. Na tento týždeň sme dostali za úlohu implementovať tento bod, preto si ho ďalej podrobne predstavíme.

Bod programu č. 2: Implementácia

Prvým krokom je inštalácia operačného systému na Raspberry Pi. Keďže nie je potrebné grafické používateľské rozhranie, vybrali sme verziu **Raspberry OS Lite** [6]. Na inštaláciu je potrebná aplikácia **Raspberry Pi Imager** [7], ktorá vykoná inštaláciu. Komunikácia s Raspberry prebieha prostredníctvom SSH. Počas inštalácie môžeme vidieť nasledovné okno:



Obr. 37: Inštalácia Raspberry OS Lite

Po inštalácii operačného systému na Raspberry Pi je potrebné nainštalovať ďalšie komponenty, ktoré umožnia správne fungovanie WordPressu. Každá časť je dôležitá na zaistenie kompletnej funkčnosti webového servera, databázy a backendu.

Inštalácia Apache webového servera

Apache [8] je jeden z najpopulárnejších open-source webových serverov, ktorý umožňuje webovým stránkam, ako napríklad WordPress, byť dostupnými cez internet. Apache spracováva požiadavky používateľov (napríklad zadané URL adresy) a odosiela im príslušný obsah zo servera. Po jeho inštalácii Raspberry Pi dokáže fungovať ako webový server, na ktorom bude WordPress umiestnený. Inštalácia vyzerá nasledovne:

```
sudo apt-get install apache2 -y
```

Výpis 7: Inštalácia Apache webového servera

Inštalácia PHP modulu

PHP (Hypertext Preprocessor) [9] je serverový skriptovací jazyk, ktorý WordPress využíva na dynamické generovanie webových stránok. PHP umožňuje WordPressu komunikovať s databázou, spracovávať požiadavky používateľov a generovať obsah na základe údajov uložených v databáze. Inštalácia PHP je klúčová pre správne fungovanie backendu WordPressu, pretože väčšina kódu WordPressu je napísaná práve v tomto jazyku. Okrem samotného PHP je často potrebné nainštalovať aj ďalšie PHP moduly.

```
sudo apt-get install php -y
```

Výpis 8: Inštalácia PHP Modulu

Inštalácia databázového správcu MariaDB

WordPress potrebuje databázu na ukladanie rôznych typov údajov, ako sú články, stránky, používateľské účty, nastavenia a ďalšie. Na tento účel sa používa **MariaDB** [10], ktorá je populárnu open-source alternatívou k MySQL. Po nainštalovaní MariaDB je nutné vytvoriť novú databázu špeciálne pre WordPress, do ktorej bude systém ukladať všetky potrebné dátu. Okrem toho sa vytvorí aj používateľ s príslušnými právami, aby mohol WordPress pristupovať k databáze. Tento krok je nevyhnutný, pretože WordPress bez databázového prístupu nemôže ukladať ani načítavať údaje.

```
sudo apt-get install mariadb-server php-mysql -y
sudo mysql_secure_installation
sudo mysql -uroot -p

CREATE wordpress DATABASE;
GRANT ALL PRIVILEGES ON wordpress.* TO 'root'@'localhost'
IDENTIFIED BY 'password';
FLUSH PRIVILEGES;
```

Výpis 9: Nastavenie databázy pre WordPress

Nastavenie WordPress

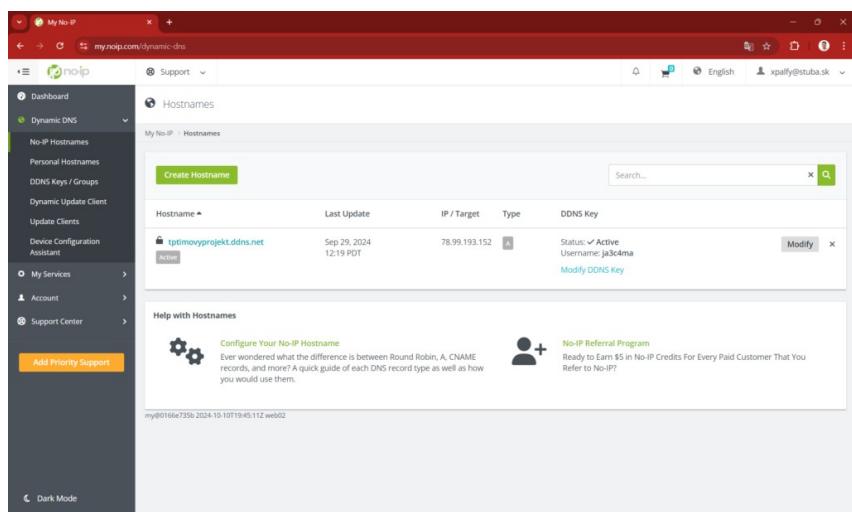
Po nainštalovaní všetkých potrebných modulov je ďalším krokom inštalácia WordPress. Najprv si musíme stiahnuť **najnovšiu verziu** WordPress z oficiálnej webovej lokality, ktorá sa dodáva ako komprimovaný súbor. Ten je potrebné rozbalíť do adresára webového servera, aby boli súbory prístupné pre Apache. Po rozbalení sa vykoná konfigurácia WordPress. Po dokončení konfigurácie je možné spustiť proces inštalácie prostredníctvom prehliadača zadáním IP adresy webového servera. Tu môžeme vytvoriť konto správcu a začať používať WordPress.

```
sudo wget http://wordpress.org/latest.tar.gz
sudo tar xzf latest.tar.gz
sudo mv wordpress/* .
sudo chown -R www-data: .
```

Výpis 10: Stiahnutie a príprava WordPress súborov na serveri

Nastavenie domain a HTTPS pomocou NO-IP.com

Naša webová stránka bude doteraz dostupná len v miestnej sieti. Aby bola dostupná cez internet, musíme na svojom smerovači nastaviť presmerovanie portov tak, aby bolo povolené presmerovanie portov **80** a **443**. Tieto porty sú potrebné na fungovanie protokolov **HTTP** a **HTTPS**. Služba bezplatnej webovej domény NO-IP.com nám umožňuje vybrať si bezplatný názov domény pre našu webovú lokalitu a prepojiť ju so službou NO-IP, čím sa webová lokalita stane dostupnou na internete. V tejto fáze však bude webová lokalita fungovať len cez protokol HTTP. Aby sme mohli používať protokol HTTPS, je potrebné získať certifikáty HTTPS. To možno vykonať na Raspberry Pi pomocou nástroja **certbot**.



Obr. 38: Nastavenie mena hostu a cieľovej IP adresy na webovej stránke NO-IP.

Vďaka tomu je naša webová stránka nielen dostupná na internete, ale môže fungovať aj prostredníctvom zabezpečeného pripojenia. V súčasnosti sme dokončili implementáciu webovej stránky s webovými lokalitami, ktoré bude dokumentovať zvyšok projektu. Odkaz na našu webovú lokalitu: <https://ptimovyprojekt.ddns.net>.

7.3 1. semester - 3. týždeň

- **Dátum:** 14.10.2024, 08:00
- **Umiestnenie:** Slovenská technická univerzita v Bratislave, Fakulta elektrotechniky a informatiky
- **Miestnosť:** C614 (blok C, 6. poschodie)
- **Účastníci:**
 - Bc. Bence Both,
 - Bc. Matyas Horváth,
 - Bc. Jozef Nyitrai,
 - Bc. Áron Tükör,
 - Bc. Vincent Pálfy
- **Vedúci:** Ing. Stanislav Marochok

Bod programu č. 1: Pokrok v implementácii a priebežné zmeny

Tretí týždeň sme sa zamerali na **návrh a realizáciu základných komponentov webovej aplikácie**, ktorej cieľom je spracovanie dokumentov prostredníctvom viacstupňových operácií. V prvom rade sme si ujasnili architektúru systému a určili hlavné funkčné prvky: **registráciu používateľov, možnosť prihlásenia a bezpečný prístup k dokumentom**, ktoré si používatelia budú môcť nahrávať, upravovať a neskôr dešifrovať.

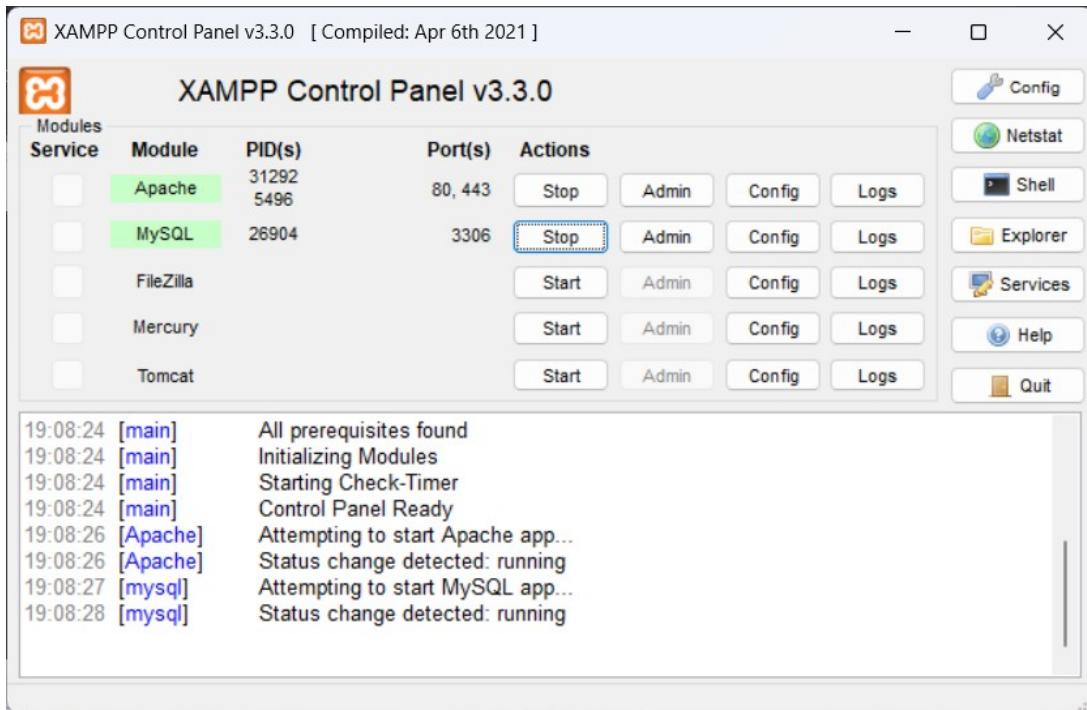
Začali sme s implementáciou regisračného a prihlasovacieho modulu. Vytvorili sme databázovú štruktúru, ktorá obsahuje **tabuľku pre používateľov**, kde každý záznam obsahuje jedinečné **ID, meno, e-mail** a **zašifrované heslo**. Týmto krokom sme zabezpečili ochranu údajov a predchádzanie duplicitám.

Registrácia prebieha prostredníctvom **formulára**, kde používateľ zadáva svoje údaje a heslo dvakrát pre overenie správnosti. V prípade chýb ako duplicitné meno alebo nesprávne heslo systém poskytuje spätnú väzbu pomocou vizuálnych upozornení. Po úspešnej registrácii sa používateľ môže prihlásiť a získať prístup k chránenej časti aplikácie, ktorá je dostupná iba **overeným používateľom**.

Na vývoj sme použili **XAMPP** [15] ako lokálne vývojové prostredie, ktoré zahŕňa Apache server, PHP, databázu MariaDB a správu cez PHPMyAdmin. Frontend aplikácie sme vytvárali s využitím **HTML** [16], **CSS** [17] a **JavaScriptu** [18], pričom sme

implementovali aj *Bootstrap* [19] a *jQuery* [20] pre vytvorenie moderného, responzívneho a používateľsky priateľského rozhrania.

Táto fáza predstavuje stabilný základ pre ďalšie kroky projektu, najmä pre spracovanie dokumentov a prácu so súbormi, ktoré budú nahrávať samotní používatelia.



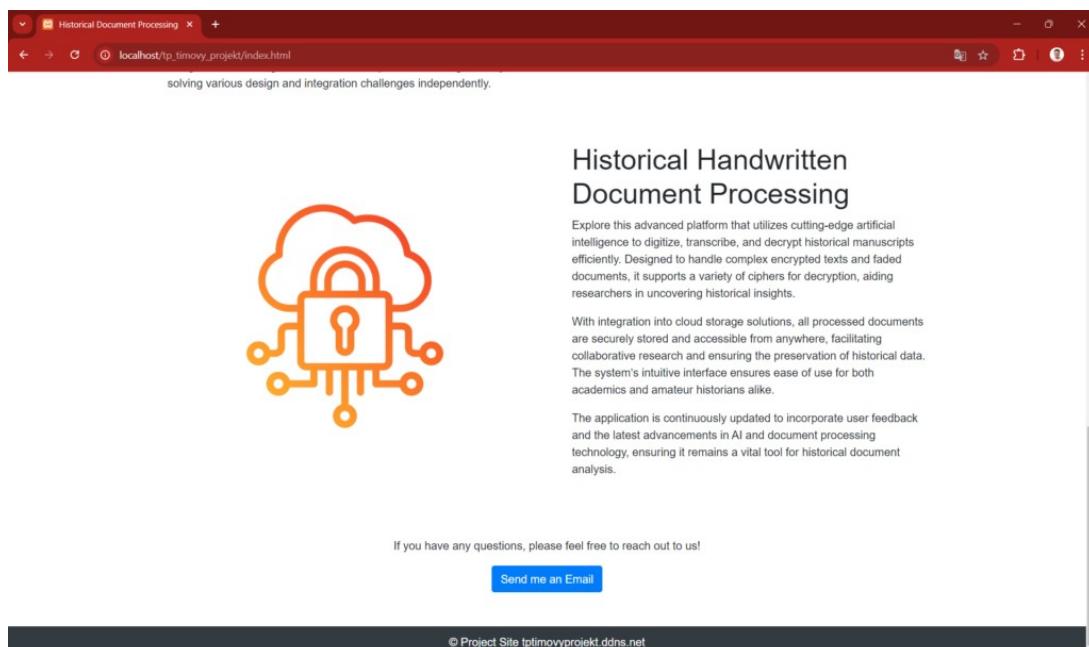
Obr. 39: Ovládací panel XAMPP

Bod programu č. 2: Plány na nasledujúci týždeň

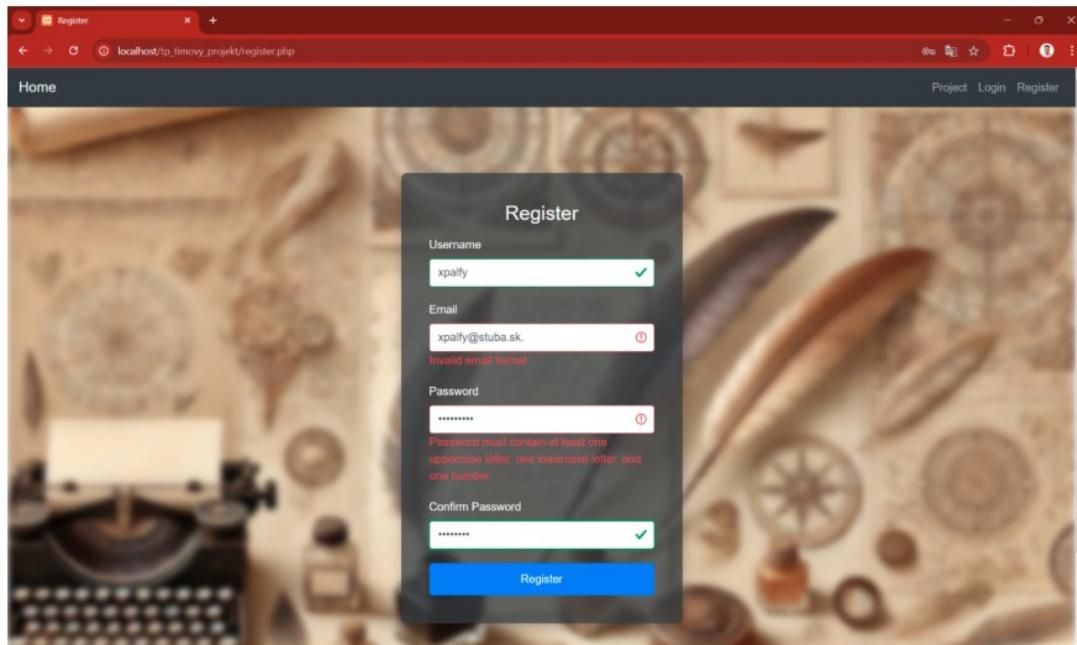
V nasledujúcim týždni plánujeme rozšíriť aplikáciu o základné používateľské rozhranie na **nahrávanie obrázkov**. Toto rozhranie bude dostupné len pre prihlásených používateľov a bude navrhnuté tak, aby umožnilo intuitívne nahrávanie buď pretiahnutím súboru do určeného poľa, alebo klasickým výberom cez správcu súborov.

Zároveň plánujeme navrhnúť **databázovú štruktúru** na ukladanie informácií o obrázkoch – vrátane cesty k súboru a prepojenia na konkrétneho používateľa. Uvažujeme aj nad základnými možnosťami kategorizácie, napríklad možnosťou označiť, či ide o obrázok obsahujúci klúč alebo textový obsah. Tieto metadáta budú v budúcnosti využité pri spracovaní dokumentov a dešifrovaní.

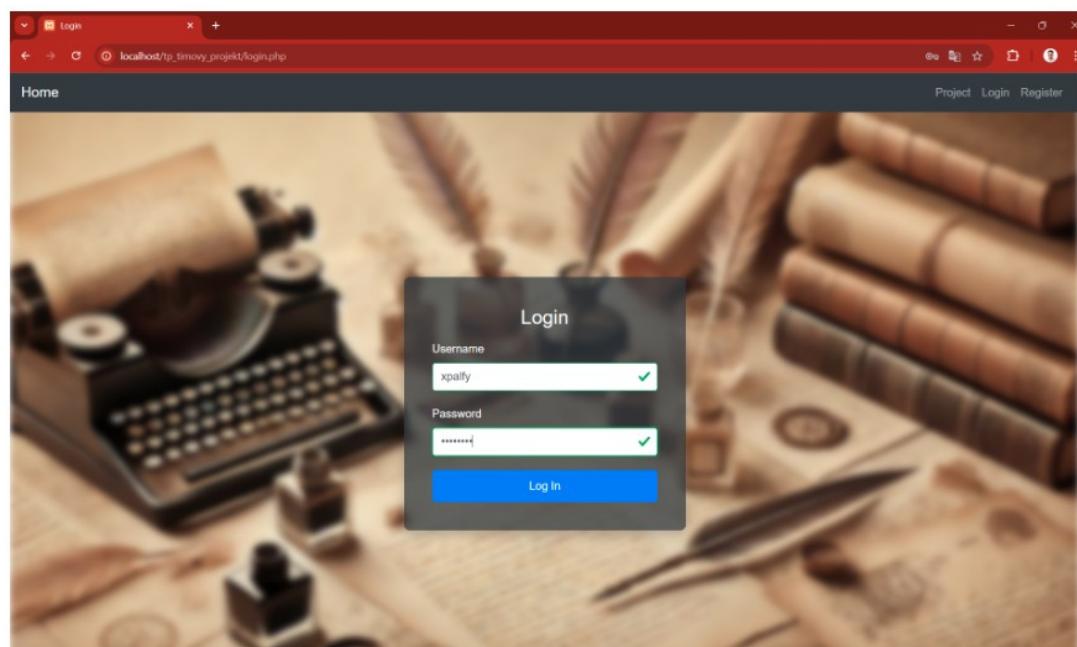
Súčasťou plánov je aj návrh ďalších častí používateľského rozhrania – napríklad **sekcie pre správu účtu** alebo **históriu nahratých súborov**, ktoré budú implementované v ďalších fázach vývoja.



Obr. 40: Hlavná stránka projektu



Obr. 41: Podstránka registrácie



Obr. 42: Podstránka prihlásenia

7.4 1. semester - 4. týždeň

- **Dátum:** 22.10.2024, 09:00
- **Umiestnenie:** Slovenská technická univerzita v Bratislave, Fakulta elektrotechniky a informatiky
- **Miestnosť:** C517 (blok C, 5. poschodie)
- **Účastníci:**
 - Bc. Bence Both,
 - Bc. Matyas Horváth,
 - Bc. Jozef Nyitrai,
 - Bc. Áron Tükör,
 - Bc. Vincent Pálfy
- **Vedúci:** Ing. Stanislav Marochok

Bod programu č. 1: Pokrok v implementácii a priebežné zmeny

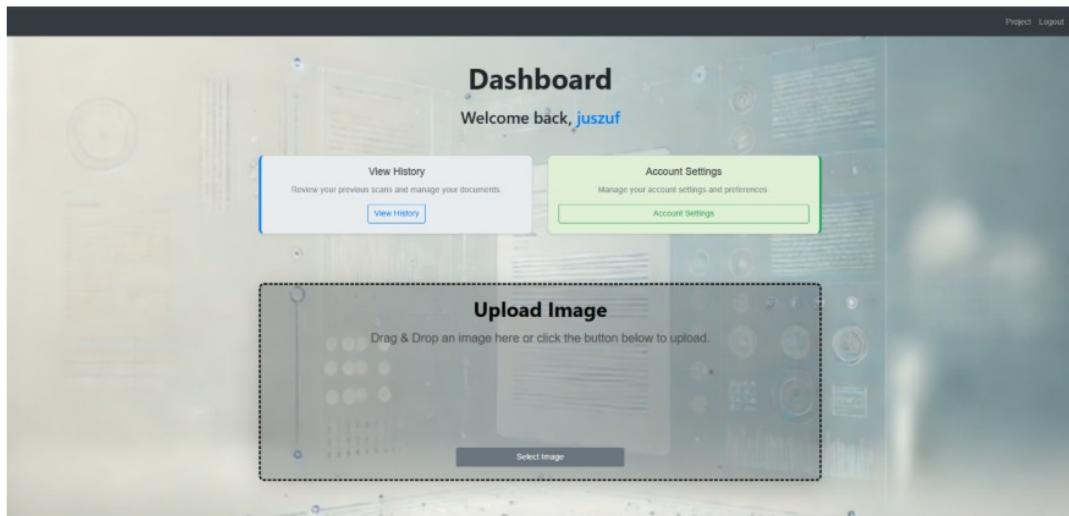
Tento týždeň sme sa zamerali na vytvorenie **používateľsky prívetivého rozhrania** na nahrávanie obrázkov, ktoré je dostupné výhradne pre prihlásených používateľov. Rozhranie bolo navrhnuté tak, aby umožňovalo pohodlné nahrávanie obrázkov buď klasickým výberom zo súborov, alebo pretiahnutím do určeného pola.

Po nahratí obrázka je spracovaný a jeho náhľad sa okamžite zobrazí v používateľskom rozhraní, čo umožňuje používateľovi skontrolovať správnosť nahrania. Samotné obrázky sa neukladajú do databázy – do databázy sa zapisuje iba **cesta k súboru**, čo zabezpečuje vyššiu výkonnosť a rýchlejšie načítavanie.

V databáze sme vytvorili tabuľku **pictures**, ktorá obsahuje ID obrázka, cestu k súboru a prepojenie na ID používateľa. Zabezpečili sme aj referenčné obmedzenia – napríklad automatické zmazanie obrázkov pri zmazaní používateľa.

Rozhranie zároveň umožňuje kategorizáciu nahratých obrázkov – používateľ môže vybrať, či ide o obrázok s klúčom (**KEY**) alebo o obrázok so šifrovaným textom (**CIPHER**). Tento údaj sa uloží spolu s obrázkom a bude slúžiť na následné spracovanie.

Zároveň sme pripravili základy pre budúce funkcionality – do rozhrania sme pridali tlačidlá pre historiu nahrávaných súborov a pre správu používateľského účtu. Tieto funkcie ešte nie sú aktívne, ale vytvárajú priestor pre ich neskoršiu integráciu.



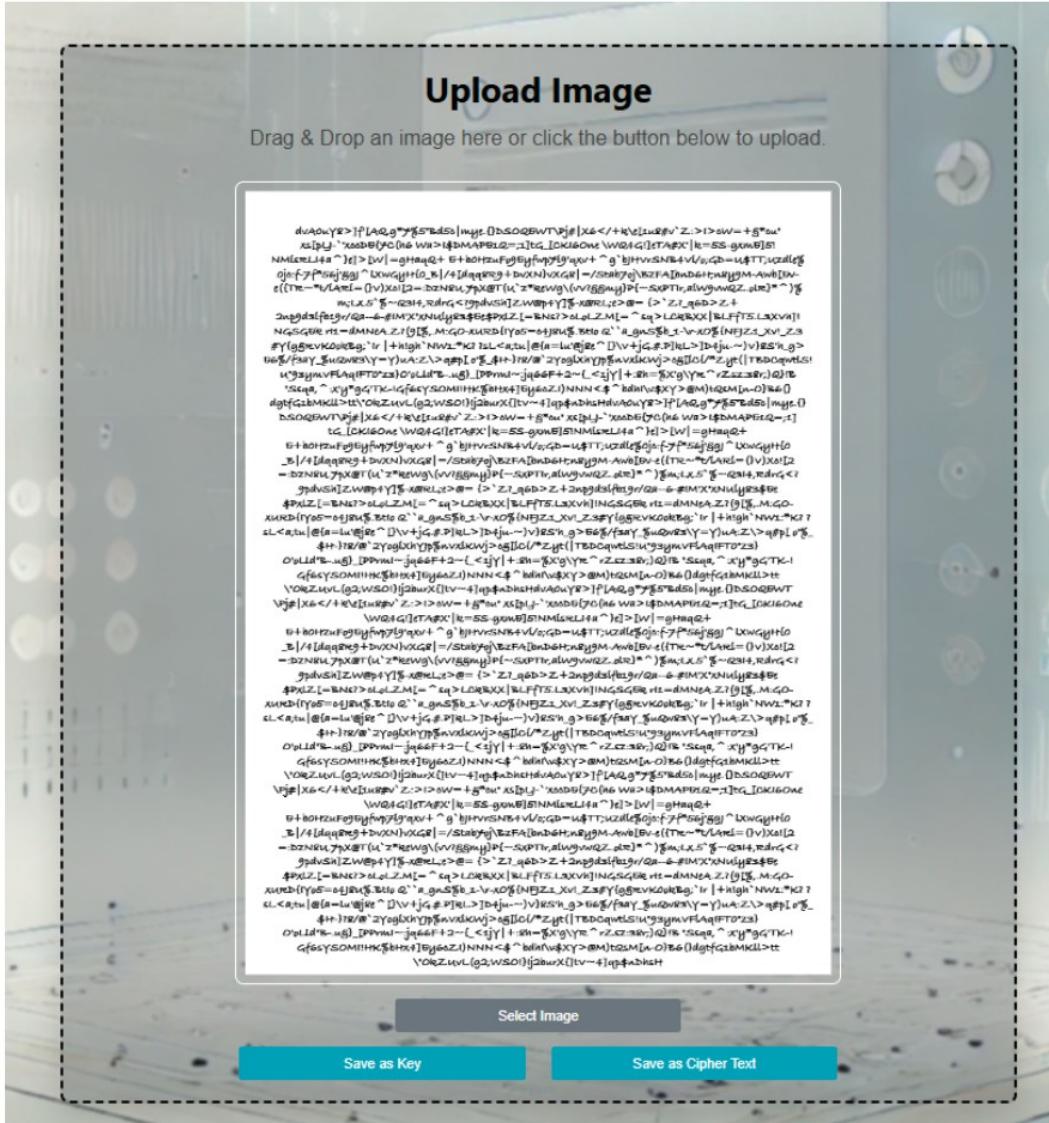
Obr. 43: Hlavná stránka po prihlásovaní

Bod programu č. 2: Plány na nasledujúci týždeň

V ďalšom týždni sa plánujeme sústrediť na zjednodušenie vývoja aplikácie a prípravu systému na online prevádzku. Cieľom je presunúť aplikáciu z lokálneho prostredia do **verejne dostupného online priestoru**.

Jedným z hlavných krokov bude nasadenie aplikácie na online server a jej zabezpečenie prostredníctvom služby Cloudflare, ktorá nám umožní spravovať DNS záznamy, poskytne ochranu pred DDoS útokmi a zaistí bezpečný prístup pomocou tunelovania (Cloudflare Zero Trust Tunnel).

Plánujeme tiež vytvoriť individuálne vývojové priečinky pre každého člena tímu, kde si bude môcť nezávisle testovať a vyvíjať nové funkcie. Hotové riešenia sa následne budú integrovať do hlavnej aplikácie cez Git a automatizovaný systém stahovania zmien. Týmto krokom sa zvýší efektivita tímovej spolupráce a zabezpečí sa prehľadnosť vývoja.



Obr. 44: Ukážka nahrania obrázka na hlavnej stránke

7.5 1. semester - 5. týždeň

- **Dátum:** 29.10.2024, 09:00
- **Umiestnenie:** Slovenská technická univerzita v Bratislave, Fakulta elektrotechniky a informatiky
- **Miestnosť:** C517 (blok C, 5. poschodie)
- **Účastníci:**
 - Bc. Bence Both,
 - Bc. Matyas Horváth,
 - Bc. Jozef Nyitrai,
 - Bc. Áron Tükör,
 - Bc. Vincent Pálfy
- **Vedúci:** Ing. Stanislav Marochok

Bod programu č. 1: Pokrok v implementácii a priebežné zmeny

Tento týždeň sme sa sústredili na analýzu požiadaviek projektu a optimalizáciu vývojového prostredia webovej aplikácie. V rámci konzultácií sme si ujasnili, že naším cieľom nie je vyvíjať samotné moduly na spracovanie obrázkov a šifrovania textov, ale vytvoriť webové rozhranie, ktoré ich umožní efektívne **integrovať** a **koordinovať**.

Doteraz sme využívali vývoj v lokálnom prostredí pomocou XAMPP na jednotlivých počítačoch. Aby sme vývoj zefektívnil a zjednotili prístup pre všetkých členov tímu, rozhodli sme sa vytvoriť centrálny online server, kde môže každý vývojár pracovať vo vlastnom vyhradenom priečinku. Vývojové verzie sú odtiaľ cez **Git** [21] synchronizované na hlavnú stránku projektu pomocou automatizovaného systému autopull.

Na sprístupnenie aplikácie verejnosti a zabezpečenie online prevádzky sme nasadili server s technológiami LAMP (Linux, Apache, MariaDB, PHP). Pre bezpečnosť a globálnu dostupnosť sme využili služby **Cloudflare** [22], vrátane **Cloudflare Zero Trust Tunnel** [23], ktorý umožňuje bezpečný prístup bez potreby presmerovania portov.

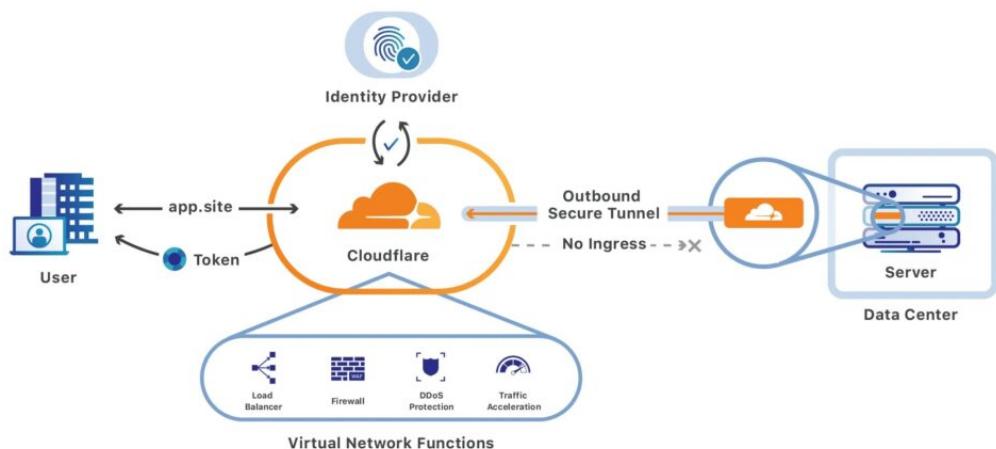
Zároveň sme si zakúpili doménu a nakonfigurovali DNS záznamy, čím sme pripravili aplikáciu na profesionálne nasadenie. Toto riešenie prinieslo nielen vyššiu bezpečnosť (SSL, firewall), ale aj lepšiu dostupnosť pre používateľov. Týmto krokom sme vytvorili stabilné a bezpečné prostredie pre ďalší vývoj aplikácie.

Odkaz na vlastný priečinok pre vývojárov je:

<https://test.tptimovskyprojekt.software/xpalfy/>

Odkaz na vlastný priečinok pre vývojárov je:

https://test.tptimovskyprojekt.software/tp_timovsky_projekt/.



Obr. 45: Integrácia brány Cloudflare a prístupu

Bod programu č. 2: Plány na nasledujúci týždeň

Na ďalší týždeň plánujeme implementovať možnosť integrácie spracovateľských modulov pomocou Pythonu. Cieľom je vytvoriť funkčné **prepojenie medzi frontendom a backendom**, ktorý bude schopný vykonávať Python skripty.

Budeme sa sústredíť na inštaláciu a konfiguráciu Pythonu na serveri, vrátane potrebných balíkov a testovania spúšťania skriptov z PHP kódu. Tento krok nám umožní postupne vybudovať systém, v ktorom budú jednotlivé kroky spracovania údajov (napr. analýza obrázkov alebo extrakcia textu) prebiehať hladko a v nadväzujúcich fázach.

Zároveň plánujeme pokračovať v analýze požiadaviek na tieto integračné procesy, aby sme minimalizovali riziká a nastavili čo najefektívnejšiu štruktúru komunikácie medzi modulmi a používateľským rozhraním.

Obr. 46: Menu aplikácie CloudFlare

Obr. 47: CloudFlare public hostname menu

7.6 1. semester - 6. týždeň

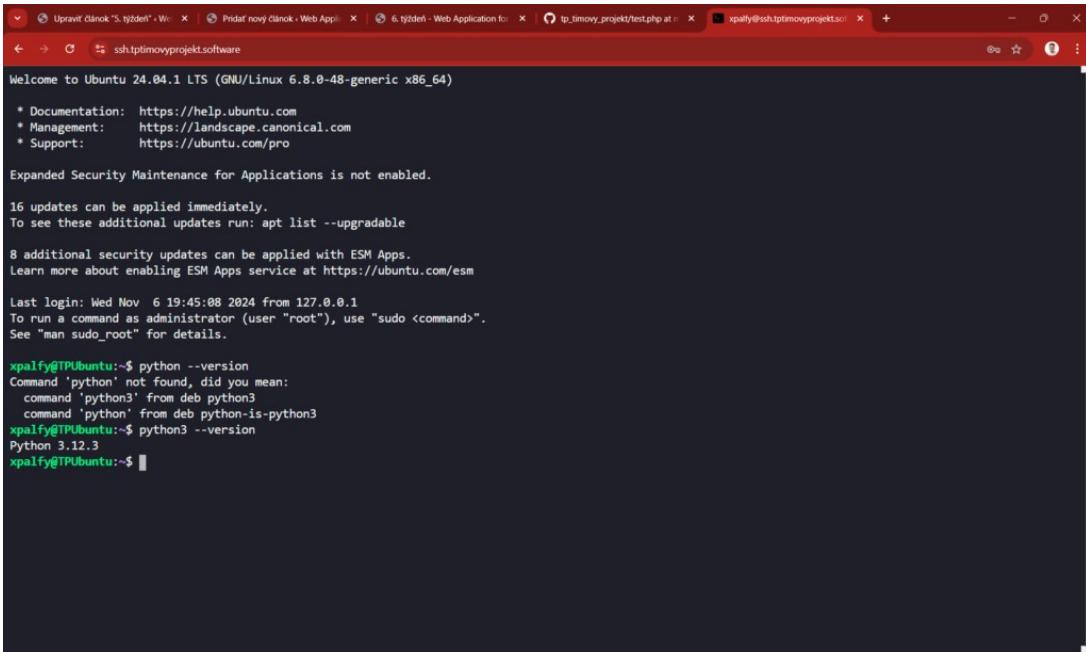
- **Dátum:** 05.11.2024, 09:00
- **Umiestnenie:** Slovenská technická univerzita v Bratislave, Fakulta elektrotechniky a informatiky
- **Miestnosť:** C517 (blok C, 5. poschodie)
- **Účastníci:**
 - Bc. Bence Both,
 - Bc. Matyas Horváth,
 - Bc. Jozef Nyitrai,
 - Bc. Áron Tükör,
 - Bc. Vincent Pálfy
- **Vedúci:** Ing. Stanislav Marochok

Bod programu č. 1: Pokrok v implementácii a priebežné zmeny

V rámci tohto týždňa sme sa sústredili na **prepojenie webovej aplikácie s back-endovými procesmi** pomocou **Pythonu** [24]. Cieľom bolo vytvoriť technické zázemie pre spúšťanie spracovateľských skriptov, ktoré budú neskôr obsluhovať obrázky s klúčmi alebo šifrovaným textom.

Na serveri s LAMP architektúrou sme úspešne nainštalovali Python a nakonfigurovali prostredie tak, aby bolo možné spúšťať Python skripty priamo z PHP. Vytvorili sme jednoduchý testovací systém, ktorý overil, že PHP dokáže preniesť vstupné dátá do Pythonu a získať spätný výstup. Týmto krokom sme pripravili základ pre refazec spracovania údajov, kde každý krok môže byť obsluhovaný samostatným Python modulom.

Zároveň sme sa venovali analýze potrieb integrácie týchto procesov, aby vývoj prebiehal plynule a umožnil neskoršiu rozšíriteľnosť systému. Cieľom bolo vytvoriť rámec, v ktorom jednotlivé komponenty (frontend, backend, spracovateľské moduly) budú navzájom prepojené, a zároveň flexibilné na budúce úpravy a rozšírenia.



```
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-48-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

16 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

8 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Wed Nov  6 19:45:08 2024 from 127.0.0.1
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

xpafy@TPUbuntu:~$ python --version
Command 'python' not found, did you mean:
  command 'python3' from deb python3
  command 'python' from deb python-is-python3
xpafy@TPUbuntu:~$ python3 --version
Python 3.12.3
xpafy@TPUbuntu:~$
```

Obr. 48: Ukážka python verzie na serveri



Obr. 49: Ukážka spušteného python skriptu na webstránke

Bod programu č. 2: Plány na nasledujúci týždeň

Na ďalší týždeň plánujeme sústrediť sa na prípravu **technickej a vizuálnej dokumentácie projektu**. V rámci dokumentácie budeme detailne popisovať postupy, ktoré sme doposiaľ realizovali, a zabezpečíme jednotnú štruktúru prostredníctvom nástroja Overleaf, ktorý umožňuje efektívnu tímovú spoluprácu v LaTeX-e.

Súčasťou plánov je aj návrh používateľského rozhrania v nástroji Figma. Prostredníctvom návrhov jednotlivých častí aplikácie si chceme ujasniť štruktúru a vzhľad stránky. Vizuálne prototypy nám pomôžu lepšie pochopiť fungovanie aplikácie, identifikovať prípadné problémy a pripraviť sa na ďalšiu fázu implementácie.

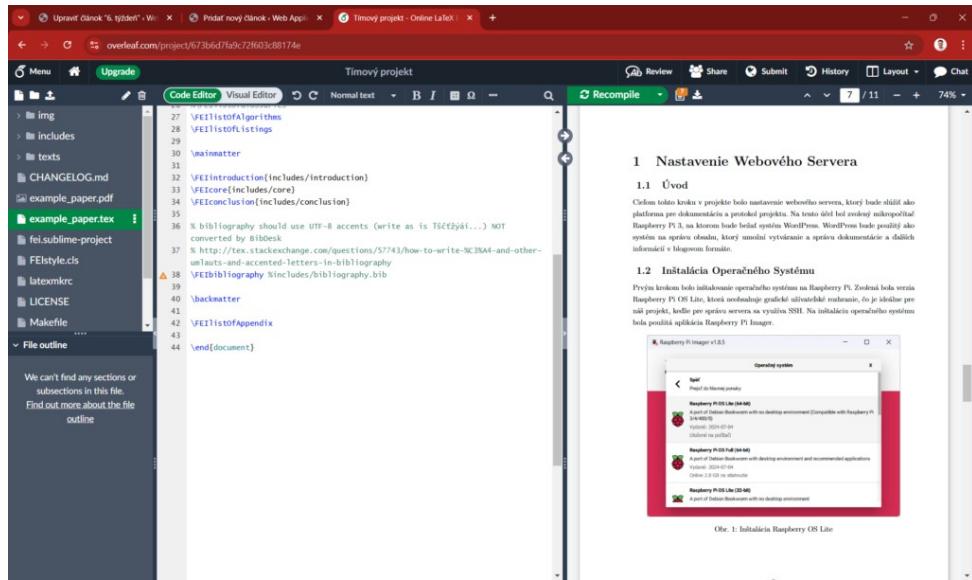
7.7 1. semester - 7. týždeň

- **Dátum:** 12.11.2024, 09:00
- **Umiestnenie:** Slovenská technická univerzita v Bratislave, Fakulta elektrotechniky a informatiky
- **Miestnosť:** C517 (blok C, 5. poschodie)
- **Účastníci:**
 - Bc. Bence Both,
 - Bc. Matyas Horváth,
 - Bc. Jozef Nyitrai,
 - Bc. Áron Tükör,
 - Bc. Vincent Pálfy
- **Vedúci:** Ing. Stanislav Marochok

Bod programu č. 1: Pokrok v implementácii a priebežné zmeny

V priebehu tohto týždňa sme sa zamerali na prehľbenú analýzu nášho projektu, s cieľom lepšie pochopiť jeho štruktúru, nároky a potreby koncových používateľov. Projekt sa postupne rozrastá a jeho komplexnosť si vyžaduje systematickú dokumentáciu a plánovanie. Z tohto dôvodu sme sa rozhodli začať s vypracovaním podrobnej **projektovej dokumentácie**, ktorá popisuje jednotlivé kroky vývoja, implementované riešenia a celkový priebeh realizácie.

Dokumentácia vzniká v prostredí **Overleaf** [25], ktoré umožňuje prácu v **LaTeX-e** [26] a efektívnu tímovú spoluprácu v reálnom čase. Vďaka tejto platforme vieme udržať jednotný formát, automaticky kontrolovať štruktúru a syntax dokumentu a zároveň predchádzať verziovacím konfliktom medzi členmi tímu.

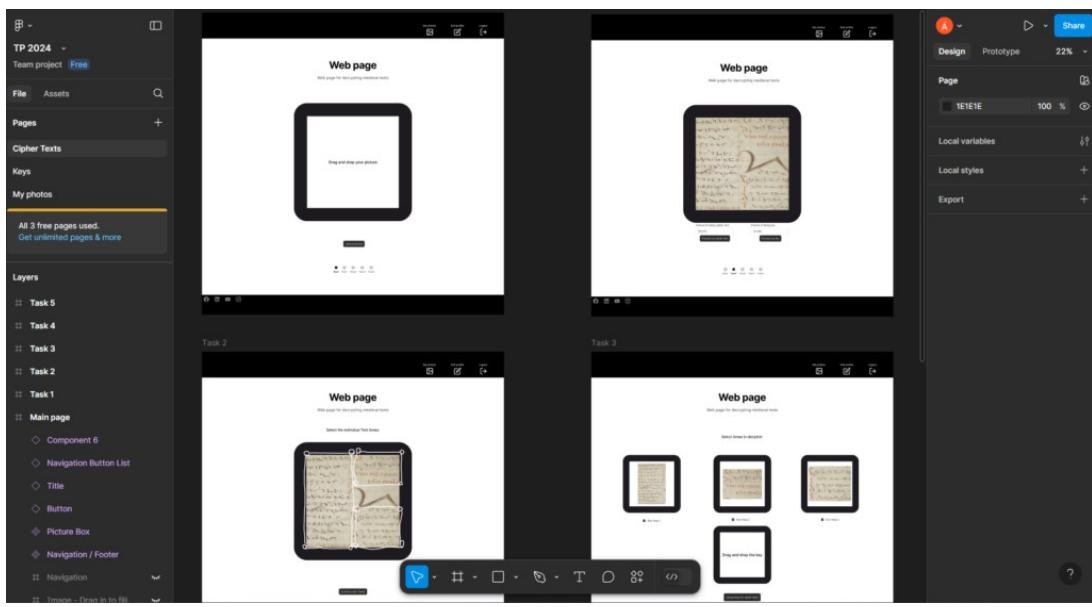


Obr. 50: Rozhranie aplikácie Overleaf

Popri písaní dokumentácie sme sa venovali aj návrhu používateľského rozhrania našej webovej aplikácie. Na tento účel sme využili nástroj ***Figma*** [27], ktorý nám umožnil navrhnúť vizuálny vzhľad a funkčné rozloženie viacerých stránok aplikácie. Pracovali sme na troch hlavných prototypoch:

- **Keys.pdf:** Tento dokument obsahuje návrh webovej stránky určenej na spracovanie fotiek súvisiacich s *Keys*. Ide o vizualizáciu rozhrania, ktoré umožní efektívne spracovať a zobrazovať tieto dátu. Link na pdf: <https://tptimovyprojekt.ddns.net/wp-content/uploads/2024/11/Keys.pdf>
- **Cipher Text.pdf:** Tento dokument obsahuje návrh webovej stránky na spracovanie fotiek súvisiacich s *Ciphered Text*. V návrhu sú zahrnuté funkčné prvky na úpravu a manipuláciu s týmito špecifickými fotkami. Link na pdf: <https://tptimovyprojekt.ddns.net/wp-content/uploads/2024/11/Cipher-Text.pdf>
- **My photos.pdf:** Tento dokument predstavuje návrh webovej stránky na manažovanie fotiek. Obsahuje funkcie na nahrávanie, usporiadanie a správu všetkých fotiek, ktoré budú súčasťou projektu. Link na pdf: <https://tptimovyprojekt.ddns.net/wp-content/uploads/2024/11/My-photos.pdf>

Vytvorením týchto návrhov sme získali jasné predstavu o rozložení prvkov, navigácií a očakávanom správaní aplikácie. Výhodou Figmy je aj možnosť komentovania a tímovej práce, čo nám umožnilo operatívne reagovať na pripomienky a postupne dolaďovať návrhy. Vizualizácia rozhrania zároveň slúži ako dôležitý podklad pre frontendovú implementáciu a lepšiu koordináciu vývojového procesu.



Obr. 51: Rozhranie aplikácie Figma

Bod programu č. 2: Plány na nasledujúci týždeň

Na budúci týždeň plánujeme implementovať systém **zdieľania obrázkov medzi používateľmi**. Táto funkcia umožní, aby jeden nahraný obrázok mohol byť priradený viacerým používateľom bez potreby opäťovného nahrávania. Cieľom je zefektívniť prácu so súbormi, znížiť zaťaženie servera a podporiť spoluprácu medzi členmi tímu.

Technicky plánujeme vytvoriť novú databázovú tabuľku `users_pictures`, ktorá bude reprezentovať vzťah **M:N** medzi používateľmi a obrázkami. Zároveň pripravíme používateľské rozhranie, ktoré umožní jednoduché zadanie mena používateľa, s ktorým chce aktuálny vlastník obrázok zdieľať. Prístupové oprávnenia budú spravované bezpečne, s dôrazom na integritu údajov a možnosť neskoršieho odvolania zdieľania.

V prípade potreby plánujeme zahrnúť aj jednoduchý notifikačný mechanizmus, ktorý informuje používateľa o nových zdieľaných súboroch.

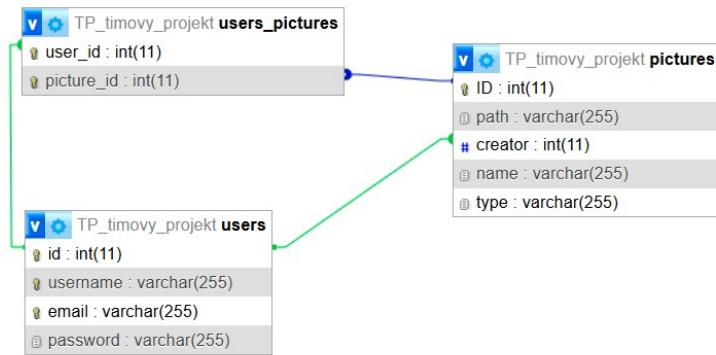
7.8 1. semester - 8. týždeň

- **Dátum:** 19.11.2024, 09:00
- **Umiestnenie:** Slovenská technická univerzita v Bratislave, Fakulta elektrotechniky a informatiky
- **Miestnosť:** C517 (blok C, 5. poschodie)
- **Účastníci:**
 - Bc. Bence Both,
 - Bc. Matyas Horváth,
 - Bc. Jozef Nyitrai,
 - Bc. Áron Tükör,
 - Bc. Vincent Pálfy
- **Vedúci:** Ing. Stanislav Marochok

Bod programu č. 1: Pokrok v implementácii a priebežné zmeny

Tento týždeň sme sa zamerali na rozšírenie funkcionality systému o možnosť **zdieľania obrázkov medzi používateľmi**. Našou snahou bolo zefektívniť správu súborov tak, aby už raz nahraté obrázky mohli byť dostupné viacerým používateľom bez nutnosti opäťovného nahrávania.

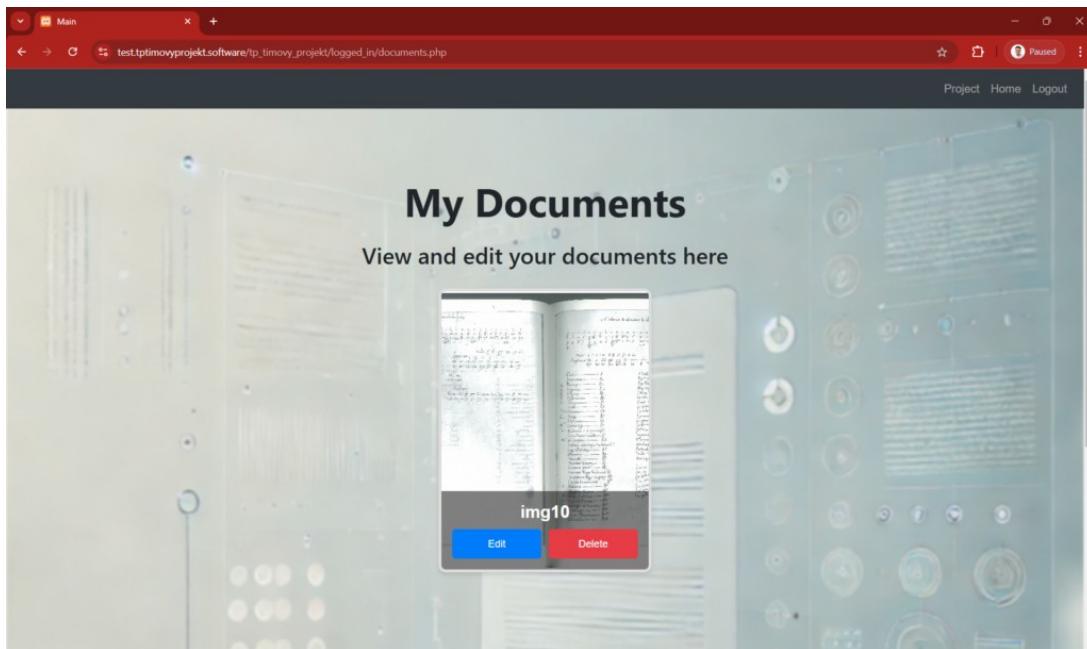
V rámci tejto úlohy sme vytvorili novú databázovú tabuľku `users_pictures`, ktorá zabezpečuje vzťah M:N medzi používateľmi a obrázkami. To znamená, že jeden obrázok môže byť priradený viacerým používateľom a naopak – jeden používateľ môže mať prístup k viacerým obrázkom. Funkcionalita bola navrhnutá tak, aby používateľ, ktorý vlastní obrázok, mohol prostredníctvom formulára zdieľať obrázok s inými používateľmi zadáním ich používateľského mena.



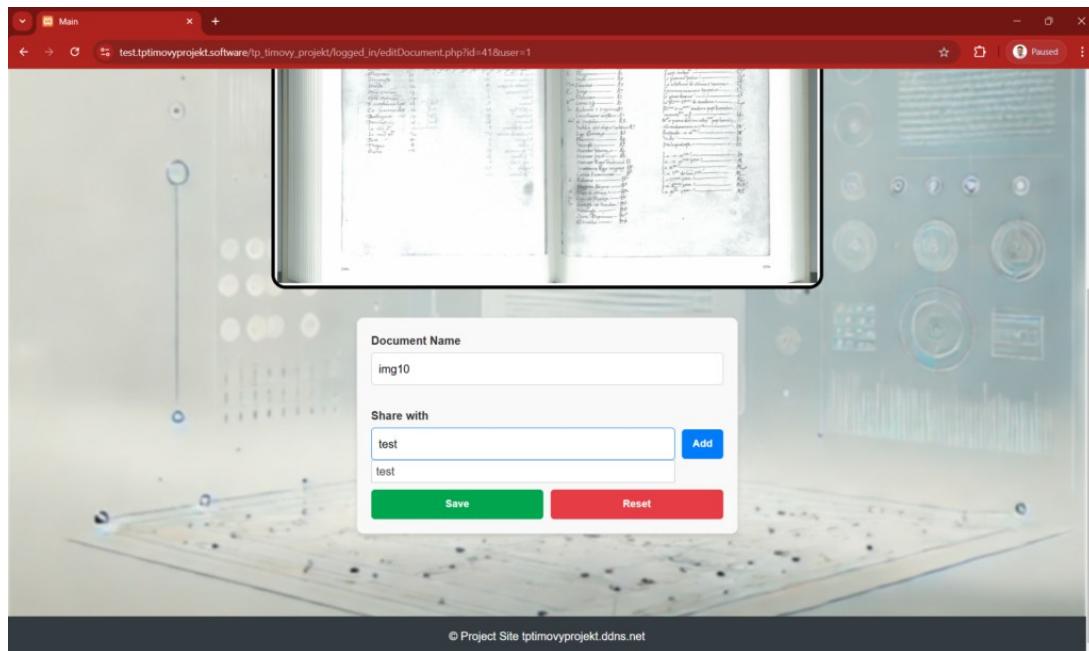
Obr. 52: Vytvorené tabuľky prepojenia používateľov s obrázkami

Tento prístup významne znižuje zataženie systému, **šetrí diskový priestor** a podporuje **spoluprácu medzi používateľmi**. Vďaka implementovanej referenčnej integrite v databáze sa zároveň zabezpečuje správna správa prepojení a spolahlivý priebeh operácií.

Popri tejto funkciaľite sme sa zaoberali aj implementáciou základného **mechanizmu notifikácií**, ktorý informuje používateľov o novo zdieľaných obrázkoch. Tento krok nám pomohol posunúť aplikáciu bližšie k použiteľnému systému pre reálnych používateľov, ktorý podporuje tímovú prácu a efektívne zdieľanie dát.



Obr. 53: Výber fotografie na úpravu



Obr. 54: Ukážka úpravy snímkov

Bod programu č. 2: Plány na nasledujúci týždeň

V nasledujúcim týždni plánujeme zamerat sa na **bezpečnostné aspekty** aplikácie a pokračovať v implementácii **autentifikačných mechanizmov**. Naším hlavným cieľom bude integrovať systém Bearer Token, ktorý umožní bezpečnú a efektívnu autorizáciu používateľov.

Plánujeme nasadiť knižnicu Firebase JWT pre PHP, prostredníctvom ktorej budeme generovať, podpisovať a overovať tokeny. Tieto tokeny budú uchovávať informácie o prihlásenom používateľovi a budú priložené ku každej požiadavke vo forme autorizačnej hlavičky.

Súčasťou práce bude aj kontrola existujúcich zápisníc a doplnenie chýbajúcich častí v dokumentácii, ako aj úprava niektorých štruktúr v rozhraní podľa aktuálnych návrhov vo Figme. Tieto kroky nám umožnia zvýšiť bezpečnosť aplikácie, zlepšiť prehľadnosť práce a pripraviť systém na ďalšie fázy vývoja.

7.9 1. semester - 9. týždeň

- **Dátum:** 26.11.2024, 09:00
- **Umiestnenie:** Slovenská technická univerzita v Bratislave, Fakulta elektrotechniky a informatiky
- **Miestnosť:** C517 (blok C, 5. poschodie)
- **Účastníci:**
 - Bc. Bence Both,
 - Bc. Matyas Horváth,
 - Bc. Jozef Nyitrai,
 - Bc. Áron Tükör,
 - Bc. Vincent Pálfy
- **Vedúci:** Ing. Stanislav Marochok

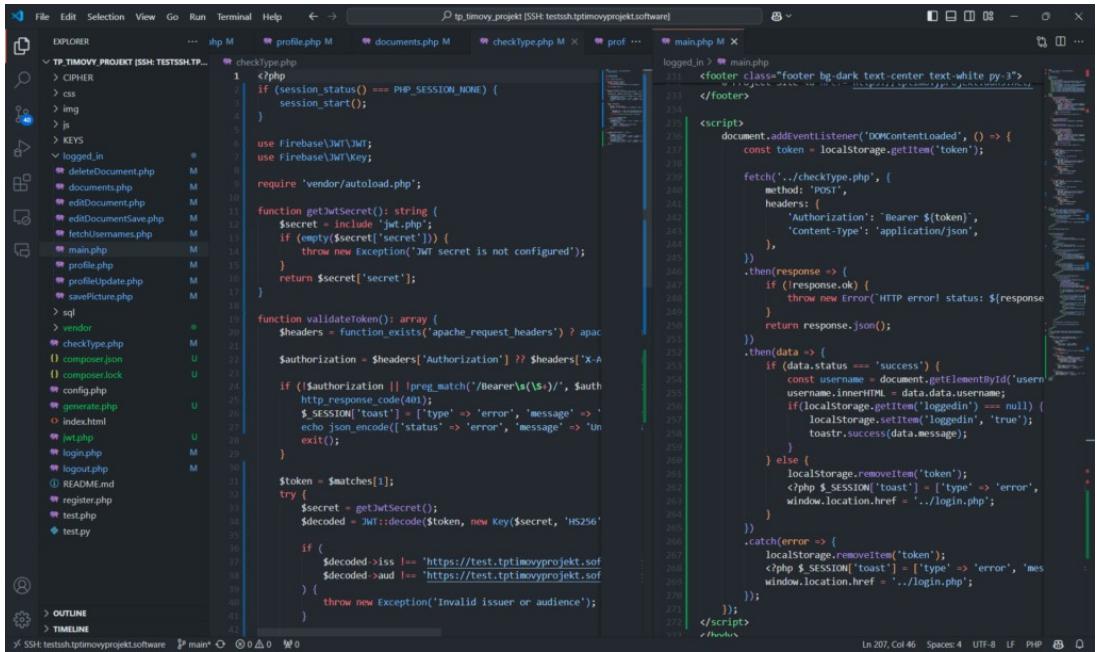
Bod programu č. 1: Pokrok v implementácii a priebežné zmeny

Tento týždeň sme sa zamerali na zvýšenie bezpečnosti našej webovej aplikácie prostredníctvom integrácie **Bearer Token** [12] autentifikácie. Táto funkcia umožňuje serveru bezpečne identifikovať používateľov pri každej požiadavke bez nutnosti opakovaného zadávania prihlásovacích údajov.

Na implementáciu sme využili knižnicu **Firebase JWT** [13] pre PHP, ktorú sme nainštalovali pomocou Composeru. Pri úspešnom prihlásení používateľa systém vygeneruje jedinečný token obsahujúci základné údaje ako ID a meno, čas vytvorenia a dobu platnosti. Token je podpísaný tajným klúčom a šifrovaný pomocou algoritmu **HS256**, čo zabezpečuje jeho ochranu pred neoprávnenou manipuláciou.

Klient následne uchová tento token a pri každej požiadavke ho zahrne do hlavičky „Authorization“. Server token overí – ak je platný a správne podpísaný, umožní prístup k chránenej funkcií; ak nie, pošle chybu **401 Unauthorized**. Tento krok výrazne zvyšuje bezpečnostnú úroveň aplikácie a umožňuje presnejšiu kontrolu prístupu.

Okrem technickej implementácie sme sa venovali aj revízie a aktualizácii projektovej dokumentácie. Identifikovali sme miesta, ktoré si vyžadujú doplnenie alebo preformulovanie, aby lepšie odrážali súčasný stav aplikácie. Zároveň sme revidovali štruktúru zápisníča a pripravili úpravy, ktoré zlepšia ich prehľadnosť a konzistentnosť.



```

<?php
if (session_status() === PHP_SESSION_NONE) {
    session_start();
}

use Firebase\JWT\JWT;
use Firebase\JWT\Key;

require 'vendor/autoload.php';

function getJwtSecret(): string {
    $secret = include 'jwt.php';
    if (empty($secret['secret'])) {
        throw new Exception('JWT secret is not configured');
    }
    return $secret['secret'];
}

function validateToken(): array {
    $headers = function_exists('apache_request_headers') ? apache_request_headers() : $_SERVER;
    $authorization = $headers['Authorization'] ?? $headers['X-AUTH-TOKEN'];

    if (!($authorization || preg_match('/Bearer\s(\S+)/', $authorization))) {
        http_response_code(401);
        $_SESSION['toast'] = ['type' => 'error', 'message' => 'Unauthorized'];
        echo json_encode(['status' => 'error', 'message' => 'Unauthorized']);
        exit();
    }

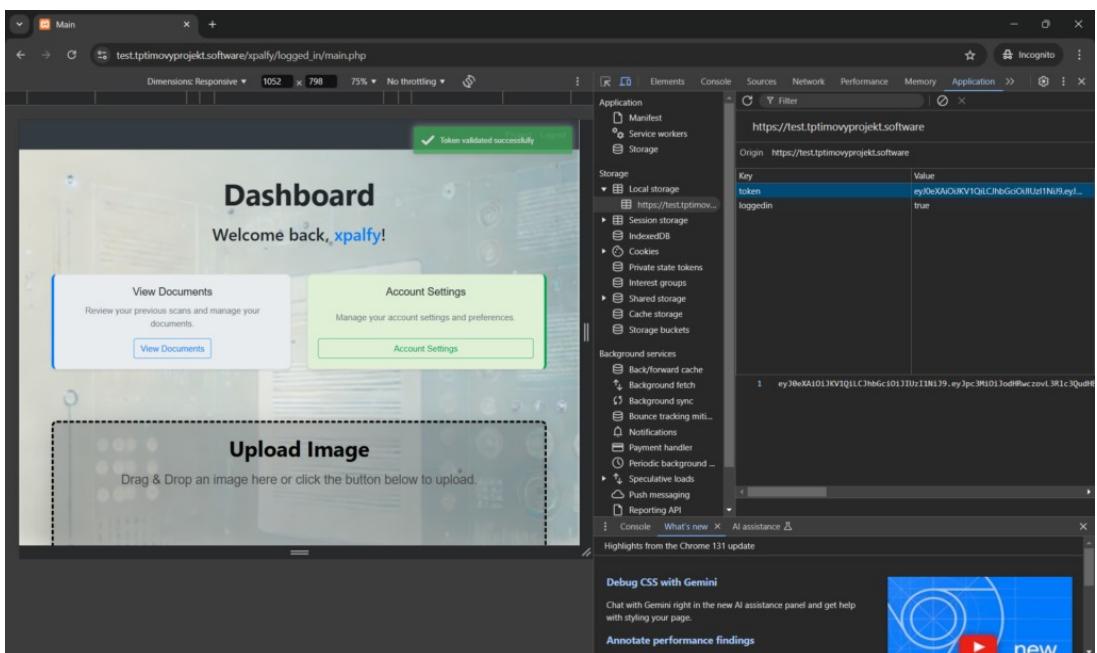
    $token = $matches[1];
    try {
        $secret = getJwtSecret();
        $decoded = JWT::decode($token, new Key($secret), 'HS256');

        if (
            $decoded->iss !== 'https://test.tptimovuprojekt.softeam.ru'
            || $decoded->aud !== 'https://test.tptimovuprojekt.softeam.ru'
        ) {
            throw new Exception('Invalid issuer or audience');
        }
    } catch (\Exception $e) {
        http_response_code(401);
        $_SESSION['toast'] = ['type' => 'error', 'message' => 'Unauthorized'];
        exit();
    }
}

if ($token) {
    $logged_in = true;
    $username = $data['username'];
    $data['logged_in'] = true;
    $data['username'] = $username;
    $data['toast'] = ['type' => 'success', 'message' => 'Token validated successfully'];
}

```

Obr. 55: Zdrojový kód JWT tokenov vo VSC



Obr. 56: Ukážka tokenu na webovej stránke

Bod programu č. 2: Plány na nasledujúci týždeň

Na nasledujúcej konzultácii sa plánujeme zameriť na **kompletnú analýzu doteraz dosiahnutých výsledkov**, ako aj na vyhodnotenie aktuálneho stavu implementácie, dokumentácie a používateľského rozhrania. Na základe tejto analýzy spoločne stanovíme konkrétné úlohy, ktorým sa budeme venovať počas zimných prázdnin.

Cieľom bude určiť oblasti, ktoré si vyžadujú opravu, doplnenie alebo optimalizáciu, a rozvrhnúť prácu tak, aby sme počas sviatočného obdobia mohli efektívne napredovať. Po konzultácii vypracujeme plán práce, ktorý nám pomôže udržať postup konzistentný a projekt posunúť smerom k úspešnému dokončeniu.

7.10 1. semester - 10. týždeň

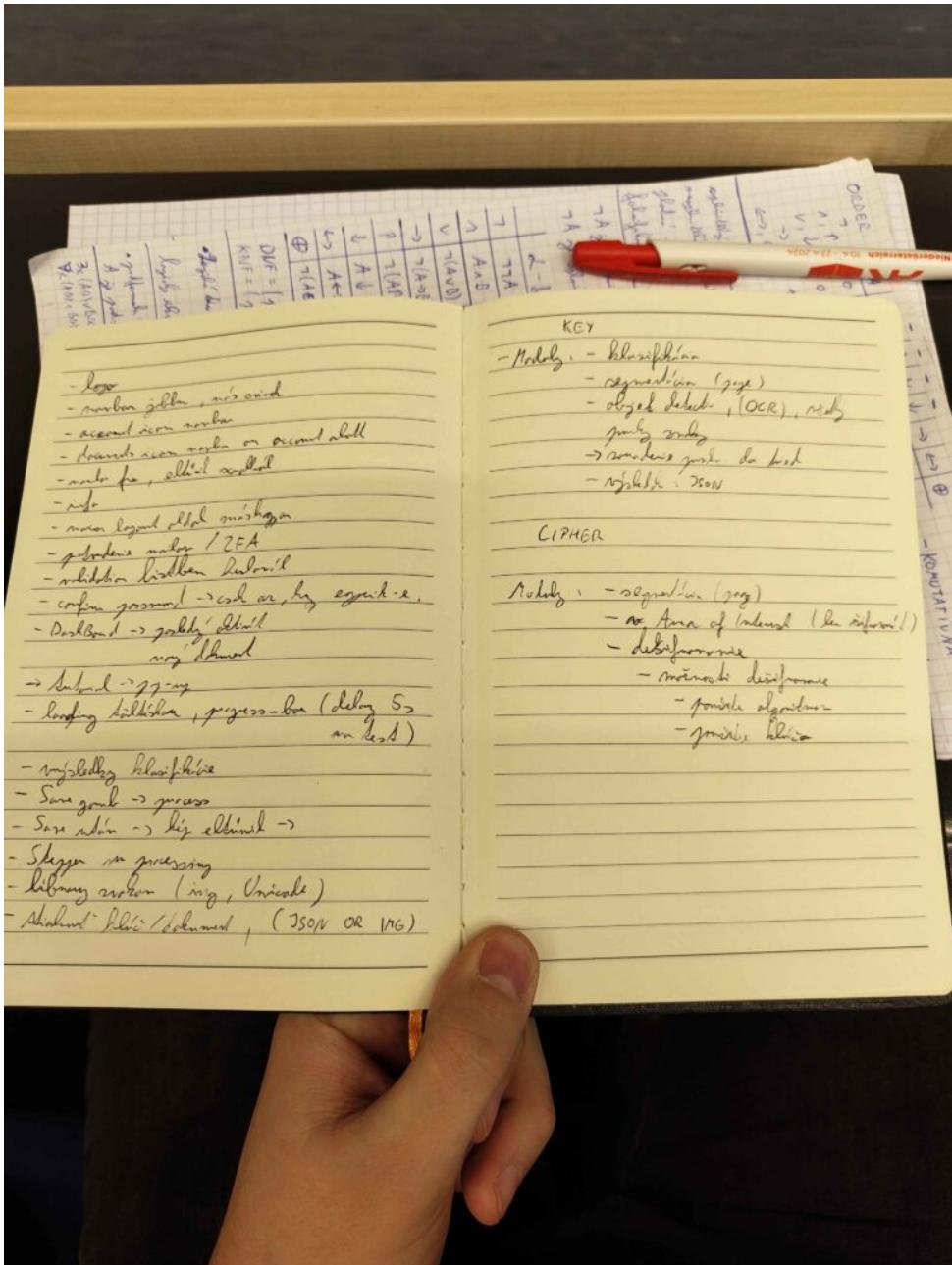
- **Dátum:** 03.12.2024, 09:00
- **Umiestnenie:** Slovenská technická univerzita v Bratislave, Fakulta elektrotechniky a informatiky
- **Miestnosť:** C517 (blok C, 5. poschodie)
- **Účastníci:**
 - Bc. Bence Both,
 - Bc. Matyas Horváth,
 - Bc. Jozef Nyitrai,
 - Bc. Áron Tükör,
 - Bc. Vincent Pálfy
- **Vedúci:** Ing. Stanislav Marochok

Bod programu č. 1: Analýza implementácií a plánovanie krokov na ich odstránenie počas zimných prázdnín

Na konzultácii sme sa rozprávali o tom, čo všetko musíme spraviť počas zimných prázdnín, aby sme úspešne dokončili projekt a splnili všetky požiadavky. Najskôr sme si ešte raz prešli hlavný cieľ projektu a čo od nás zadávateľ očakáva, aby sme mali jasno v tom, na čo sa máme zamerať.

Diskutovali sme o aktuálnych problémoch a chybách v našej implementácii. Presne sme si pomenovali, čo nefunguje, a dohodli sme sa na krokoch, ako to opraviť. Tiež sme hovorili o tom, ako by sa niektoré veci dali spraviť lepšie, aby bol celý systém efektívnejší.

Na konzultácii sme si zopakovali aj dôležité veci, o ktorých sme už hovorili na predchádzajúcich stretnutiach, a jasne sme si stanovili, čo všetko je potrebné spraviť. V ideálnom prípade chceme projektovú dokumentáciu dokončiť a úspešne ho odovzdať do konca prázdnin.



Obr. 57: Poznámky z konzultácií

7.11 2. semester - 1. týždeň

- **Dátum:** 04.03.2025, 12:00
- **Umiestnenie:** Slovenská technická univerzita v Bratislave, Fakulta elektrotechniky a informatiky
- **Miestnosť:** Discord
- **Účastníci:**
 - Bc. Bence Both,
 - Bc. Matyas Horváth,
 - Bc. Jozef Nyitrai,
 - Bc. Áron Tükör,
 - Bc. Vincent Pálfy
- **Vedúci:** Ing. Stanislav Marochok

Bod programu č. 1: Pokrok v implementácii a plánovanie ďalších krokov

Na stretnutí sme sa venovali hodnoteniu doterajšieho pokroku v implementácii a plánovaniu ďalších krokov na dokončenie projektu. Najprv sme si prešli aktuálny stav projektu a zhodnotili, ktoré úlohy sme už úspešne splnili a ktoré ešte treba dokončiť. Hlavne sme sa sústredili na úlohy, ktoré sme dostali na konci predchádzajúceho semestra.

Hovorili sme o problémoch, ktoré stále treba vyriešiť, a o možných riešeniach. Určili sme hlavné nedostatky, ktoré musíme odstrániť, a pridelili sme konkrétnie úlohy jednotlivým členom tímu. Dohodli sme sa na termínoch, dokedy by mali byť jednotlivé časti hotové.

Okrem opravy chýb sa budeme venovať aj zlepšeniu súčasných riešení, aby bol systém rýchlejší a spoľahlivejší. Pridáme tiež chýbajúce funkcie, ktoré sú potrebné na splnenie požiadaviek zadávateľa.

Dohodli sme sa aj na termínoch konzultácií v najbližšom období tak, aby vyhovovali všetkým členom tímu. Cieľom je zabezpečiť pravidelnú komunikáciu a efektívnu spoluprácu pri riešení úloh.

Na záver sme si zopakovali hlavné ciele projektu a uistili sa, že každý člen tímu má jasné úlohy a rozumie svojim povinnostiam. Naším cieľom je dokončiť projekt načas a v požadovanej kvalite.

	8:00 - 8:50	9:00 - 9:50	10:00 - 10:50	11:00 - 11:50	12:00 - 12:50	13:00 - 13:50	14:00 - 14:50	15:00 - 15:50	16:00 - 16:50	17:00 - 17:50
Pondelok	I-ADS (1,2,3,4,5)			I-KS (3,4,5)			I-PPDS (2,3,4,5)			I-AFJ (1,2,3,4,5)
Úterok				I-ADS (1,2,3,4,5)			I-AFJ (1,2,3,4,5)			I-KS (3,4,5)
Středa	I-PPDS (2,3,4,5)						I-BISPP (1,2)			
Štvrtok	I-BIOM (1,2,4,5)			I-SKS (1,2)			I-BIOM (1,2,4,5)			I-BISPP (1,2)
Piatok										
Preduška	Cvičenie		1	Bc. Bence Boháč						
			2	Bc. Matúš Honzátko						
			3	Bc. Jozef Nystrík						
			4	Bc. Áron Tukör						
			5	Bc. Vincent Pálfi						

Obr. 58: Rozvrh študentov v letnom semestri

7.12 2. semester - 2. týždeň

- **Dátum:** 11.03.2025, 12:00
- **Umiestnenie:** Slovenská technická univerzita v Bratislave, Fakulta elektrotechniky a informatiky
- **Miestnosť:** A208 (blok A, 2. poschodie)
- **Účastníci:**
 - Bc. Bence Both,
 - Bc. Matyas Horváth,
 - Bc. Jozef Nyitrai,
 - Bc. Áron Tükör,
 - Bc. Vincent Pálfy
- **Vedúci:** Ing. Stanislav Marochok

Bod programu č. 1: Pokrok v implementácii a priebežné zmeny

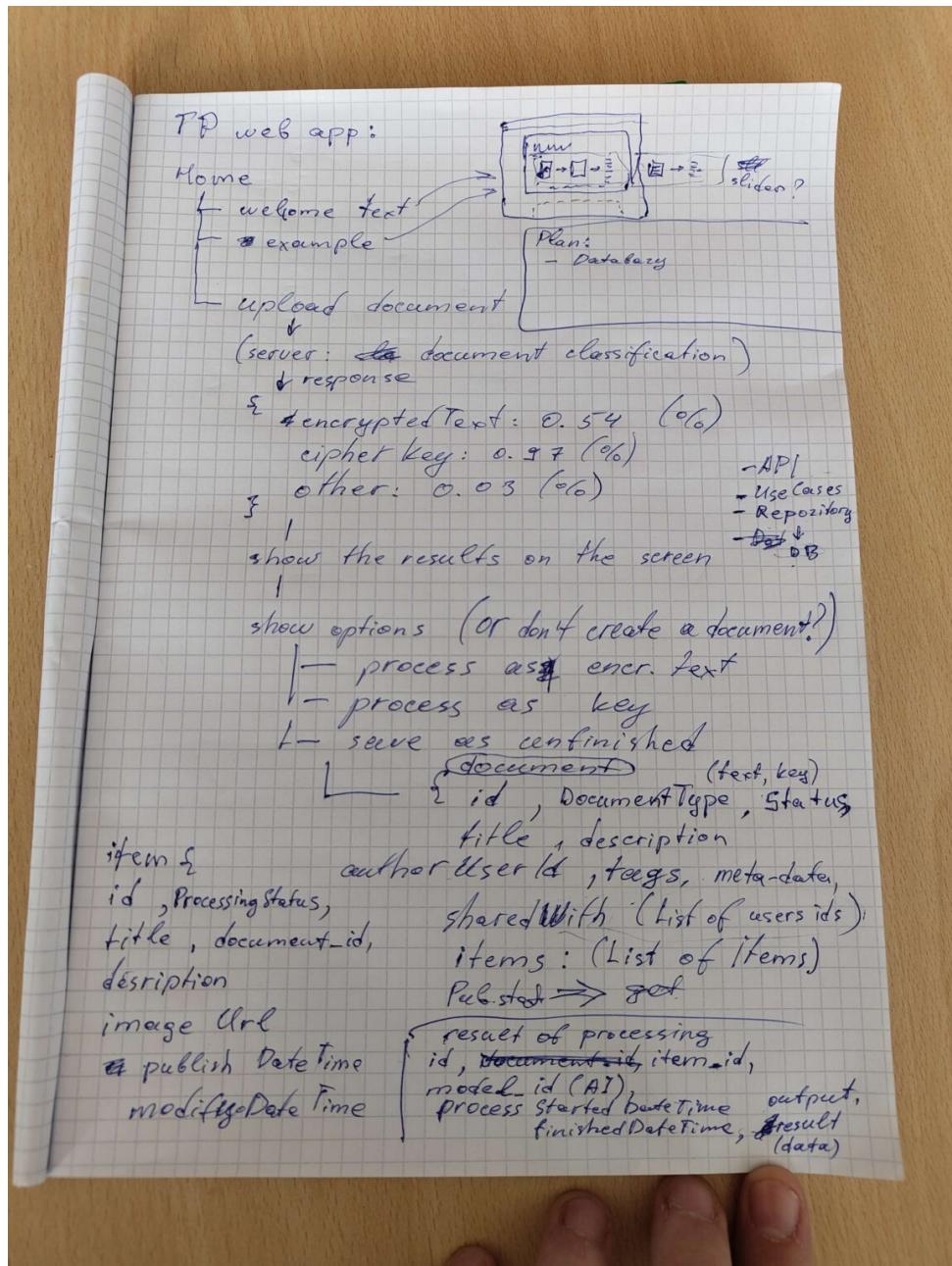
Počas posledného týždňa sme úspešne dokončili implementáciu **Flask** [28] backendu, ktorý teraz stabilne spracováva požiadavky používateľov. Flask je mikro webový framework v jazyku Python, ktorý sme zvolili namesto predchádzajúceho riešenia kvôli jeho jednoduchosti, flexibilite a nízkym nárokom na systémové zdroje.

Hlavným dôvodom prechodu na Flask bolo, že poskytuje **rýchle a efektívne API** na spracovanie požiadaviek, čo nám umožnilo jednoduchšiu správu dát a prípravu na budúcu integráciu klasifikačného modulu. Predchádzajúce riešenie malo problémy s výkonom a škálovatelnosťou, pričom Flask nám poskytuje väčšiu kontrolu nad spracovaním požiadaviek a lepšiu možnosť rozširovania funkcionálít.

Okrem toho sme prepracovali proces nahrávania dokumentov, aby sa po výbere súboru okamžite nahrával do **dočasného priečinka (temp)**. Aktuálne sa nahraný dokument ešte nekategorizuje, keďže modul klasifikácie zatiaľ neboli implementované, ale je plánovaný na ďalšiu fázu vývoja. Po jeho integrácii systém umožní automatické priradenie kategórie a hodnotenie dokumentov.

Ďalším dôležitým cieľom je zlepšenie backendu aj frontendu aplikácie. Na backendovej strane sa zameriame na **optimalizáciu výkonu a stabilitu systému**, zatiaľ čo frontend plánujeme zlepšiť tak, aby bol **intuitívnejší a vizuálne prehľadnejší**.

Podrobnosti plánovaných zmien sú uvedené v priloženom obrázku, ktorý dokumentuje architektúru a ďalšie plánované funkcionality systému:



Obr. 59: Poznámky z konzultácie

Bod programu č. 2: Plány na nasledujúci týždeň

Na stretnutí sme zhodnotili doterajší pokrok v implementácii a prediskutovali ďalšie kroky. Hlavným bodom diskusie bola integrácia Flask backendu a úprava procesu nahrávania a spracovania dokumentov. Prešli sme si aj možné vylepšenia aplikácie, ako aj identifikované problémy, ktoré by bolo potrebné riešiť.

Zamerali sme sa najmä na optimalizáciu rýchlosťi spracovania a architektúru projektu. Modul klasifikácie zatiaľ nie je úplne implementovaný, ale bude súčasťou budúcich fáz vývoja. Po jeho pridanej integrácii systém umožní automatické rozpoznávanie a kategorizáciu nahraných dokumentov. Diskutovali sme aj o ďalších plánovaných funkcionálitách a možnostiach rozšírenia aplikácie.

7.13 2. semester - 3. týždeň

- **Dátum:** 18.03.2025, 12:00
- **Umiestnenie:** Slovenská technická univerzita v Bratislave, Fakulta elektrotechniky a informatiky
- **Miestnosť:** A208 (blok A, 2. poschodie)
- **Účastníci:**
 - Bc. Bence Both,
 - Bc. Matyas Horváth,
 - Bc. Jozef Nyitrai,
 - Bc. Áron Tükör,
 - Bc. Vincent Pálfy
- **Vedúci:** Ing. Stanislav Marochok

Bod programu č. 1: Pokrok v implementácii a plánovanie ďalších krokov

V uplynulom týždni sme sa sústredili najmä na frontend aplikácie a dokumentáciu. Vzhľadom na obmedzený čas sme sa nevenovali výrazným zmenám v backendovej časti, ale uskutočnili sme drobné vylepšenia a pokračovali v práci na existujúcich funkciaalitách.

Jednou z hlavných tém diskusie bolo rozhodnutie, či aplikáciu vyvíjať ako **plne responzívnu**, alebo sa sústrediť primárne na **desktopovú verziu**. Zatiaľ sme v tomto smere neprijali konečné rozhodnutie a analyzujeme, aký prínos by responzivita mala pre používateľov a celkovú funkciaalitu aplikácie. Toto rozhodnutie ovplyvní nielen dizajn, ale aj celkovú architektúru projektu.

Okrem toho sme plánovali rozšírenie aplikácie o možnosť prihlásenia cez Google účet. Táto funkciaalita bude zlepšovať používateľský komfort a zvýši bezpečnosť prihlásovania. Implementácia je zatiaľ v plánovacej fáze, ale predpokladáme jej realizáciu v najbližších týždňoch.

Z predchádzajúceho týždňa zostávajú otvorené plány na integráciu klasifikačného modulu a ďalšie optimalizácie backendu aj frontendu. Nadalej sa budeme sústrediť na stabilitu a výkon aplikácie, pričom postupne rozširujeme jej funkciaalitu podľa potreby.

Podrobnosti o ďalšom postupe budú uvedené v nasledujúcej zápisnici podľa pokroku v implementácii.

7.14 2. semester - 4. týždeň

- **Dátum:** 25.03.2025, 12:30
- **Umiestnenie:** Slovenská technická univerzita v Bratislave, Fakulta elektrotechniky a informatiky
- **Miestnosť:** A208 (blok A, 2. poschodie)
- **Účastníci:**
 - Bc. Bence Both,
 - Bc. Matyas Horváth,
 - Bc. Jozef Nyitrai,
 - Bc. Áron Tükör,
 - Bc. Vincent Pálfy
- **Vedúci:** Ing. Stanislav Marochok

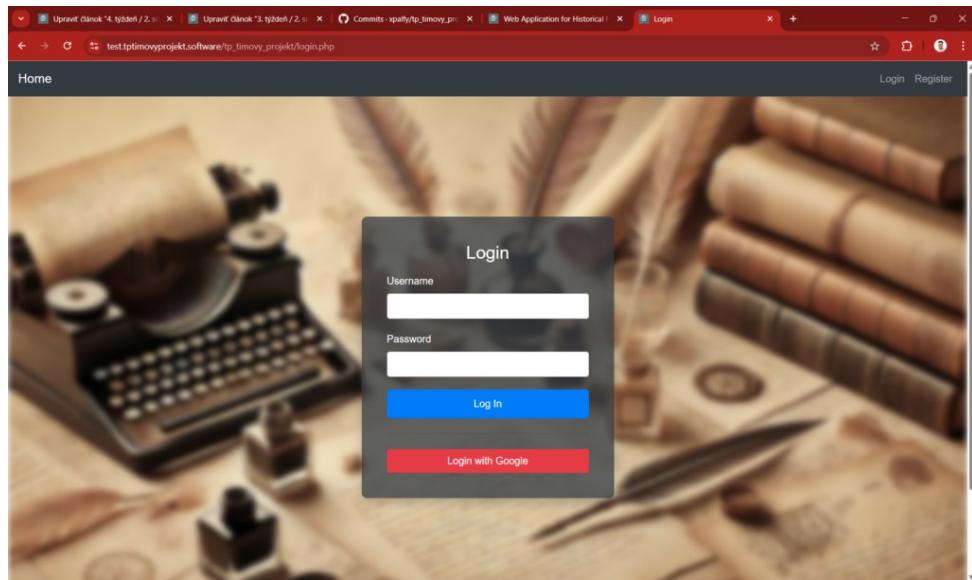
Bod programu č. 1: Pokrok v implementácii a priebežné zmeny

V tomto týždni sme zaznamenali výrazný pokrok vo viacerých oblastiach projektu. Najvýznamnejším prírastkom je implementácia funkcie **Google Login**, ktorá používateľom umožňuje pohodlné a bezpečné prihlásovanie do aplikácie prostredníctvom ich Google účtu. Táto funkcionalita zároveň zjednoduší správu používateľov a bude do budúcnosti slúžiť ako základ pre rôzne úrovne prístupových práv.

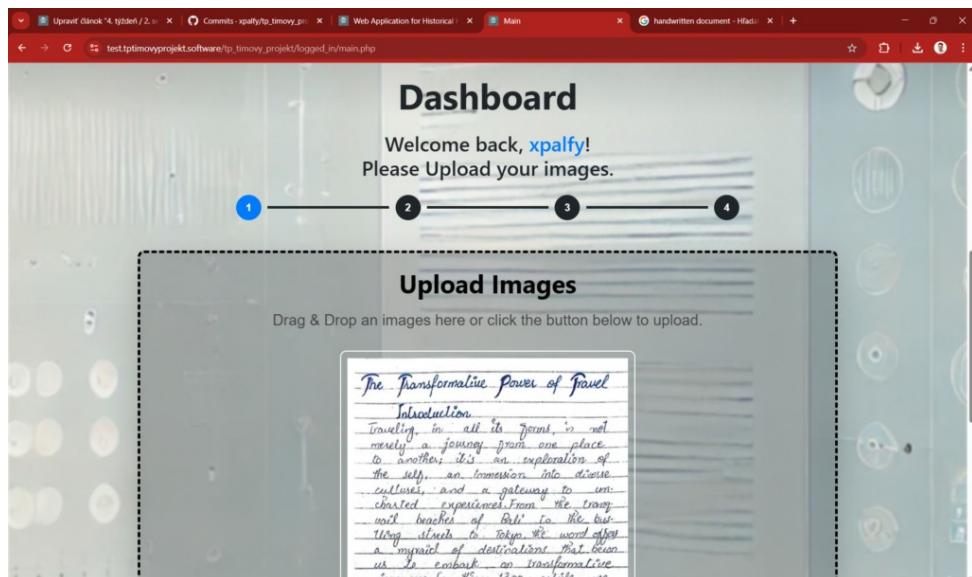
Ďalej bol nasadený **prvý krok spracovania obrazovky – klasifikácia**, ktorý zahŕňa výber obrázkov, ich nahranie a zobrazenie výsledkov pomocou vizuálneho indikátora progresu. Tento proces zatiaľ využíva testovacie (umelé) klasifikačné moduly, ktoré budú neskôr nahradené plnohodnotnými modelmi.

Nasledoval **druhý krok – segmentácia**, kde sme implementovali základný panel a vizualizáciu obdĺžníkov nad obrázkami. Opäť ide o testovaci verziu, ktorej cieľom je overiť správnu interakciu používateľa s výsledkami spracovania.

Frontend bol zároveň optimalizovaný z pohľadu **štýlov**, **tlačidiel**, **hlásení** a **uploadovania viacerých súborov naraz**. Pribudli **loading animácie**, lepšie **hlásenia o spracovaní a možnosti zdieľania dokumentov** s rôznymi úrovňami práv (zobrazenie/upravenie).



Obr. 60: Prihlásovacia stránka



Obr. 61: Webová aplikácia s ukážkou s nahraným obrazom

Bod programu č. 2: Plány na nasledujúci týždeň

V ďalšom týždni sa zameriame hlavne na vylepšenie vzhľadu a štruktúry používateľského rozhrania. Chceme, aby frontend pôsobil svetlejšie a modernejšie, a zároveň, aby bola aplikácia pre používateľov jednoduchšia na ovládanie. Plánujeme zjednodušiť navigáciu medzi jednotlivými krokmi spracovania dokumentov a zlepšiť celkový vizuálny štýl.

Dôležitou časťou budú aj ďalšie kroky po segmentácii obrázkov. Máme pripravený základ segmentácie, teraz chceme navrhnúť a začať implementovať ďalšie fázy, ako napríklad rozpoznávanie textu, pridávanie anotácií alebo iné formy analýzy historických dokumentov. Tento vývoj je súčasťou širšieho cieľa – vytvoriť webovú aplikáciu na spracovanie historických ručne písaných dokumentov.

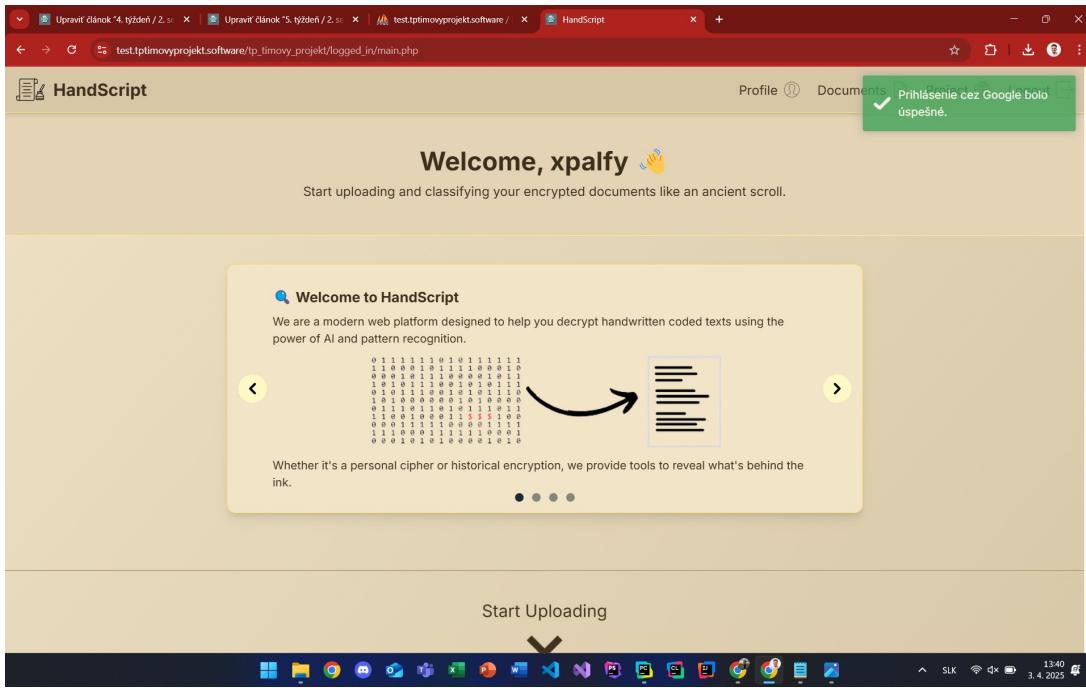
Ďalším plánom je pripraviť aplikáciu tak, aby fungovala ako **PWA (Progressive Web App)**. To znamená, že si ju používatelia budú môcť pridať do svojho zariadenia ako bežnú aplikáciu a používať ju aj offline, čo výrazne zlepší dostupnosť a používateľský zážitok.

7.15 2. semester - 5. týždeň

- **Dátum:** 01.04.2025, 12:00
- **Umiestnenie:** Slovenská technická univerzita v Bratislave, Fakulta elektrotechniky a informatiky
- **Miestnosť:** A208 (blok A, 2. poschodie)
- **Účastníci:**
 - Bc. Bence Both,
 - Bc. Matyas Horváth,
 - Bc. Jozef Nyitrai,
 - Bc. Áron Tükör,
 - Bc. Vincent Pálfy
- **Vedúci:** Ing. Stanislav Marochok

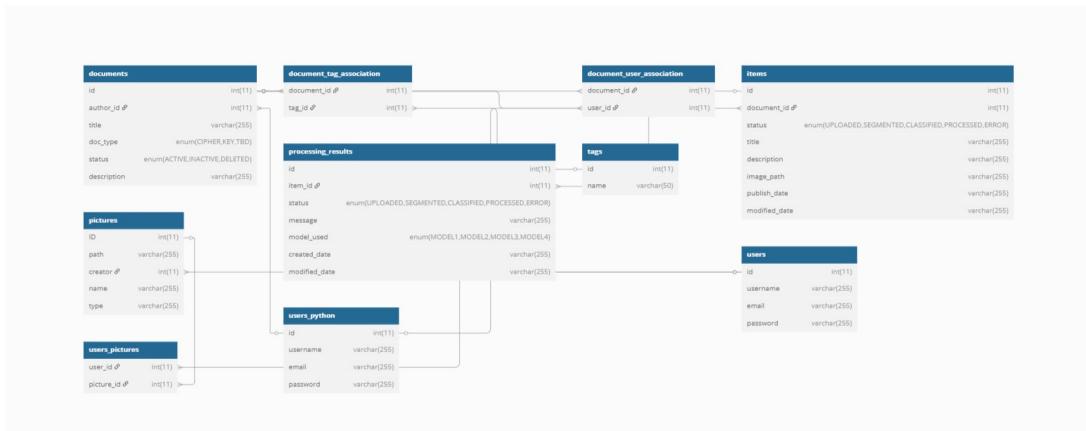
Bod programu č. 1: Pokrok v implementácii a priebežné zmeny

Tento týždeň sme sa sústredili najmä na **úpravu používateľského rozhrania a rozšírenie databázy**, čo predstavuje dôležitý krok vpred pre celý projekt. Jednou z hlavných zmien bol prechod na svetlý dizajn aplikácie, ktorý pôsobí modernejšie, čistejšie a je pre používateľov vizuálne prívetivejší. Na realizáciu tejto zmeny sme použili framework **TailwindCSS** [29], ktorý nám umožnil efektívne upravovať vzhľad komponentov, štýly tlačidiel, textových polí aj hlásení. Táto technológia zároveň priniesla väčšiu flexibilitu a konzistentnosť pri ďalšom vývoji frontendových častí.



Obr. 62: Zmena vo vizualizácii vo webovej aplikácii

Paralelne s vizuálnymi zmenami sme sa venovali aj architektonickým úpravám na strane databázy. Rozšírený model teraz lepšie odráža potreby aplikácie, pričom medzi najdôležitejšie prírastky patrí tabuľka `processing_results`, ktorá uchováva informácie o priebehu spracovania jednotlivých položiek. Zaznamenáva status, použitý model, hlásenia o chybách alebo výsledkoch, ako aj dátumy vytvorenia a úpravy. Vďaka tomu máme k dispozícii ucelený prehľad o tom, ako prebieha automatizované spracovanie dokumentov a obrázkov.



Obr. 63: Zmena v databáze

Zmeny sa dotkli aj prepojení medzi dokumentmi, používateľmi a tagmi – dokumenty je teraz možné efektívne zdieľať, priraďovať konkrétnym používateľom a označovať pomocou tagov. Tieto nové väzby rozširujú možnosti správy prístupových práv a organizácie obsahu. V databáze pribudla aj entita pre obrázky spolu s väzbou na používateľov, čo otvára možnosti pre budúce funkcie ako anotovanie, prehľad zmien alebo archiváciu historických záznamov. Všetky tieto úpravy tvoria základ pre nasledujúce etapy vývoja, vrátane pokročilej analýzy historických dokumentov, rozpoznávania textu a interaktívnej práce s obrazovým obsahom.

Bod programu č. 2: Plány na nasledujúci týždeň

V nasledujúcom týždni plánujeme pokračovať v prácach, ktoré sme nestihli v predchádzajúcom období. Prioritou bude rozšírenie nového svetlého dizajnu aj na zvyšné časti aplikácie, ktoré zatiaľ neboli vizuálne upravené. Cielom je dosiahnuť **jednotný vizuálny štýl** naprieč celým rozhraním a vytvoriť prehľadnejšie a používateľsky prívetivejšie prostredie. Implementácia bude aj nadalej prebiehať pomocou **TailwindCSS**, vďaka čomu si zachováme flexibilitu a konzistentnosť.

Zároveň sa budeme sústredovať na pokračovanie spracovania dokumentov po fáze segmentácie. V predchádzajúcich týždňoch sme úspešne pripravili základ segmentácie, preto sa teraz chceme zamerať na ďalšie etapy spracovania, konkrétnie na **rozpoznávanie textu (OCR)** zo segmentovaných oblastí, **pridávanie anotácií**, a prípravu podmienok pre **analýzu historických dokumentov**. Tieto kroky predstavujú dôležitý posun smerom k finálnemu cieľu – vytvoreniu komplexnej webovej aplikácie na spracovanie historických ručne písaných dokumentov.

7.16 2. semester - 6. týždeň

Konzultácie sa neuskutočnili z dôvodu konania Študentskej vedeckej odbornej činnosti (ŠVOČ), vďaka čomu mali študenti študentské voľno.

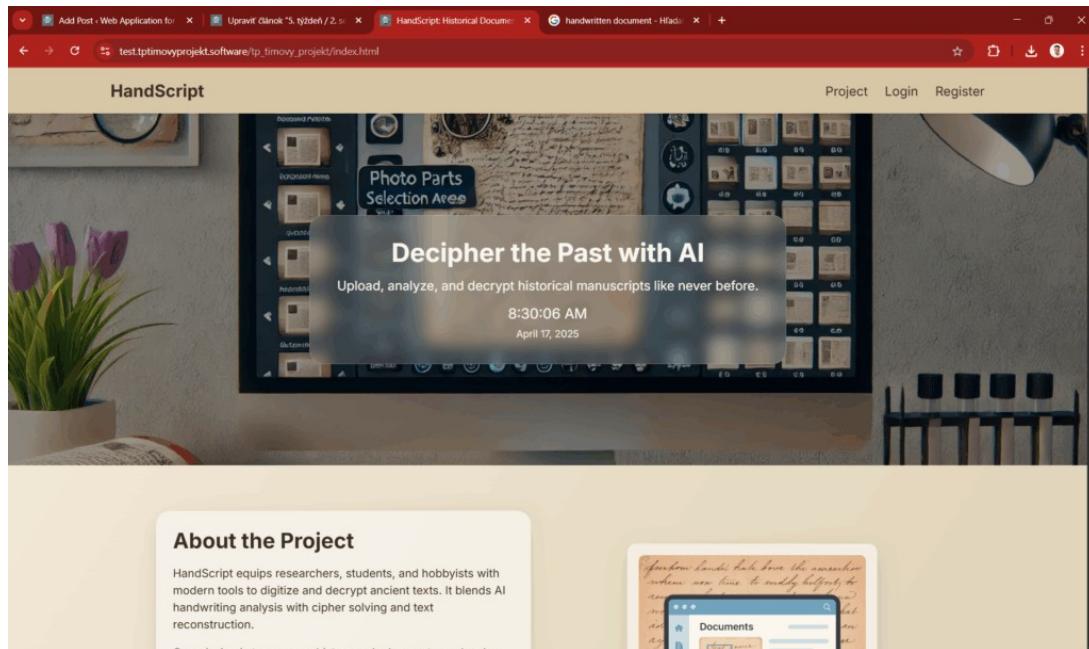
7.17 2. semester - 7. týždeň

- **Dátum:** 15.04.2025, 12:00
- **Umiestnenie:** Slovenská technická univerzita v Bratislave, Fakulta elektrotechniky a informatiky
- **Miestnosť:** A208 (blok A, 2. poschodie)
- **Účastníci:**
 - Bc. Bence Both,
 - Bc. Matyas Horváth,
 - Bc. Jozef Nyitrai,
 - Bc. Áron Tükör,
 - Bc. Vincent Pálfy
- **Vedúci:** Ing. Stanislav Marochok

Bod programu č. 1: Pokrok v implementácii a priebežné zmeny

V tomto týždni sme sa zamerali najmä na dokončenie vizuálnych úprav frontendovej časti aplikácie, ktoré sme začali v predchádzajúcim období. Na základe späťnej väzby a interného hodnotenia sme doplnili dizajn o jemné prechody (transition efekty), čím sme dosiahli interaktívnejší a modernejší vzhľad celej webovej aplikácie. Vďaka použitiu **TailwindCSS** sme mohli flexibilne upravovať jednotlivé komponenty, pričom sme dbali na konzistentnosť a vizuálnu čistotu celého používateľského rozhrania.

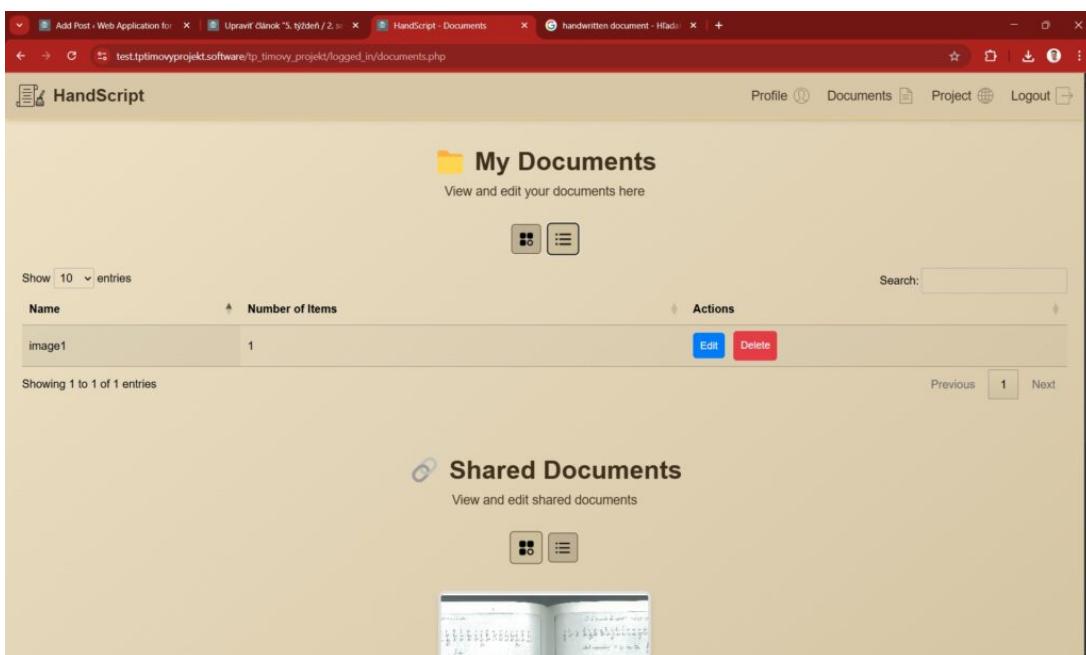
Okrem estetických úprav sme implementovali aj novú funkcia – možnosť prepínania medzi dvoma rôznymi pohľadmi na dokumenty: tabuľkovým a kartičkovým. Používateľ si tak môže zvoliť zobrazenie, ktoré mu vyhovuje – v jednom prípade má prehľadnejší výpis s počtom položiek a možnosťou rýchlej editácie, v druhom vizuálne atraktívnejšie zobrazenie s náhľadom dokumentu. Tieto zmeny významne zvyšujú komfort pri práci s dokumentmi, ako je vidieť na priložených ukážkach.



Obr. 64: Nový dizajn hlavnej stránky



Obr. 65: Nový dizajn prihlásovacej stránky



Obr. 66: Aktualizovaná stránka dokumentov

Bod programu č. 2: Plány na nasledujúci týždeň

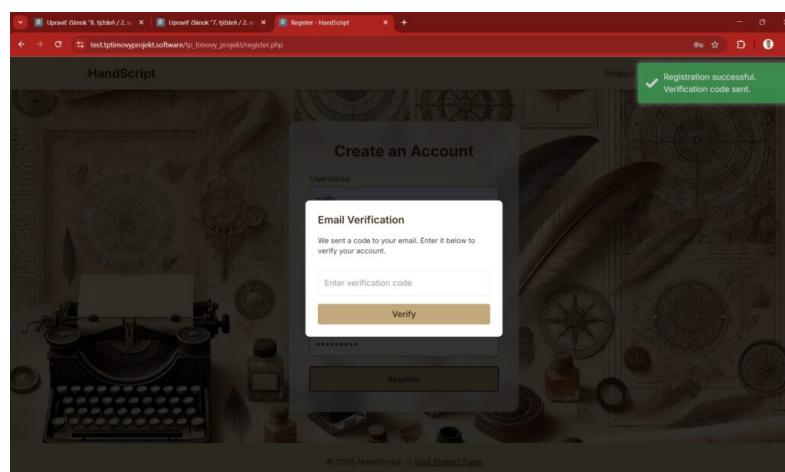
V nasledujúcim týždni sa presunieme späť k funkčnej časti aplikácie, konkrétnie k ďalšej fáze spracovania dokumentov. Zameriame sa na pokračovanie po fáze segmentácie, ktorú sme už pripravili. Prioritne sa budeme venovať implementácii rozpoznávania textu (**OCR**) zo segmentovaných oblastí a možnosti anotovania výstupov. Naším cieľom je posilniť funkcionality aplikácie tak, aby sme sa priblížili k finálnej podobe systému určeného na efektívne a interaktívne spracovanie historických ručne písaných dokumentov.

7.18 2. semester - 8. týždeň

- **Dátum:** 22.04.2025, 12:00
- **Umiestnenie:** Slovenská technická univerzita v Bratislave, Fakulta elektrotechniky a informatiky
- **Miestnosť:** A208 (blok A, 2. poschodie)
- **Účastníci:**
 - Bc. Bence Both,
 - Bc. Matyas Horváth,
 - Bc. Jozef Nyitrai,
 - Bc. Áron Tükör,
 - Bc. Vincent Pálfy
- **Vedúci:** Ing. Stanislav Marochok

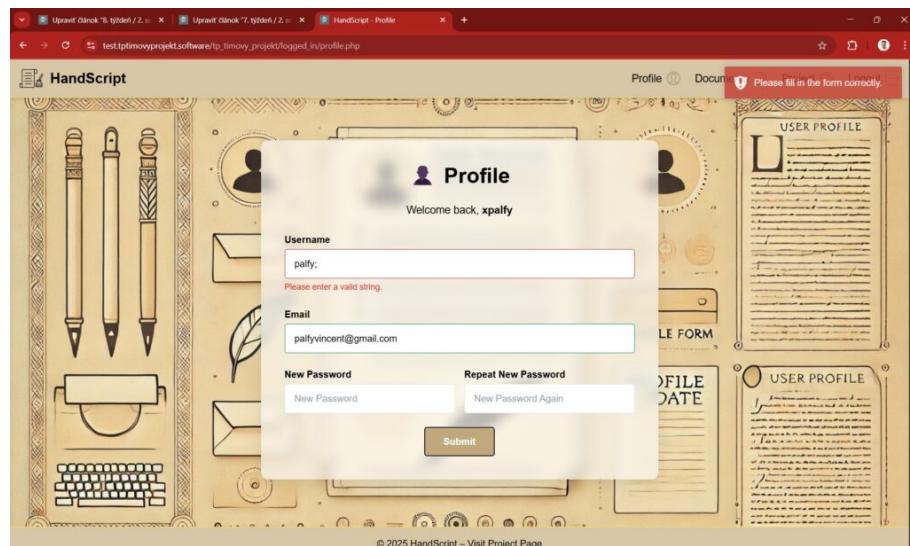
Bod programu č. 1: Pokrok v implementácii a priebežné zmeny

Počas tohto týždňa sme sa zamerali na implementáciu overenia emailovej adresy pri registrácii a prihlásení do systému. Zavedením šestciferného overovacieho kódu sme zabezpečili, že registráciu môžu vykonať iba používatelia, ktorí vlastnia platnú emailovú adresu a sú schopní ju overiť. Tento krok výrazne zvyšuje bezpečnosť a spoločnosť používateľskej bázy.



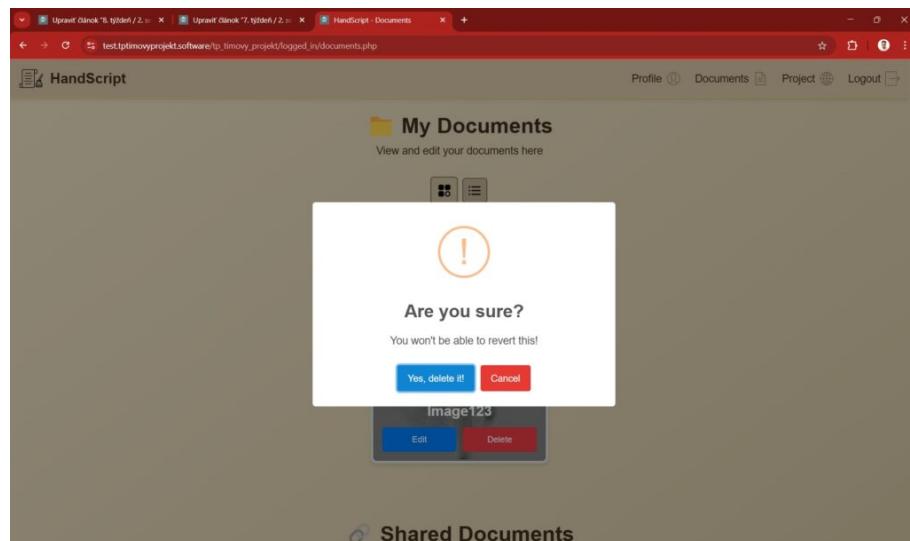
Obr. 67: Overenie e-mailovej adresy

Okrem toho sme sa venovali opravám a spresneniu funkcionality **úpravy profilových údajov**. Odstránili sme identifikované chyby a zabezpečili sme presnejšie a stabilnejšie správanie formulárov pri zmene používateľských údajov.



Obr. 68: Podstránka upravenia informácií o používateľovi

Významným pokrokom bolo aj rozšírenie funkcionality aplikácie o možnosť správy obrázkov na základe princípov **CRUD** (Create, Read, Update, Delete). Používatelia tak teraz môžu efektívne nahrávať, zobrazovať, upravovať a mazať obrázky v rámci aplikácie, čo vytvára pevný základ pre budúce rozšírenie systému o prácu s vizuálnym obsahom.



Obr. 69: Príklad operácie CRUD

Bod programu č. 2: Plány na nasledujúci týždeň

V nasledujúcim týždni zostáva našou prioritou pokračovanie v práci na funkcionálite spracovania dokumentov, konkrétnie nadvázovanie na predchádzajúcu fázu segmentácie. Hlavným cieľom bude implementácia rozpoznávania textu (OCR) z jednotlivých segmentov a pridanie možnosti anotovania získaných výstupov. Súčasťou plánovaných úloh je aj vytvorenie prostredia na ukladanie rozpoznaných znakov – buď vo forme JSON výstupov, alebo priamo do databázy.

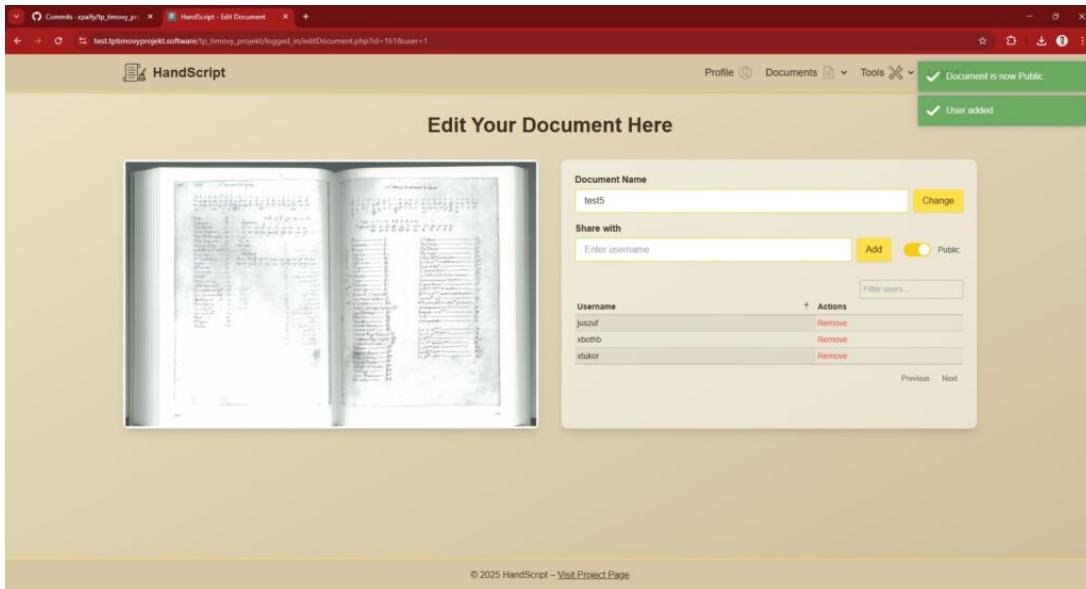
Zameriame sa predovšetkým na prípravu príslušných backendových endpointov a frontendových komponentov, ktoré umožnia základné spracovanie, zobrazenie a prácu s týmito údajmi. Samotná logika rozpoznávania, segmentácie či klasifikácie obrázkov však nepatrí do rozsahu našich úloh – naším cieľom nie je vyvíjať tieto moduly, ale zabezpečiť funkčné a flexibilné prostredie, v ktorom ich bude možné integrovať a používať.

7.19 2. semester - 9. týždeň

- **Dátum:** 29.04.2025, 12:00
- **Umiestnenie:** Slovenská technická univerzita v Bratislave, Fakulta elektrotechniky a informatiky
- **Miestnosť:** A208 (blok A, 2. poschodie)
- **Účastníci:**
 - Bc. Bence Both,
 - Bc. Matyas Horváth,
 - Bc. Jozef Nyitrai,
 - Bc. Áron Tükör,
 - Bc. Vincent Pálfy
- **Vedúci:** Ing. Stanislav Marochok

Bod programu č. 1: Pokrok v implementácii a priebežné zmeny

Tento týždeň sme dosiahli viacero významných pokrokov v rámci správy dokumentov a spracovania vizuálnych údajov. Zaviedli sme možnosť rozlišovať medzi tromi typmi dokumentov: **Public**, **Private** a **Shared**. Používatelia teraz môžu flexibilne nastavovať zdieľanie dokumentov – dokumenty je možné publikovať, zdieľať s konkrétnymi používateľmi alebo nastavenia zdieľania kedykoľvek odvolať. Tieto zmeny sa vykonávajú v editovacom rozhraní dokumentu, ktoré sme prispôsobili na jednoduchú správu týchto nastavení.



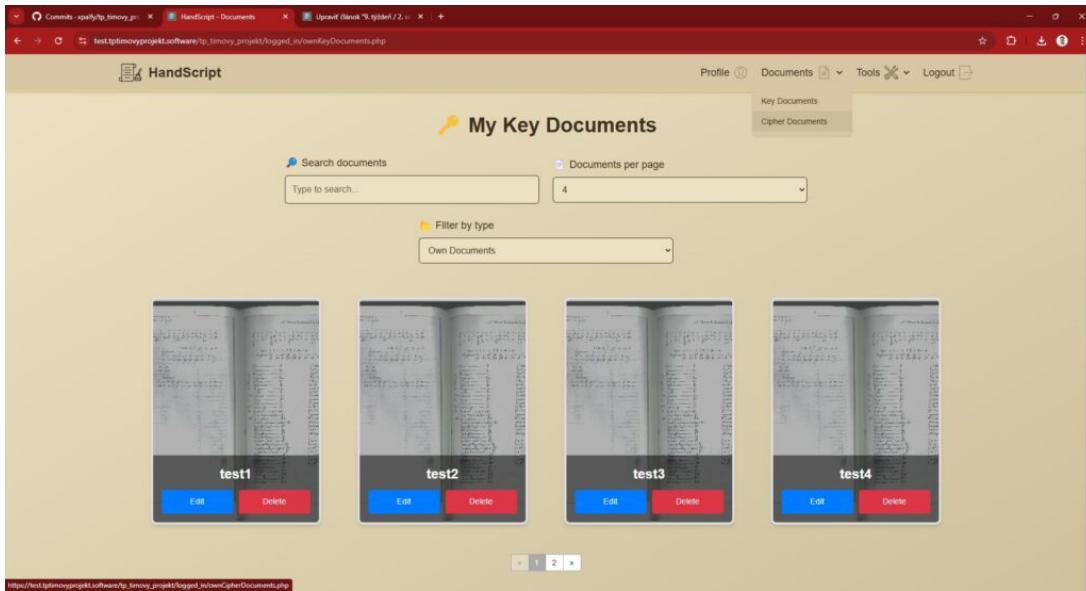
Obr. 70: Podstránka zobrazujúce upravenia dokumentov

Zásadne sme vylepšili aj prehliadanie dokumentov. Používatelia môžu filtrovať dokumenty podľa názvu alebo typu, čím sme zlepšili prehľadnosť systému. Dokumenty typu **CIPHER** a **KEY** sú teraz zobrazované na samostatných podstránkach, čo prispieva k lepšej orientácii v obsahu.

Zaviedli sme aj rozdiely v možnostiach úprav jednotlivých typov dokumentov:

- **Vlastné dokumenty** možno upravovať aj mazať. Ak je dokument zdieľaný,
- **Zdieľané dokumenty** možno buď odstrániť z vlastného účtu, alebo zrušiť zdieľanie.
- **Public dokumenty** možno iba zobraziť a používať – nie je ich možné upravovať ani mazať.

V oblasti spracovania obrázkov sme pokračovali v oddelení logiky pre spracovanie **CIPHER** a **KEY** dokumentov. Implementovali sme extrakciu znakov zo súborov a ich uloženie vo forme **JSON** výstupov. Pre **KEY** dokumenty sa extrahuje zoznam znakov s ich kódmi, zatiaľ čo **CIPHER** dokumenty vracajú čitateľný text.



Obr. 71: Dokumenty používateľa rozdelené podľa typu - klúče a šifrované texty

Bod programu č. 2: Plány na nasledujúci týždeň

Na nasledujúci týždeň plánujeme dokončiť poslednú fázu spracovania dokumentov – **dekódovanie CIPHER textov pomocou vybraného KEY dokumentu**. Používateľ si bude môcť zvoliť jeden z dostupných KEY dokumentov a na jeho základe dekódovať text obsiahnutý v CIPHER dokumente. Tento proces bude využívať už existujúce JSON výstupy.

Okrem toho plánujeme vytvoriť **úvodnú stránku pre používateľov**, ktorá bude obsahovať stručný text a video s vysvetlením, ako systém funguje. Cieľom je poskytnúť novým používateľom rýchlu orientáciu a lepšie porozumenie možnostiam webovej aplikácie.

7.20 2. semester - 10. týždeň

- **Dátum:** 06.05.2025, 12:00
- **Umiestnenie:** Slovenská technická univerzita v Bratislave, Fakulta elektrotechniky a informatiky
- **Miestnosť:** A208 (blok A, 2. poschodie)
- **Účastníci:**
 - Bc. Bence Both,
 - Bc. Matyas Horváth,
 - Bc. Jozef Nyitrai,
 - Bc. Áron Tükör,
 - Bc. Vincent Pálfy
- **Vedúci:** Ing. Stanislav Marochok

Bod programu č. 1: Pokrok v implementácii a priebežné zmeny

Tento týždeň sme sa zamerali na viaceré oblasti vylepšenia používateľského rozhrania a posilnenia podpory pre nových aj existujúcich používateľov. Významným prínosom je implementácia novej stránky **FAQ**, ktorá ponúka prehľadné odpovede na najčastejšie otázky týkajúce sa používania systému. Táto stránka je interaktívna – používateľ si môže jednotlivé témy rozkliknúť, čím sa mu zobrazia podrobnejšie informácie. Navyše sme pripravili aj tutoriálové video, ktoré bude priamo dostupné na FAQ stránke a poskytne používateľom vizuálny sprievod systémom a jeho funkciami. Cieľom tejto novinky je zjednodušiť prvý kontakt so systémom a zvýšiť celkovú dostupnosť informácií.

Ďalšou novinkou je modálne okno informujúce používateľa o používaní súborov **cookie**, ktoré sme pridali v súlade s požiadavkami na ochranu osobných údajov a transparentnosť voči používateľom. Táto funkcia sa zobrazuje pri **prvej návštive** stránky a zabezpečuje informovaný súhlas s ukladaním údajov.

Z pohľadu navigácie a orientácie v dokumentoch sme uskutočnili vizuálne úpravy v rozložení pohľadu na dokumenty. Filtračné možnosti boli premiestnené do samostatného **ľavého panelu (aside)**, čo zlepšuje čitateľnosť rozhrania a umožňuje používateľom rýchlejšie nájsť požadované dokumenty podľa typu alebo názvu.

The screenshot shows a section titled "Questions" on the HandScript website. It displays five expandable dropdown boxes containing frequently asked questions. Below the questions is a navigation bar with two items: "1" and "2". Underneath this is a form titled "Ask a Question" with a text input field and a "Submit" button.

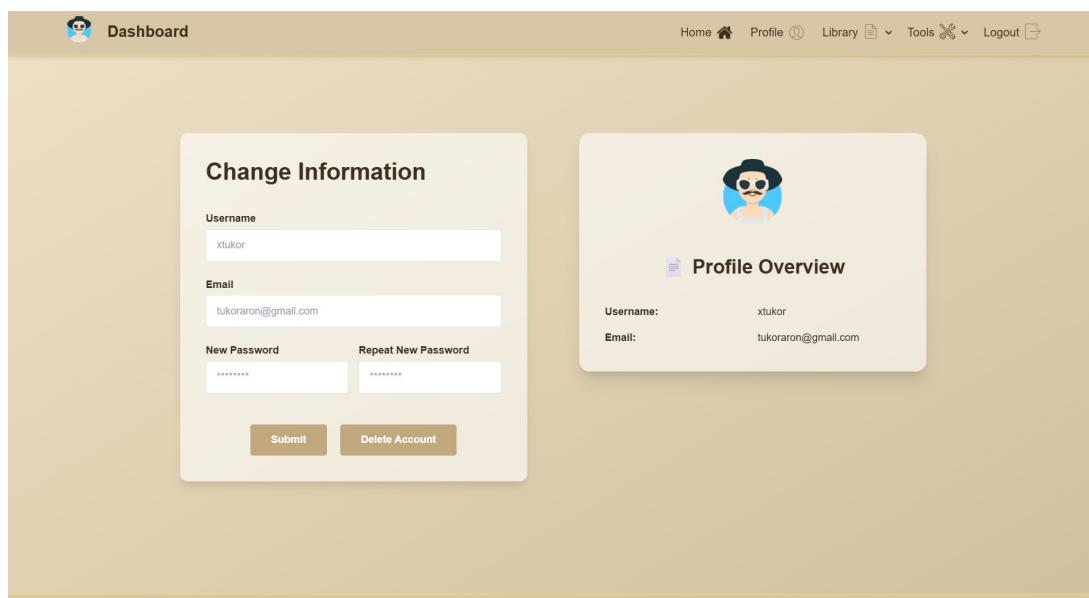
- Can the app handle multiple languages or old scripts?
- Is the processing done locally or in the cloud?
- How accurate is the keyword extraction?
- What document formats does it support?
- Do I need any technical skills to use the app?

Obr. 72: Podstránka často kladených otázok

The screenshot shows the HandScript homepage. A prominent cookie consent dialog box is displayed in the center. The dialog title is "We Use Cookies" and contains the text: "This website uses cookies to ensure you get the best experience on our platform." It includes two buttons: "Accept Cookies" and "Decline". The background of the page shows a dark-themed interface with a lamp and a computer monitor displaying a historical manuscript analysis tool.

Obr. 73: Akceptácia uloženie cookies

Pre zlepšenie personalizácie systému sme každému používateľovi automaticky priradili **avatar** na základe **hashovanej e-mailovej adresy**. Tieto avatary sa následne zobrazujú v rozhraní a pomáhajú rýchlejšie identifikovať používateľov, napríklad pri zdieľaných dokumentoch alebo vo verejných zoznamoch. Zároveň sme rozšírili stránku na aktualizáciu profilu, ktorá teraz obsahuje dve prehľadné panely – v ľavej časti sa zobrazujú aktuálne používateľské údaje, ako meno, e-mail a avatar, zatiaľ čo v pravej časti môže používateľ tieto údaje priamo upraviť. Táto zmena prispieva k prehľadnejšiemu spracovaniu informácií a umožňuje rýchlu orientáciu pri úprave osobného profilu.



Obr. 74: Nová vylepšená verzia podstránky profilu

V neposlednom rade sme dokončili novú verziu zobrazenia pre dokumenty typu **CIPHER**, ktorá je teraz vizuálne aj funkčne zjednotená so zobrazením dokumentov typu **KEY**. Umožňuje jednoduchšie prehliadanie, filtrovanie a prácu s obsahom šifrovaných údajov.

Bod programu č. 2: Plány na nasledujúci týždeň

V priebehu nasledujúceho týždňa sa zameriame na finalizáciu back-endovej časti projektu, najmä na implementáciu unit testov, ktoré overia správnosť a stabilitu kľúčových komponentov systému. Súčasne plánujeme dokončiť posledný krok v spracovaní dokumentov – dekódovanie CIPHER textov pomocou vybraného KEY dokumentu. Táto funkcionálita bude vychádzať z už vytvorených JSON výstupov a uzavrie celý proces spracovania vizuálnych údajov.

Popri technickej implementácii budeme tiež pracovať na finalizácii dokumentácie systému a záverečnej príprave tutoriálového videa. Cieľom je zabezpečiť, aby boli všetky informácie o používaní aplikácie zrozumiteľné, prístupné a dobre zdokumentované pre všetkých používateľov.

8 Zoznam hotových stránok

Na uvedenom odkaze je možné náš celý projekt zobraziť: https://github.com/xpalfy/tp_timovy_projekt

Nižšie je uvedený zoznam hotových webových stránok, ktoré sme počas práce vytvorili:

- https://test.tptimovyprojekt.software/tp_timovy_projekt/
- https://test.tptimovyprojekt.software/tp_timovy_projekt/task.php
- https://test.tptimovyprojekt.software/tp_timovy_projekt/team.php
- https://test.tptimovyprojekt.software/tp_timovy_projekt/faq.php
- <https://python.tptimovyprojekt.software/apidocs/>
- https://test.tptimovyprojekt.software/tp_timovy_projekt/login.php
- https://test.tptimovyprojekt.software/tp_timovy_projekt/register.php
- <https://tptimovyprojekt.ddns.net/>
- https://test.tptimovyprojekt.software/tp_timovy_projekt/logged_in/main.php
- https://test.tptimovyprojekt.software/tp_timovy_projekt/logged_in/profile.php
- https://test.tptimovyprojekt.software/tp_timovy_projekt/logged_in/openKeyDocuments.php
- https://test.tptimovyprojekt.software/tp_timovy_projekt/logged_in/openCipherDocuments.php
- https://test.tptimovyprojekt.software/tp_timovy_projekt/logged_in/modules/segmentModule.php
- https://test.tptimovyprojekt.software/tp_timovy_projekt/logged_in/modules/analyzeModule.php
- https://test.tptimovyprojekt.software/tp_timovy_projekt/logged_in/modules/lettersModule.php

- https://test.tptimovyprojekt.software/tp_timovy_projekt/logged_in/modules/editJsonModule.php
- https://test.tptimovyprojekt.software/tp_timovy_projekt/logged_in/modules/decipherModule.php

Záver

Počas tohto roka sme mali pravidelné týždenné konzultácie, na ktorých sme plánovali a následne pracovali na jednotlivých častiach projektu. Táto skúsenosť nám poskytla cenný pohľad na to, ako môže väčší a dlhodobejší projekt prebiehať v reálnom svete.

Získali sme tiež skúsenosti s webhostingom, prácou na front-ende a back-ende webových aplikácií a zároveň sme sa naučili, ako fungujú niektoré historické šifry. Tieto poznatky nám pomohli lepšie pochopiť teoretické aj praktické aspekty vývoja a bezpečnosti aplikácií.

Zoznam použitej literatúry

1. Dostupné tiež z: <https://www.transkribus.org/>.
2. Dostupné tiež z: <https://readcoop.org/>.
3. Dostupné tiež z: <https://de-crypt.org/>.
4. Dostupné tiež z: <https://www.raspberrypi.com/products/raspberry-pi-3-model-b/>.
5. Dostupné tiež z: <https://wordpress.com/>.
6. Dostupné tiež z: <https://www.raspberrypi.com/software/operating-systems/>.
7. Dostupné tiež z: <https://www.raspberrypi.com/news/raspberry-pi-imager-imaging-utility/>.
8. Dostupné tiež z: <https://httpd.apache.org/>.
9. Dostupné tiež z: <https://www.php.net/>.
10. Dostupné tiež z: <https://mariadb.org/>.
11. Dostupné tiež z: <https://jwt.io/>.
12. Dostupné tiež z: https://swagger.io/docs/specification/v3_0/authentication/bearer-authentication/.
13. Dostupné tiež z: <https://github.com.firebaseio/php-jwt>.
14. Dostupné tiež z: <https://oauth.net/2/>.
15. Dostupné tiež z: <https://www.apachefriends.org/>.
16. Dostupné tiež z: <https://html.com/>.
17. Dostupné tiež z: <https://www.w3schools.com/css/default.asp>.
18. Dostupné tiež z: <https://www.javascript.com/>.
19. Dostupné tiež z: <https://getbootstrap.com/docs/5.3/getting-started/introduction/>.
20. Dostupné tiež z: <https://jquery.com/>.
21. Dostupné tiež z: <https://git-scm.com/>.
22. Dostupné tiež z: <https://www.cloudflare.com/>.
23. Dostupné tiež z: <https://developers.cloudflare.com/cloudflare-one/connections/connect-networks/>.

24. Dostupné tiež z: <https://www.python.org/>.
25. Dostupné tiež z: <https://www.overleaf.com/>.
26. Dostupné tiež z: <https://www.latex-project.org/>.
27. Dostupné tiež z: <https://www.figma.com/>.
28. Dostupné tiež z: <https://flask.palletsprojects.com/en/stable/>.
29. Dostupné tiež z: <https://tailwindcss.com/>.

Prílohy

A Štruktúra elektronického nosiča	II
---	----

A Štruktúra elektronického nosiča

TP_2025_documentation.pdf

- pdf súbor dokumentácie tímovej práce

/images

- priečinok s funkčnými obrázkami

/cipher.png

- obrázok šifrovaného textu

/key.png

- obrázok klúča