

# **Webová aplikace pro správu týmových projektů**

Evgenii Ponomarev

---

Bakalářská práce  
2025



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
Ústav počítačových a komunikačních systémů

Akademický rok: 2024/2025

# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Evgenii Ponomarev**  
Osobní číslo: **A22460**  
Studijní program: **B0688A140008 Informační technologie v administrativě**  
Forma studia: **Prezenční**  
Téma práce: **Webová aplikace pro správu týmových projektů**  
Téma práce anglicky: **Web Application for Team Project Management**

## Zásady pro vypracování

- Provedte rešerši existujících webových aplikací pro správu týmových projektů.
- Vypracujte stručný rozbor technologií a frameworků, které budou použity pro vývoj aplikace.
- Provedte analýzu a rozbor požadavků na zvolené řešení.
- Zpracujte prototyp aplikace na základě výsledků analýzy.
- Věnujte pozornost zabezpečení webové aplikace.

Forma zpracování bakalářské práce: **tištěná/elektronická**

**Seznam doporučené literatury:**

1. FOWLER, Martin. Patterns of Enterprise Application Architecture. Boston: Addison-Wesley, 2002. ISBN 0-321-12742-0
2. MUNZNER, Tamara. Visualization analysis and design. 2nd ed. Boca Raton: CRC Press, 2014. ISBN 978-1-4987-5971-7
3. NARAMORE, Elizabeth. Vytváříme webové aplikace v PHP5, MySQL a Apache. Praha: Grada Publishing, 2009. ISBN 978-80-251-1073-7
4. LUBBERS, Peter. HTML5 – Programujeme moderní webové aplikace. Praha: Computer Press, 2011. ISBN 978-80-251-3539-6
5. DUCKETT, Jon. HTML and CSS: Design and Build Websites. Hoboken: John Wiley, 2011. ISBN 978-1-118-00818-8

Vedoucí bakalářské práce: **doc. Ing. Petr Šilhavý, Ph.D.**  
Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce: **4. prosince 2024**  
Termín odevzdání bakalářské práce: **9. června 2025**

**doc. Ing. Jiří Vojtěšek, Ph.D. v.r.**  
děkan



**Ing. Miroslav Matýsek, Ph.D. v.r.**  
ředitel ústavu

Ve Zlíně dne 4. prosince 2024

# PROHLÁŠENÍ AUTORA BAKALÁŘSKÉ PRÁCE

## Beru na vědomí, že

- odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., v platném znění bez ohledu na výsledek obhajoby;
- bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému a bude dostupná k nahlédnutí;
- jedno vyhotovení bakalářské práce v listinné podobě bude ponecháno Univerzitě Tomáše Bati ve Zlíně k uložení;
- na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- podle § 60 odst. 1 autorského zákona má Univerzita Tomáše Bati ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- podle § 60 odst. 2 a 3 mohu užít své dílo – bakalářskou práci – nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tj. k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá; neodevzdání této součásti může být důvodem k neobhájení práce.

## Prohlašuji, že

- jsem na bakalářské práci pracoval(a) samostatně a použitou literaturu jsem řádně citoval(a); v případě publikace výsledků budu uveden(a) jako spoluautor;
- odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou obsahově totožné.

Ve Zlíně, dne .....

.....  
podpis autora



## **ABSTRAKT**

Hlavním cílem této bakalářské práce je vytvoření webové aplikace pro správu týmových projektů. Během práce bude studovány další webové aplikace, aby bylo možné porozumět hlavním funkcím požadovaným v takových aplikacích. Bude vyvinut prototyp webové aplikace, se kterou lze dále pracovat. Bude přidány klíčové funkce a pozornost bude věnována ochraně proti úniku dat. Pro tvorbu budou použity značkovací jazyky HTML a CSS, programovací jazyky PHP a JavaScript a systém pro správu databází MySQL.

Klíčová slova: webová aplikace, týmové projekty, HTML, CSS, PHP, JavaScript, MySQL

## **ABSTRACT**

The main goal of this bachelor's thesis is to create a web application for managing team projects. During the work, other web applications will be studied in order to understand the main functions required in such applications. A prototype of a web application will be developed, which can be further worked with. Key functions will be added and attention was paid to protection against data leakage. The markup languages HTML and CSS, the programming languages PHP and JavaScript, and the MySQL database management system will be used for the creation.

Keywords: web application, team projects, HTML, CSS, PHP, JavaScript, MySQL

# OBSAH

|                                                                         |           |
|-------------------------------------------------------------------------|-----------|
| <b>ÚVOD.....</b>                                                        | <b>11</b> |
| <b>I TEORETICKÁ ČÁST .....</b>                                          | <b>12</b> |
| <b>1 WEBOVÁ APLIKACE.....</b>                                           | <b>13</b> |
| 1.1 DEFINICE .....                                                      | 13        |
| 1.2 DRUHY WEBOVÝCH APLIKACÍ.....                                        | 13        |
| 1.2.1 Statické webové aplikace .....                                    | 13        |
| 1.2.2 Dynamické webové aplikace .....                                   | 13        |
| 1.2.3 Jednostránkové aplikace.....                                      | 13        |
| 1.2.4 Vícestránkové aplikace.....                                       | 14        |
| 1.3 ARCHITEKTURA WEBOVÝCH APLIKACÍ .....                                | 14        |
| 1.3.1 Client-server model .....                                         | 14        |
| 1.3.2 Vzor Model-View-Controller (MVC).....                             | 15        |
| 1.3.3 Vzor Model-View-ViewModel (MVVM) .....                            | 16        |
| 1.3.4 Architektonický styl REST (Representational State Transfer) ..... | 17        |
| <b>2 EXISTUJÍCÍ ŘEŠENÍ .....</b>                                        | <b>19</b> |
| 2.1 TRELLO .....                                                        | 19        |
| 2.2 ASANA.....                                                          | 20        |
| 2.3 MICROSOFT TEAMS .....                                               | 21        |
| 2.4 JIRA .....                                                          | 22        |
| 2.5 MONDAY .....                                                        | 23        |
| 2.6 SOUHRN.....                                                         | 24        |
| <b>3 TECHNOLOGIE VÝVOJE .....</b>                                       | <b>25</b> |
| 3.1 HTML.....                                                           | 25        |
| 3.2 CSS.....                                                            | 26        |
| 3.3 PHP.....                                                            | 26        |
| 3.4 JAVASCRIPT .....                                                    | 27        |
| 3.5 MYSQL.....                                                          | 28        |
| 3.6 APACHE.....                                                         | 29        |
| 3.7 PHPMYADMIN .....                                                    | 30        |
| <b>II PRAKTICKÁ ČÁST .....</b>                                          | <b>31</b> |
| <b>4 NÁVRH APLIKACE .....</b>                                           | <b>32</b> |
| 4.1 NÁVRH UŽIVATELSKÉHO ROZHRANÍ .....                                  | 32        |
| 4.1.1 Registrace .....                                                  | 32        |
| 4.1.2 Autorizace.....                                                   | 33        |
| 4.1.3 Pracovní stránka .....                                            | 34        |
| 4.1.4 Blok „Seznam úkolů“ .....                                         | 34        |
| 4.1.5 Blok „Graf“ .....                                                 | 35        |

|          |                                            |           |
|----------|--------------------------------------------|-----------|
| 4.1.6    | Blok „Koláčový graf“ .....                 | 36        |
| 4.1.7    | Blok „Kalendář“ .....                      | 37        |
| 4.1.8    | Blok „Poznámky“ .....                      | 38        |
| 4.1.9    | Blok „Sledovač času“ .....                 | 39        |
| 4.1.10   | Blok „Komentáře“ .....                     | 40        |
| 4.1.11   | Menu „Účastníci“ .....                     | 41        |
| 4.1.12   | Nastavení uživatele .....                  | 41        |
| 4.1.13   | Menu pozvánek .....                        | 42        |
| 4.2      | NÁVRH DATOVÉ STRUKTURY .....               | 43        |
| 4.2.1    | Tabulka users .....                        | 43        |
| 4.2.2    | Tabulka workspaces .....                   | 44        |
| 4.2.3    | Tabulka workspace_blocks .....             | 44        |
| 4.2.4    | Tabulka workspace_invitations .....        | 45        |
| 4.2.5    | Tabulka workspace_participants .....       | 45        |
| 4.3      | LOGIKA A SCÉNÁŘE POUŽITÍ .....             | 46        |
| 4.3.1    | Autentizační scénář .....                  | 46        |
| 4.3.2    | Scénář pro práce na pracovní stránce ..... | 47        |
| 4.3.3    | Scénář pro práce s boky .....              | 48        |
| 4.3.4    | Scénář pro interakci s uživateli .....     | 49        |
| <b>5</b> | <b>IMPLEMENTACE APLIKACE .....</b>         | <b>51</b> |
| 5.1      | STRUKTURA PROJEKTU .....                   | 51        |
| 5.1.1    | db.php .....                               | 51        |
| 5.1.2    | login.php .....                            | 52        |
| 5.1.3    | index.php .....                            | 52        |
| 5.1.4    | logout.php .....                           | 53        |
| 5.1.5    | user_actions.php .....                     | 53        |
| 5.1.6    | styles.css .....                           | 54        |
| 5.1.7    | login_styles.css .....                     | 54        |
| 5.1.8    | avatars/.htaccess .....                    | 54        |
| 5.2      | UKÁZKA FINÁLNÍHO ROZHRAŇÍ .....            | 54        |
| 5.2.1    | Registrace .....                           | 54        |
| 5.2.2    | Přihlášení do účtu .....                   | 55        |
| 5.2.3    | Pracovní stránka .....                     | 56        |
| 5.2.4    | Blok „Seznam úkolů“ .....                  | 57        |
| 5.2.5    | Blok „Graf“ .....                          | 58        |
| 5.2.6    | Blok „Koláčový graf“ .....                 | 59        |
| 5.2.7    | Blok „Kalendář“ .....                      | 60        |
| 5.2.8    | Blok „Poznámky“ .....                      | 62        |
| 5.2.9    | Blok „Sledovač času“ .....                 | 62        |
| 5.2.10   | Blok „Komentáře“ .....                     | 63        |
| 5.2.11   | Menu „Účastníci“ .....                     | 64        |
| 5.2.12   | Nastavení uživatele .....                  | 65        |
| 5.2.13   | Menu pozvánek .....                        | 65        |
| 5.3      | OCHRANA PŘED ZRANITELNOSTMI .....          | 66        |
| 5.3.1    | SQL injekce .....                          | 66        |

|          |                                                 |           |
|----------|-------------------------------------------------|-----------|
| 5.3.2    | Bezpečné uložení hesel .....                    | 66        |
| 5.3.3    | Zobrazování systémových chyb .....              | 66        |
| 5.3.4    | Nesprávné zpracování nahraných souborů .....    | 67        |
| 5.3.5    | Útoky CSRF .....                                | 67        |
| 5.3.6    | Cross-site scripting (XSS) .....                | 67        |
| <b>6</b> | <b>TESTOVÁNÍ APLIKACE.....</b>                  | <b>68</b> |
| 6.1      | FUNKČNÍ TESTOVÁNÍ.....                          | 68        |
| 6.1.1    | Testování registrace.....                       | 68        |
| 6.1.2    | Testování autorizace.....                       | 69        |
| 6.1.3    | Testování práce s pracovními prostory .....     | 70        |
| 6.1.4    | Testování bloků .....                           | 72        |
| 6.1.5    | Testování interakcí mezi lidmi .....            | 73        |
| 6.1.6    | Testování rolí.....                             | 76        |
| 6.1.7    | Výsledky funkčního testování .....              | 78        |
| 6.2      | BEZPEČNOSTNÍ TESTOVÁNÍ.....                     | 78        |
| 6.2.1    | Testování ochrany proti SQL-injekcím .....      | 78        |
| 6.2.2    | Testování bezpečného uložení hesel .....        | 79        |
| 6.2.3    | Testování skrytí interních chyb .....           | 79        |
| 6.2.4    | Testování validace uploadu avataru .....        | 80        |
| 6.2.5    | Testování ochrany proti CSRF .....              | 80        |
| 6.2.6    | Testování odolnosti proti XSS .....             | 81        |
| 6.2.7    | Výsledky bezpečnostního testování .....         | 82        |
|          | <b>ZÁVĚR.....</b>                               | <b>83</b> |
|          | <b>SEZNAM POUŽITÉ LITERATURY .....</b>          | <b>84</b> |
|          | <b>SEZNAM OBRÁZKŮ .....</b>                     | <b>87</b> |
|          | <b>SEZNAM TABULEK.....</b>                      | <b>89</b> |
|          | <b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK .....</b> | <b>90</b> |
|          | <b>SEZNAM PŘÍLOH.....</b>                       | <b>92</b> |

## ÚVOD

Týmová práce hraje v dnešním světě klíčovou roli při úspěšné realizaci projektů bez ohledu na jejich obor. Organizace práce v týmu vyžaduje jasné rozdělení úkolů, kontrolu nad jejich prováděním a efektivní interakci mezi účastníky. K řešení těchto problémů se používají specializované webové aplikace pro správu týmových projektů jako Trello, Asana, Jira a další. Mnohá ze stávajících řešení však mají buď přílišnou funkcionalitu, která práci komplikuje, nebo naopak omezené možnosti, které neumožňují flexibilní přizpůsobení potřebám konkrétních týmů.

Cílem této práce je vyvinout webovou aplikaci pro správu týmových projektů, poskytující pohodlné a intuitivní rozhraní, funkcionalitu pro distribuci úkolů, sledování jejich realizace a interakci mezi členy týmu.

Práce prověří stávající řešení v této oblasti, analyzuje požadavky na webovou aplikaci a identifikuje technologie a nástroje nezbytné pro její vývoj. Aplikace bude implementována pomocí HTML, CSS, JavaScript, PHP a MySQL, což zajistí její flexibilitu, multiplatformní a možnost práce ve webovém prostředí.

## **I    TEORETICKÁ ČÁST**

# 1 WEBOVÁ APLIKACE

## 1.1 Definice

Webová aplikace je typ aplikace, kterou si uživatel nestahuje a neinstaluje na svůj počítač. Přístup k němu a jeho funkcím je přítomen přímo v prohlížeči. [1]

## 1.2 Druhy webových aplikací

V dnešní době existuje mnoho typů webových aplikací. Identifikoval jsem čtyři hlavní typy, které lze zkoumat a popsat.

### 1.2.1 Statické webové aplikace

Statické webové aplikace jsou nejjednodušším typem webových aplikací. Jiný název pro ně je webová stránka.

Webové stránky jsou speciální stránky, které představují sadu statických stránek s informacemi. Uživatel s nimi nemůže interagovat ani jakkoli měnit obsah těchto stránek. Na rozdíl od webových aplikací můžete k vytváření webových stránek použít pouze zdrojový kód v HTML a CSS.

Vzhledem ke své jednoduchosti a nemožnosti uživatelů ovlivnit obsah, který je na webové stránce, mají tyto stránky velmi omezenou funkčnost.

### 1.2.2 Dynamické webové aplikace

Dynamické webové aplikace mají oproti statickým více funkčnosti a složitější strukturu. Jsou speciální online stránky, které se skládají z dynamických stránek, se kterými může uživatel pracovat a měnit je. K vytvoření takových aplikací nebude stačit samotný HTML a CSS kód. Takové webové aplikace jsou také často propojeny s databázemi, které slouží uživatelům k práci s obsahem stránky.

### 1.2.3 Jednostránkové aplikace

Jednostránkové webové aplikace nebo Single-Page Applications (SPA) fungují tak, že načtou jednu HTML stránku, po které se obsah dynamicky aktualizuje bez nutnosti opětovného načítání. To umožňuje výrazné zvýšení rychlosti a plynulejší uživatelský zážitek. SPA lze použít pro různé informační webové stránky, kde není potřeba, aby s nimi uživatel interagoval. Hlavní výhodou SPA je okamžitá odezva rozhraní, protože po prvním

načtení stránky je výměna dat se serverem omezena na minimum. Tento přístup má však svá úskalí. Vyhledávače mohou například indexovat obsah hůře, protože se načítá dynamicky a samotná aplikace vyžaduje načítání všech potřebných zdrojů najednou, což prodlužuje dobu prvního načítání.

#### **1.2.4 Vícestránkové aplikace**

Na rozdíl od jednostránkových webových aplikací (SPA) se vícestránkové webové aplikace nebo Multi-Page Applications (MPA) skládají z více webových stránek, což znamená, že při přepínání mezi nimi a při interakci uživatele se stránkou je aplikace nucena stránku aktualizovat, což může ovlivnit rychlost webové aplikace.

Vícestránkové aplikace jsou vhodné pro velké projekty, protože mohou obsahovat velké objemy informací a různé informační bloky. Jsou také schopny poskytnout uživateli pohodlnou příležitost analyzovat a pracovat s těmito datovými poli.

Jednou z výhod MPA je lepší optimalizace pro vyhledávače. Vzhledem k tomu, že každou stránku lze indexovat samostatně, zjednodušuje to proces SEO optimalizace a zpřístupňuje obsah vyhledávačům. [2]

### **1.3 Architektura webových aplikací**

#### **1.3.1 Client–server model**

Moderní webové aplikace jsou téměř vždy postaveny na principu architektury klient-server. V tomto modelu více klientských zařízení (obvykle prohlížečů) odesílá požadavky a přijímá služby z centrálního serveru. Klient je zodpovědný za uživatelské rozhraní a generování požadavků a server je zodpovědný za zpracování těchto požadavků a vrácení výsledků klientovi. Důležitou vlastností systému klient-server je oddělení odpovědností: logika uživatele je oddělena od logiky ukládání a zpracování dat, což zvyšuje škálovatelnost a spravovatelnost systému. V podstatě webový prohlížeč uživatele hraje roli klienta a odesílá HTTP požadavky na server, na kterém běží aplikace (webový server nebo aplikace s vestavěným webovým serverem), a server může přistupovat k databázi nebo jiným zdrojům za účelem generování odpovědi. [3]



Obrázek 1 Jednoduchá klient-server aplikace

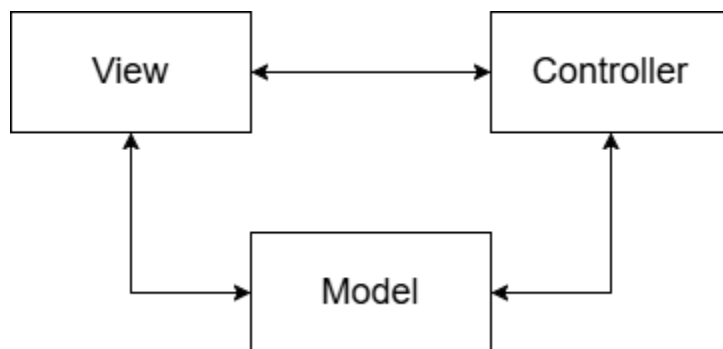
(zdroj: <https://www.open.edu/openlearn/science-maths-technology/an-introduction-web-applications-architecture/content-section-1.1>)

Složitější webové aplikace implementují vícevrstvý architektonický model – rozvoj myšlenky „klient-server“. Třívrstvá architektura se zpravidla rozlišuje s následujícími úrovněmi: prezentace (klientské uživatelské rozhraní), aplikace (obchodní logika na serveru) a data (DBMS nebo úložiště). Každá úroveň může být spuštěna na samostatné infrastruktuře a škálována nezávisle, což zjednodušuje podporu a vývoj systému. Například klientská úroveň je zodpovědná za zobrazování informací a sběr uživatelských vstupů (obvykle se jedná o kód HTML/CSS/JS spouštěný v prohlížeči), serverová úroveň implementuje obchodní pravidla (například v Pythonu, Javě, C# atd.) a datová úroveň ukládá a spravuje informace (relační DBMS nebo NoSQL úložiště). Spojovacím článkem je často API: klient přistupuje k serveru prostřednictvím standardizovaného rozhraní (například RESTful API) a serverová úroveň přistupuje k datové úrovni prostřednictvím požadavků na databázi. Tato vrstvená organizace odpovídá třívrstvé architektuře, která se v posledních desetiletích stala de facto klasickým vzorem pro webové aplikace.

Aplikace s více než třemi úrovněmi jsou vzácné, protože další úrovně poskytují jen malý přínos a mohou aplikaci zpomalit, ztížit její správu a zdražit její provoz. V důsledku toho je n-vrstvá a vícevrstvá architektura obvykle synonymem pro třívrstvou architekturu. [4]

### 1.3.2 Vzor Model–View–Controller (MVC)

Model–View–Controller (MVC) je jeden z nejznámějších vzorů softwarové architektury. Začal jako framework vyvinutý Trygve Reenskaugem pro platformu Smalltalk na konci 70. let. Vzor MVC rozděluje aplikaci na tři komponenty: Model, View a Controller, přičemž každá má svou vlastní roli.



Obrázek 2 Schéma MVC

*(zdroj: vlastní)*

Model obsahuje data aplikace a logiku správy dat (obchodní pravidla), View je zodpovědný za zobrazení stavu modelu (uživatelského rozhraní, se kterým uživatel interaguje) a Controller přijímá uživatelské vstupy a převádí je do akcí na modelu. Toto oddělení zajišťuje, že obchodní logika je izolována od kódu rozhraní, což zlepšuje udržitelnost a opětovné použití kódu. [5]

Ve webovém vývoji byl MVC upraven tak, aby oddělil logiku zobrazování webových stránek od serverové obchodní logiky. Například v frameworkech jako Ruby on Rails, Django, Laravel a ASP.NET MVC zpracovávají kontrolery HTTP požadavky a určují, jaká data načíst a jaké pohledy vrátit uživateli. Model ve webové aplikaci obvykle zahrnuje datové struktury a kód databáze, zatímco pohledy jsou šablony stránek, které generují HTML na základě dat modelu. Kontroler funguje jako prostředník: když obdrží požadavek (například URL nebo odeslání formuláře), přistupuje k modelu (aktualizuje stav nebo načítá data) a vybere vhodný pohled pro zobrazení výsledků. Mezi výhody MVC pro webové aplikace patří možnost samostatně vyvíjet a testovat komponenty. Vývojové týmy mohou paralelně pracovat na aplikační logice (Model), návrhu rozhraní (Pohled) a zpracování uživatelských akcí (Kontrolér). [6]

### 1.3.3 Vzor Model–View–ViewModel (MVVM)

Model–View–ViewModel (MVVM) je architektonický vzor, který je velmi podobný MVC, ale implementuje separaci poněkud odlišným způsobem. MVVM navrhli inženýři společnosti Microsoft Ken Cooper a Ted Peters v polovině prvního desetiletí 21. století s cílem zjednodušit vývoj uživatelského rozhraní ve Windows a formálně jej popsal John Gossman v roce 2005. Jeho podstatou je zavedení komponenty ViewModel, která funguje jako prostředník mezi datovým modelem a zobrazením.

V MVVM, stejně jako v MVC, existuje Model a View. Model představuje data aplikace a obchodní logiku (možná bez znalosti rozhraní) a View je zodpovědný za vizuální část (okno, webová stránka, komponenty uživatelského rozhraní). ViewModel obsahuje stav specifický pro tento view a logiku nezbytnou pro interakci uživatelského rozhraní s Modelem. Na rozdíl od MVC nemá ViewModel v MVVM přímé spojení s View – místo toho poskytuje data a příkazy, na které je View vázán. V prostředích, jako je WPF, umožňuje vázání dat automatickou synchronizaci rozhraní s vlastnostmi ViewModel: změny dat se automaticky projeví na obrazovce a akce uživatele (například kliknutí na tlačítko) způsobí spuštění příkazů definovaných ve ViewModel. [7]

#### 1.3.4 Architektonický styl REST (Representational State Transfer)

S rozvojem internetu a distribuovaných systémů vyvstala otázka standardizace interakcí mezi klienty a servery. Jedním z nejvlivnějších přístupů byl architektonický styl REST. Nejedná se o protokol ani specifickou technologii, ale o soubor interakčních stylů pro návrh distribuovaných systémů, zejména webových aplikací.

Klíčové myšlenky RESTu spočívají v tom, že systém je vnímán jako sada zdrojů, z nichž každý je adresován jednotným způsobem, a na těchto zdrojích je prováděna omezená sada operací, nebo v případě webu standardní sada metod HTTP:

- CONNECT – naváže „tunel“ k serveru určenému zdrojem.
- DELETE – smaže zadaný zdroj.
- GET – požaduje reprezentaci zdroje. Požadavky používající tuto metodu mohou pouze načítat data.
- HEAD – požaduje zdroj stejným způsobem jako metoda GET, ale bez těla odpovědi.
- OPTIONS – se používá k popisu možností připojení ke zdroji.
- POST – se používá k odesílání entit do konkrétního zdroje. Často způsobuje změnu stavu nebo nějaké vedlejší účinky na serveru.
- PUT – nahradí všechny aktuální reprezentace zdroje daty požadavku.
- TRACE – volá vrácenou testovací zprávu ze zdroje.
- PATCH – se používá k částečné úpravě zdroje. [8]

Téměř všechny hlavní webové služby nyní poskytují RESTful API – rozhraní, která se řídí principy REST, obvykle používají HTTP a výměnný formát JSON nebo XML k reprezentaci dat. Například RESTful webová aplikace může reprezentovat uživatelské informace jako zdroj /users. Požadavek GET /users/123 vrací reprezentaci uživatele s ID 123, PUT /users/123 aktualizuje tohoto uživatele, GET /users – seznam všech uživatelů atd. Ani klient, ani server neukládají kontext interakce mezi požadavky: každý požadavek je soběstačný. Pokud jeden server selže, jiný může požadavek zpracovat – klient si toho nevšimne.

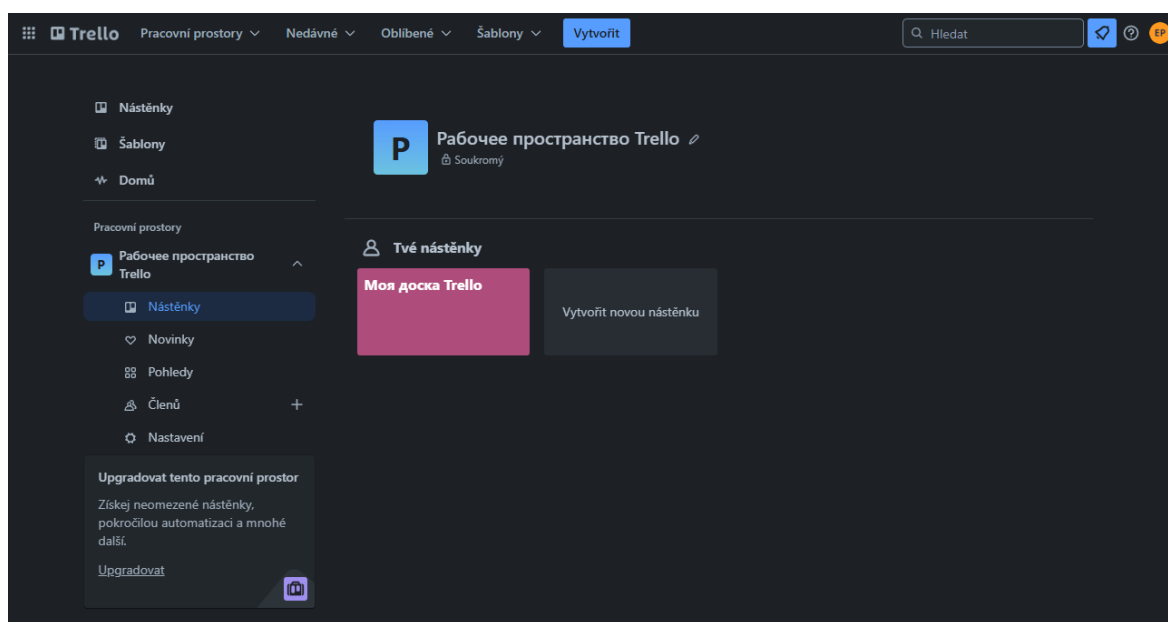
Z architektonického hlediska REST usnadnil přechod od náročných komunikačních modelů (např. SOAP, CORBA) k jednodušším, otevřeným webovým službám. Nízká hmotnost a škálovatelnost RESTu z něj učinily de facto standard pro integraci různých komponent webových aplikací a vytváření veřejných API. REST také prospívá údržbě systému: jednotlivé služby se mohou vyvíjet nezávisle, pokud dodržují smlouvy API. Vzhledem k tomu, že REST je styl a nikoli rigidní standard, mohou se konkrétní implementace lišit, ale dodržování principů REST obvykle znamená: návrh URL orientovaný na zdroje, používání správných metod HTTP s jasnou sémantikou, používání kódů odpovědí HTTP a neukládání stavu uživatele na server mezi požadavky. [9]

## 2 EXISTUJÍCÍ ŘEŠENÍ

Než se pustím do vývoje své webové aplikace pro správu týmových projektů, nejprve si prostuduji existující řešení dostupná na trhu. Analýza populárních platforem, jako jsou Trello, Asana, Jira, Monday a Microsoft Teams mi pomůže identifikovat jejich klíčové vlastnosti, silné a slabé stránky a určit, jaké funkce uživatelé skutečně potřebují. Tento průzkum mi umožní nejen vyhnout se duplikaci zbytečných funkcí, ale také přidat do mé aplikace nástroje, které ji učiní pro koncové uživatele pohodlnější a efektivnější. Studium konkurence mi navíc pomůže lépe porozumět moderním trendům ve vývoji podobných systémů a zvolit nejlepší přístup k realizaci svého projektu.

### 2.1 Trello

Trello je webová aplikace pro správu projektů a týmovou spolupráci, založená na metodě Kanban. Umožňuje uživatelům organizovat úkoly a projekty prostřednictvím vizuálních nástěnek, seznamů a karet, což zjednodušuje sledování postupu prací a zlepšuje komunikaci v týmu. [10]



Obrázek 3 Domovská stránka Trello

(zdroj: *trello.com*)

Hlavní funkce Trello:

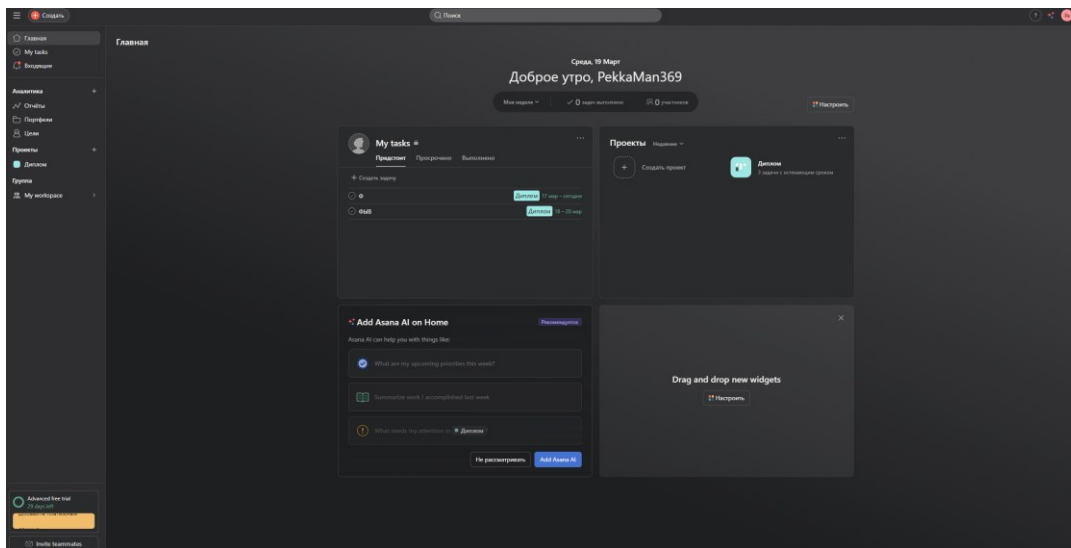
- Vytvoření různých pracovních prostorů
- Vytvoření nových nástěnek

- Možnost pozvat nebo nakopnout účastníky
- Přidávání poznámek do kalendáře
- Přidávání další produkce Atlassian
- Nastavení přístupu k úpravě nástěnky pro účastníky
- Možnost označit oblíbené nástěnky
- Rozdělení účastníků na členy, hosty a ty, kteří požádali o připojení
- Různé schopnosti různých účastníků0
- Možnost přizpůsobení profilu

Při použití Trello se nenačítají žádné další html soubory a aktualizuje se pouze pracovní plocha, takže se jedná o jednostránkovou webovou aplikaci.

## 2.2 Asana

Asana je speciální platforma pro řízení projektů, řízení zdrojů, stanovování cílů, reporting a automatizaci workflow. [11]



Obrázek 4 Domovská stránka Asana

(zdroj: [app.asana.com](https://app.asana.com))

Hlavní funkce Asana:

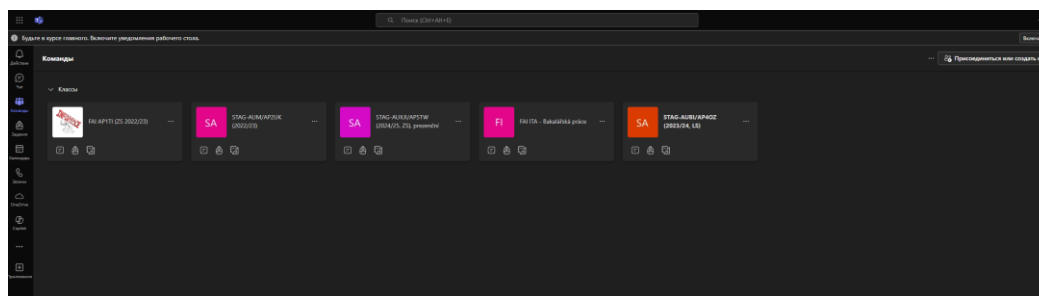
- Schopnost přizpůsobit widgety zobrazené na domovské stránce uživatele
- Vytvoření seznamu cílů

- Vytváření projektů
- Pozvání skupinových kolegů
- Vytváření portfolia pro organizaci projektů
- Sledování stavu dokončených úkolů
- Schopnost vytvořit osobní poznámkový blok
- Schopnost odesílat a přijímat komentáře
- Možnost přizpůsobení profilu
- Schopnost zobrazit analýzy pro vaši společnost/tým
- Použití vnitřní umělé inteligence k analýze celého pracovního prostoru

Při provádění akcí a navigaci uvnitř webové aplikace Asana se ukazuje, že se jedná o jednostránkovou webovou aplikaci, protože při přesunu do různých sekcí se neaktualizuje celá stránka, ale pouze pracovní plocha a nenačtou se nové html soubory.

## 2.3 Microsoft Teams

Microsoft Teams je speciální aplikace, ve které mohou uživatelé spolupracovat, vytvářet společné projekty a využívat online schůzky k diskusi o těchto projektech. [12]



Obrázek 5 Domovská stránka Microsoft Teams

(zdroj: [teams.microsoft.com](https://teams.microsoft.com))

Hlavní funkce Microsoft Teams:

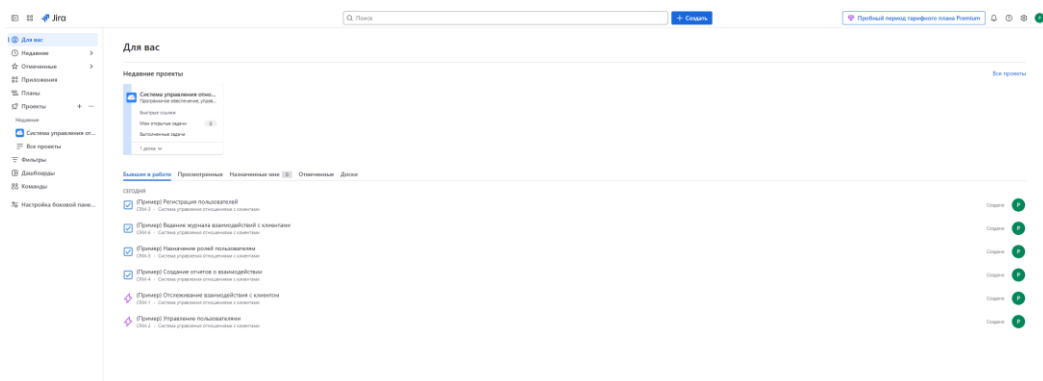
- Schopnost vytvářet týmy pro informování lidí
- Možnost zobrazení seznamu aktivních úkolů
- Možnost rozdělení informací a úkolů do různých skupin

- Chat pro komunikaci a přenos informací
- Možnost přizpůsobení profilu
- Integrace dalších aplikací společnosti Microsoft, jako je OneDrive
- Možnost přizpůsobení profilu
- Možnost stažení a instalace desktopové verze aplikace

Při přepínání mezi hlavními kartami Microsoft Teams se načítají nové dokumenty, což znamená, že se jedná o vícestránkovou webovou aplikaci

## 2.4 Jira

Jira je nástrojem pro řízení projektů pro plánování a sledování práce v každém týmu, plánování a spouštění kampaní, sledování produktů a problémů, plánování a sledování projektů, vytváření kreativních pracovních postupů a vytváření vlastních procesů [13]



Obrázek 6 Domovská stránka Jira

(zdroj: [atlassian.com](https://atlassian.com))

Hlavní funkce Jira:

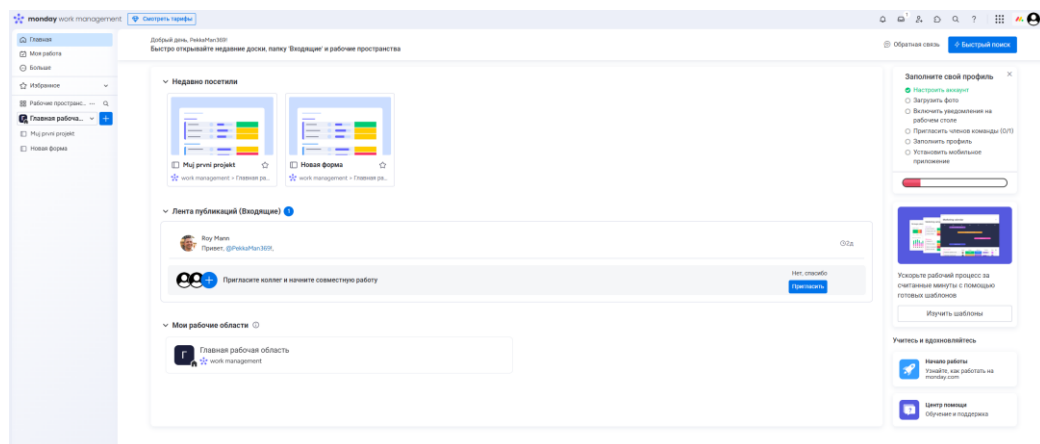
- Vytváření více projektů
- Vytváření více dashboardů
- Schopnost vytvářet týmy a zvát členy do týmu
- Schopnost vytvářet plány
- Možnost vytváření závislostí mezi různými akcemi
- Možnost zobrazení statistik akcí

- Schopnost stanovit priority úkolů
- Možnost přizpůsobení profilu

Některé z možností webové aplikace Jira vyžadují dodatečné načítání stránky a načítání nových HTML dokumentů, což naznačuje, že se jedná o vícestránkovou webovou aplikaci.

## 2.5 Monday

Monday je pracovní platforma pro zefektivnění pracovních postupů a získání jasného přehledu o výkonnosti týmů pro strategická rozhodnutí. [14]



Образек 7 Domovská stránka Monday

(zdroj: monday.com)

Hlavní funkce Monday:

- Schopnost vytvářet pracovní prostory
- Možnost vytvářet dashboards
- Možnost přidat kolegy pro spolupráci
- Možnost přidat do projektu widgety (grafy, časová osa, rozpočet atd.)
- Schopnost pracovat s kalendářem
- Schopnost zobrazit zdroj aktivit vašich kolegů
- Možnost přidání do oblíbených
- Možnost importu dat z jiných aplikací
- Možnost využití AI pro organizační úkoly

- Možnost přizpůsobení profilu

Při použití Monday se aktualizuje pouze pracovní prostor. Nejsou načteny žádné jiné soubory html, což znamená, že se jedná o jednostránkovou webovou aplikaci.

## 2.6 Souhrn

Po prozkoumání existujících webových aplikací pro řízení týmových projektů jsem identifikoval několik klíčových funkcí, které by moje aplikace měla obsahovat, aby splnila hlavní požadavky uživatelů. Mezi ně patří:

- Vytváření pracovních prostorů.
- Přidávání účastníků do pracovních prostorů.
- Nastavení oprávnění k přístupu.
- Sledování úkolů a projektů.
- Kalendář a plánování (termíny, připomínky)
- Komentování a diskuse o úkolech (zprávy, zpětná vazba)
- Přizpůsobení uživatelského profilu (avatar, informace, preference)

Také webová aplikace by měla vypadat krásně a měla by být pro uživatele intuitivní. Abych identifikoval dobré a špatné aspekty designu, analyzoval jsem vzhled existujících aplikací na trhu. Webová aplikace Jira se ukázala jako velmi nepohodlná a neintuitivní. Je obtížné pochopit a snadno se zmást, což činí používání této aplikace nepohodlným, a dokonce nepříjemně matoucím. Nejkrásnější, nepohodlnější a nejintuitivnější aplikací pro mě bylo Monday. Vypadá pěkně, není zahlcený zbytečnými informacemi a dokázal jsem na to přijít během pár minut používání, i když se od ostatních hodně liší.

### 3 TECHNOLOGIE VÝVOJE

Technologie vývoje webových aplikací zahrnují značkovací jazyky HTML a CSS, programovací jazyky JavaScript a PHP a technologie pro správu databází.

#### 3.1 HTML

HTML je standardizovaný hypertextový značkovací jazyk pro prohlížení webových stránek v prohlížeči. Značkovací jazyk je systém pro anotaci dokumentu, zviditelnění jeho struktury a dalších prvků pro softwarové zpracování. Používá se k definování textu a vkládání příkazů nebo instrukcí do dokumentu, které mohou řídit jeho formátování nebo strukturu. Značkovací jazyky se používají ke specifikaci struktury a prezentace dat, spíše než k provádění výpočtů nebo logických operací. [15]

Základními pojmy značkovacích jazyků jsou prvky, tagy a atributy. Tag je prvek, který definuje strukturu dokumentu. Existují otevírací a zavírací tagy. Element je značka spolu s daty, která obsahuje. Atribut se používá ve spojení s prvkem k definování libovolného z jeho parametrů.

```
<a href="https://example.com">Webová stránka</a>
```

V tomto kódu je celý řádek prvkem a obsahuje některá data.

`<a>` a `</a>` jsou otevírací a zavírací tagy.

`href="https://example.com"` je atribut, který obsahuje odkaz na web.

HTML5 oproti svým předchůdcům výrazně rozšířil své možnosti. Do jazyka byly přidány nové sémantické prvky (např. `<article>`, `<section>`, `<header>`, `<footer>`, `<nav>`), které jasněji popisují strukturu stránky a zlepšují přístupnost obsahu. HTML5 navíc integruje vložená multimédia – podporu pro přehrávání video a audio obsahu bez použití pluginů. Objevilo se vestavěné grafické API `<canvas>` pro dynamické vykreslování rastrové grafiky a her přímo v prohlížeči a také nativní podpora pro vloženou vektorovou grafiku SVG a matematické značky MathML. HTML5 navíc přineslo vylepšení webových formulářů – definovalo nové typy vstupních polí, jako například „email“, „tel“, „url“, „search“ či „range“. Přibyly také atributy „placeholder“ a „autofocus“, které zvyšují uživatelskou přívětivost formulářů. [16]

## 3.2 CSS

CSS není programovací jazyk. Jazyk CSS se používá k popisu vzhledu prvků, které jsou napsány v jazyce HTML nebo XML. Popisuje, jak mají být všechny prvky zobrazeny na informačních nosičích. CSS je jedním ze základních jazyků otevřeného webu a je standardizováno ve všech webových prohlížečích podle specifikací W3C. [17]

Hlavním principem, na kterém je založeno CSS, je oddělení obsahu od designu: webová stránka je vyvíjena jako sémanticky strukturovaný HTML dokument a veškeré informace o tom, jak by se tento dokument měl uživateli zobrazovat, jsou umístěny v externím CSS stylu. Toto oddělení zlepšuje zpracovatelnost a opětovné použití kódu: obsah lze upravovat nezávisle na designu a naopak, což snižuje složitost údržby webových stránek.

CSS dostalo svůj název podle mechanismu kaskádových stylů: pokud se na jeden prvek vztahuje několik pravidel CSS (například z různých stylových listů nebo uživatelských nastavení), určí se priorita a tato pravidla se sloučí do výsledného stylu. CSS také podporuje dědičnost vlastností: podřízené prvky mohou automaticky převzít některé styly svých rodičů, pokud není uvedeno jinak. Tyto mechanismy zjednodušují aplikaci jednotného designu pro celý web a flexibilní přizpůsobení jednotlivých částí.

Zatímco CSS byl zpočátku primárně zodpovědný za dekorativní design stránek, dnes se CSS používá k implementaci komplexních adaptivních rozhraní a zajištění uživatelsky přívětivé interakce napříč zařízeními. Pochopení principů HTML a CSS umožňuje všem účastníkům webových projektů – od designérů až po manažery – vytvářet atraktivnější a uživatelsky přívětivější weby a efektivně spolupracovat během procesu vývoje. [18]

## 3.3 PHP

PHP nebo Hypertext Preprocessor je skriptovací jazyk na straně serveru, který umožňuje, aby vaše webové stránky byly skutečně dynamické. Jeho flexibilita a relativně malá křivka učení z něj činí jeden z nejpobulárnějších skriptovacích jazyků v okolí.

Na rozdíl od JavaScriptu běží PHP na straně serveru: skripty interpretuje webový server a generuje dynamický HTML kód, který je následně odeslán klientovi. PHP je tedy zodpovědný za obchodní logiku webové aplikace na serveru – zpracování formulářů, správu relací, generování stránek na základě dat z databáze atd. Podle vývojového týmu PHP se jedná o „rychlý, flexibilní a praktický jazyk“, který se používá všude – od vytváření jednoduchých blogů až po implementaci nejpobulárnějších webových stránek na světě. [19]

PHP byl původně nástroj pro vkládání dynamického obsahu do HTML stránek. Jeho syntaxe a koncepty jsou podobné jazykům C/Java a Perl, takže se snadno učí. Kód PHP je vkládán přímo do HTML kódu stránky (bloky kódu jsou umístěny mezi speciálními tagy `<?php ... ?>`) a interpret na serveru tento kód spustí s každým požadavkem a výsledky vloží do výstupního HTML. Tento model provádění se ukázal jako mimořádně úspěšný: PHP umožňuje relativně jednoduché vytváření dynamických stránek a interakci s databázemi, což vedlo k explozivnímu nárůstu jeho popularity na začátku roku 2000. Vznikl klasický balíček LAMP (Linux, Apache, MySQL, PHP) - rozšířený stack open source softwaru, na jehož základě fungovala řada webových stránek a aplikací. LAMP získal status jedné z nejoblíbenějších a nejspolehlivějších platform pro nasazení dynamických webových systémů. [20]

V průběhu času se jazyk PHP neustále vyvíjí, zachovává si zpětnou kompatibilitu a zohledňuje moderní požadavky. Aktuální větev PHP 8 (verze 8.0–8.3) přinesla významná vylepšení výkonu (včetně JIT kompilace), sníženou spotřebu paměti a rozšířené možnosti silného typování, což jí umožňuje konkurovat alternativním serverovým platformám na Node.js, Pythonu, Javě atd. Do jazyka byly přidány nové syntaktické funkce (silné typy, anonymní třídy, šipkové funkce atd.), byl vylepšen systém pro ošetření chyb a výjimek a objevila se moderní kryptografická rozšíření. Ekosystém PHP je stále rozsáhlý: tento jazyk používají miliony vývojářů po celém světě, existuje bohatá sada knihoven a frameworků a rozmanitá komunita nadále přispívá k otevřeným projektům a sdílí znalosti. [21]

### 3.4 JavaScript

JavaScript se vyvinul z jednoduchého skriptovacího jazyka pro webové stránky v klíčovou technologii moderního webového vývoje. Dnes hraje nezastupitelnou roli při tvorbě interaktivních webových aplikací a jeho využití se rozšířilo i mimo prostředí webových prohlížečů – například na servery, mobilní zařízení či desktopové aplikace. Díky své univerzálnosti a široké podpoře patří mezi nejrozšířenější programovací jazyky na světě.

Tento programovací jazyk nijak nesouvisí s jazykem Java. Používá se na straně klienta i serveru při vytváření a používání webových stránek a webových aplikací. Na straně klienta je JavaScript primárně zodpovědný za vykreslování a interakci s uživatelským rozhraním, zatímco na straně serveru umožňuje spouštět složité aplikace a ovlivňovat informace. [22]

JavaScript umožnil implementovat interaktivní vizualizace dat a komplexní uživatelská rozhraní ve webovém prostředí. HTML5 k tomu poskytuje potřebné nástroje – vestavěné

prvky <canvas> a SVG grafiku – které v kombinaci s JavaScriptem umožňují vytvářet dynamicky se aktualizující grafy, tabulky a další vizuální reprezentace dat přímo v prohlížeči uživatele. Účinnost takových vizualizací je přirozeně určena kompetentním návrhem. Výzkumníci v oblasti vizuální analytiky zdůrazňují potřebu pečlivě zvážit vhodnost formy vizuální reprezentace pro řešený problém. [23]

Celkově si JavaScript upevnil své místo jako univerzální jazyk pro webový vývoj. Moderní webové aplikace jsou bez JavaScriptu téměř nemyslitelné – v roce 2025 téměř 99 % webových stránek používá klientský JS kód. [24]

### 3.5 MySQL

MySql je systém pro správu relačních databází, který je distribuován jako svobodný software. Je jedním z nejoblíbenějších, protože je flexibilní, lehký a snadno se používá. Slovo "relační" znamená, že databáze jsou reprezentovány jako související informace a jsou popsány jako soubor vztahů. MySQL pracuje s dotazovacím jazykem SQL, který se tradičně používá v databázích. Některé z nejpopulárnějších funkcí tohoto programu jsou následující:

- Prostřednictvím vláken jádra lze použít více procesorů
- Multiplatformní provoz
- Mnoho typů sloupců pokrývá prakticky každý typ dat
- Skupinové funkce pro matematické výpočty a řazení
- Příkazy, které umožňují snadné a stručné zobrazení informací o databázích správce
- Názvy funkcí, které neovlivňují názvy tabulek nebo sloupců
- Systém ověřování hesel a uživatelů pro vyšší zabezpečení
- Je povoleno až 32 indexů na tabulku; tato funkce byla úspěšně implementována na úrovních 60 000 tabulek a 5 000 000 000 řádků (verze 4.1.2, aktuálně ve vývoji, umožní 64 indexy)

Mezinárodní hlášení chyb použitelné v mnoha různých zemích MySQL je perfektní volbou pro poskytování dat přes internet díky své schopnosti zvládat velké zatížení a svým pokročilým bezpečnostním opatřením. [25]

Další výhodou MySQL je jeho snadné osvojení a používání. Nastavení serveru a databázi je intuitivní, syntaxe dialektu SQL se blíží standardu. Vývojář má přístup k četným grafickým nástrojům a utilitám (například phpMyAdmin, MySQL Workbench) pro správu databází a také k ovladačům pro téměř všechny programovací jazyky. V prostředí PHP byla podpora MySQL zpočátku implementována prostřednictvím PHP rozšíření (funkce `mysql_*`, později vylepšené `mysqli` a `PDO`). To vedlo k téměř de facto kombinaci PHP + MySQL v nespočtu webových projektů. V kontextu webové aplikace pro projektový management se MySQL obvykle používá k ukládání perzistentních systémových dat: informací o uživateli, týmech, projektech, úkolech, termínech, průběhu atd. Relační datový model je pro takové strukturované informace dobře vhodný - lze definovat související tabulky (například tabulku projektu, tabulku úkolů s cizím klíčem k projektu, tabulku uživatelů a jejich rolí atd.), což zajišťuje integritu a konzistenci dat. Pomocí jazyka SQL aplikace provede potřebné dotazy – výběr úkolů pro projekt, aktualizaci stavů, agregaci dat pro reporty atd. – a systém MySQL DBMS tyto dotazy efektivně zpracuje a poskytne výsledky.

Spolu s MySQL existují další populární DBMS v moderním vývoji webových aplikací. Relační model konkuruje rodině úložišť NoSQL, která obětují striktní strukturu ve prospěch škálovatelnosti a flexibility (například MongoDB, Cassandra atd.). Kombinace relační databáze a serverového jazyka však zůstává základní šablonou pro implementaci obrovského množství projektů. MySQL zaujímá zvláštní místo jako nejrozšířenější open-source SŘBD a jeho status je podpořen desítkami let úspěšného používání v oboru na platformách, jako je Facebook, Twitter, YouTube atd. [26]

### 3.6 Apache

Apache HTTP Server (často označovaný jednoduše jako Apache) je multiplatformní webový server s otevřeným zdrojovým kódem, vyvinutý a spravovaný organizací Apache Software Foundation. Apache byl poprvé vydán v roce 1995 a do dubna 1996 se stal nejoblíbenějším HTTP serverem na internetu. [27]

Apache je založen na modulární architektuře, která umožňuje propojení různých komponent a rozšíření funkčnosti serveru tak, aby vyhovoval jakýmkoli potřebám. Apache běží na většině populárních operačních systémů (Unix/Linux, Windows, macOS) a podporuje mechanismus virtuálního hostingu, který umožňuje obsluhovat více webových

stránek na jedné instanci server. Konfiguraci lze flexibilně nastavit jak globálním souborem `httpd.conf`, tak na úrovni jednotlivých adresářů pomocí souborů `.htaccess`.

Apache se integruje do populárních platforem pro vývoj webových aplikací. Zejména je nedílnou součástí klasického LAMP stacku (Linux, Apache, MySQL, PHP), který se široce používá k nasazování dynamických webů a webových aplikací. Díky své otevřenosti a rozšiřitelnosti se Apache snadno kombinuje s různými programovacími jazyky a systémy pro správu databází a lze jej také zahrnout do softwarových prostředí (například balíčků jako XAMPP) pro lokální vývoj. Aktivní komunita a bohatá dokumentace přispívají k neustálému vylepšování Apache v souladu s aktuálními webovými standardy. [28]

Všechny tyto vlastnosti dělají z Apache HTTP Serveru jedno z nejspolehlivějších a nejvšestrannějších řešení pro provoz webových serverů, žádané jak v průmyslovém provozu, tak i ve vzdělávacích projektech.

### 3.7 phpMyAdmin

phpMyAdmin je bezplatná webová aplikace napsaná v PHP, určená pro správu databází MySQL/MariaDB prostřednictvím prohlížeče. phpMyAdmin poskytuje uživateli grafické rozhraní pro provádění nejběžnějších operací se systémem DBMS: vytváření a mazání databází, tabulek, polí, správa relací a indexů, uživatelských účtů a jejich práv atd. - to vše lze provádět interaktivně bez nutnosti ručního psaní SQL kódu. Zároveň je pro větší flexibilitu k dispozici možnost přímého spouštění libovolných SQL dotazů, což umožňuje kombinovat pohodlí grafického rozhraní s plnou kontrolou nad databází.

phpMyAdmin má bohatou sadu funkcí, které výrazně zjednodušují správu databáze. Patří mezi ně import a export dat v různých formátech (CSV, SQL, XML, PDF, OpenDocument, JSON atd.), vizuální zobrazení struktury databáze (generování diagramů), podpora uložených procedur a triggerů, mechanismus pro konstrukci složitých dotazů na základě vzorku (QBE), globální vyhledávání podle databáze a mnoho dalšího. Je možné současně připojit a spravovat několik serverů MySQL prostřednictvím jediného panelu phpMyAdmin, což je výhodné pro složitou infrastrukturu. [29]

V prostředí pro vývoj webových aplikací je phpMyAdmin ceněn jako spolehlivý a vizuální nástroj, který umožňuje rychle provádět administrativní úkony (prohlížení a změna dat, zálohování, nastavení přístupových práv atd.) v rámci projektu.

## II PRAKTICKÁ ČÁST

## 4 NÁVRH APLIKACE

Než začnu s vývojem týmové aplikace, musím navrhnout klíčové aspekty budoucí aplikace. V této fázi se jedná o promyšlení uživatelského rozhraní, vytvoření a sestavení funkční a související datové struktury a definování uživatelských scénářů pro práci s aplikací, které určí logiku aplikace. Tento přístup umožní vytvořit pohodlí pro uživatele a ponechá prostor pro další rozvoj aplikace.

Makety uživatelského rozhraní byly vytvořeny pomocí mockflow.com.

### 4.1 Návrh uživatelského rozhraní

#### 4.1.1 Registrace

Registrační stránka vypadá jednoduše a přehledně. Je okamžitě jasné, jaké údaje je třeba zadat k vytvoření nového účtu. V horní části je zvýrazněn nadpis „Registrace“, takže je jasné, kde se právě nacházíte. Povinné jsou pouze tři pole: e-mail, křestní jméno a příjmení. Díky tomu je registrace rychlá a není zahlcena zbytečnými informacemi. Z bezpečnostních důvodů jsou zde dvě pole pro heslo: do jednoho zadejte požadované heslo a do druhého jej potvrďte. To pomáhá vyhnout se chybám. Ostatní údaje, jako je datum narození a organizace, jsou volitelné, ale můžete je vyplnit, pokud chcete více personalizace. Všechna pole mají jasné popisky a příklady, takže vyplnění formuláře je snadné. Dole je patrné tlačítko pro dokončení registrace a odkaz na přihlašovací stránku pro ty, kteří již účet mají. To pomáhá vyhnout se nejasnostem: pokud jste se omylem dostali k registraci, můžete se rychle vrátit k přihlášení. Celkově je proces registrace pohodlný a rychlý.

The registration form is titled "Registrace". It contains the following fields and elements:

- E-mail\* (Zadejte e-mail)
- Heslo\* (Zadejte heslo)
- Potvrďte heslo\* (Zadejte heslo znovu)
- Jméno\* (Zadejte jméno)
- Příjmení\* (Zadejte příjmení)
- Datum narození (dd.mm.rrrr)
- Organizace (Zadejte organizaci)
- Registrovat se (button)
- Přihlásit se (link)

Obrázek 8 Návrh registračního formuláře

*(zdroj: vlastní)*

#### 4.1.2 Autorizace

Přihlašovací stránka je provedena co nejjednodušeji a nejpráhledněji. Uprostřed formuláře jsou dvě pole pro zadání e-mailu a hesla, nad každým je napsáno, co je třeba zadat. Po vyplnění stačí kliknout na tlačítko „Přihlásit se“ pro odeslání údajů. Pokud uživatel ještě nemá účet, ihned uvidí pod formulářem odkaz „Vytvořit účet“.

The login form is titled "Přihlášení do účtu". It contains the following fields and elements:

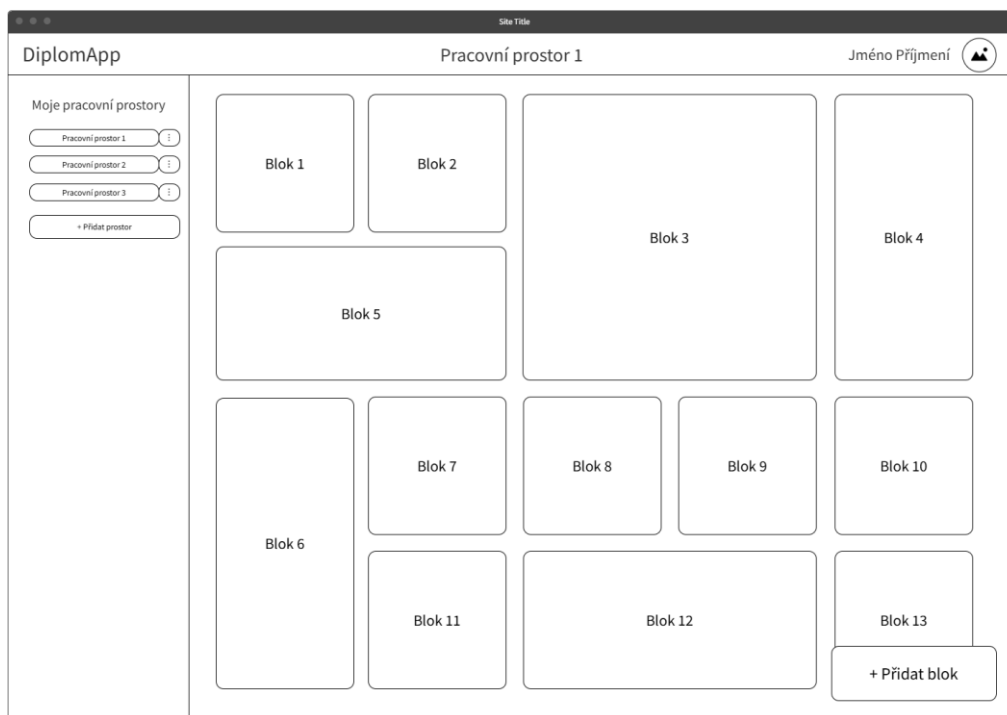
- E-mail (Zadejte e-mail)
- Heslo (Zadejte heslo)
- Přihlásit se (button)
- Vytvořit účet (link)

Obrázek 9 Návrh přihlašovacího formuláře

*(zdroj: vlastní)*

### 4.1.3 Pracovní stránka

Toto je hlavní obrazovka pracovní stránky aplikace. Vlevo se nachází boční menu se seznamem pracovních prostorů. Můžete mezi nimi rychle přepínat nebo přidávat nové. Po kliknutí na tlačítka menu napravo od názvu pracovního prostoru se otevře vyskakovací menu, kde můžete prostor přejmenovat, zobrazit účastníky nebo prostor smazat. Název aktuálního prostoru je uveden v horní části obrazovky a vpravo jsou viditelná data uživatelského profilu. Po kliknutí na avatara se otevře vyskakovací menu, kde můžete otevřít nastavení uživatele, zobrazit pozvánky od ostatních účastníků nebo se odhlásit z profilu. Hlavní část obrazovky je vyplněna bloky různých velikostí a účelů, pokud je uživatel přidal. Každý blok lze podle potřeby přidat nebo odebrat, k tomuto účelu je vpravo dole speciální tlačítko. Uživatel si sám vybere, které bloky budou do pracovního prostoru přidány.



Obrázek 10 Návrh pracovní stránky

(zdroj: vlastní)

### 4.1.4 Blok „Seznam úkolů“

Blok „Seznam úkolů“ je určen pro správu úkolů. Zde může uživatel rychle přidat nový úkol pouhým zadáním jeho názvu a kliknutím na tlačítko „+“. Každý úkol se zobrazuje v samostatném řádku. Napravo od úkolu se nacházejí tlačítka pro jeho dokončení nebo smazání. V případě potřeby může uživatel úkol označit jako dokončený nebo jej ze seznamu zcela odstranit. Díky posouvání uživatel vždy vidí celý seznam úkolů, i když jich je mnoho.

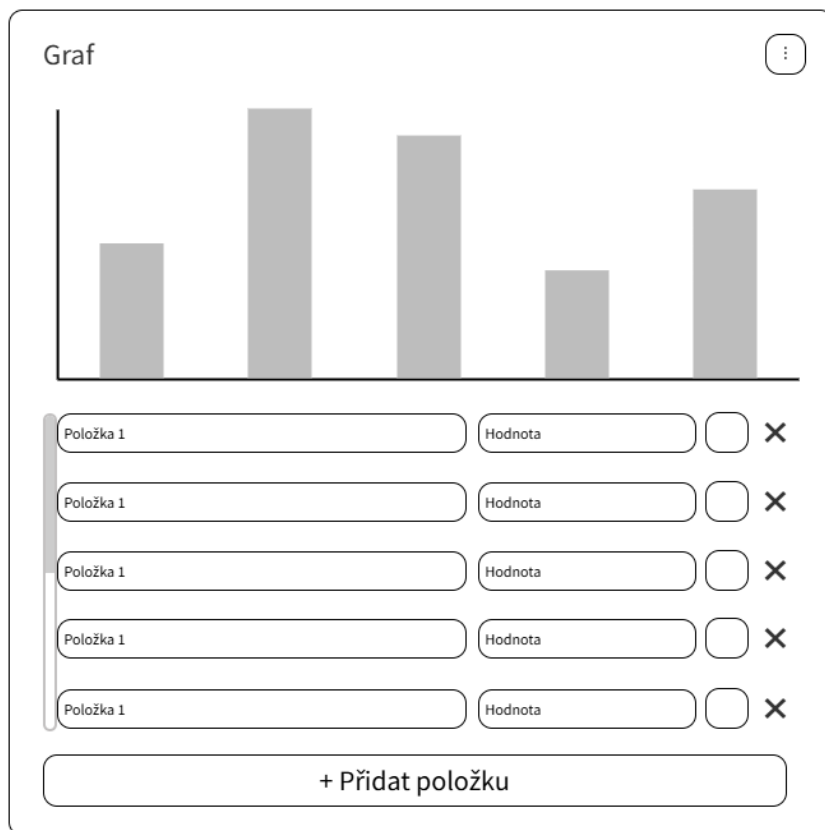
| Seznam úkolů     |   |   |
|------------------|---|---|
| Přidat úkol... + |   |   |
| Úkol 1           | ✓ | ✕ |
| Úkol 2           | ✓ | ✕ |
| Úkol 3           | ✓ | ✕ |
| Úkol 4           | ✓ | ✕ |

Obrázek 11 Návrh bloku „Seznam úkolů“

*(zdroj: vlastní)*

#### 4.1.5 Blok „Graf“

Blok „Graf“ je určen k zobrazení dat ve formě diagramu. Uživatel klikne na tlačítko „+ Přidat položku“, po kterém se zobrazí prázdná pole pro zadání názvu prvku a jeho hodnoty. Po vyplnění těchto polí se prvek okamžitě zobrazí v grafu jako samostatný sloupec. Každý prvek lze smazat nebo změnit jeho barvu. K tomuto účelu jsou napravo od něj umístěna speciální tlačítka. Díky tomu si uživatel graf flexibilně konfiguruje a přidává nebo odebírá potřebná data.



Obrázek 12 Návrh bloku „Graf“

*(zdroj: vlastní)*

#### 4.1.6 Blok „Koláčový graf“

Blok „Koláčový graf“ je určen k zobrazení dat ve formě koláčového diagramu. Uživatel klikne na tlačítko „+ Přidat položku“, po kterém se zobrazí prázdná pole pro zadání názvu prvku a jeho hodnoty. Po vyplnění těchto polí se prvek okamžitě zobrazí v grafu jako samostatný sloupec. Každý prvek lze smazat nebo změnit jeho barvu. K tomuto účelu jsou napravo od něj umístěna speciální tlačítka. Díky tomu si uživatel graf flexibilně konfiguruje a přidává nebo odebírá potřebná data.

Koláčový graf

Položka 1

Hodnota

×

Položka 1

Hodnota

×

Položka 1

Hodnota

×

Položka 1

Hodnota

×

Položka 1

Hodnota

×

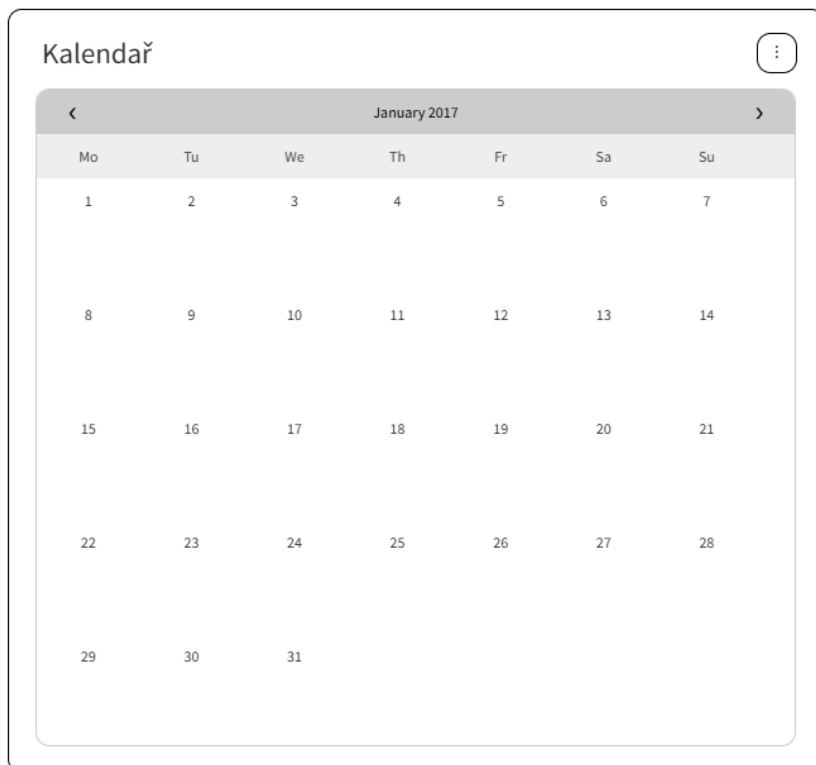
+ Přidat položku

Obrázek 13 Návrh bloku „Koláčový graf“

*(zdroj: vlastní)*

#### 4.1.7 Blok „Kalendář“

Blok „Kalendář“ umožňuje uživateli zobrazit data a zanechávat poznámky k jednotlivým dnům. V horní části bloku jsou tlačítka pro přepínání mezi měsíci a zobrazuje se aktuální měsíc a rok. Při výběru dne v kalendáři si uživatel může vytvořit nebo zobrazit již vytvořené poznámky k tomuto datu. Kalendář tak slouží jako praktický nástroj pro plánování a sledování událostí.

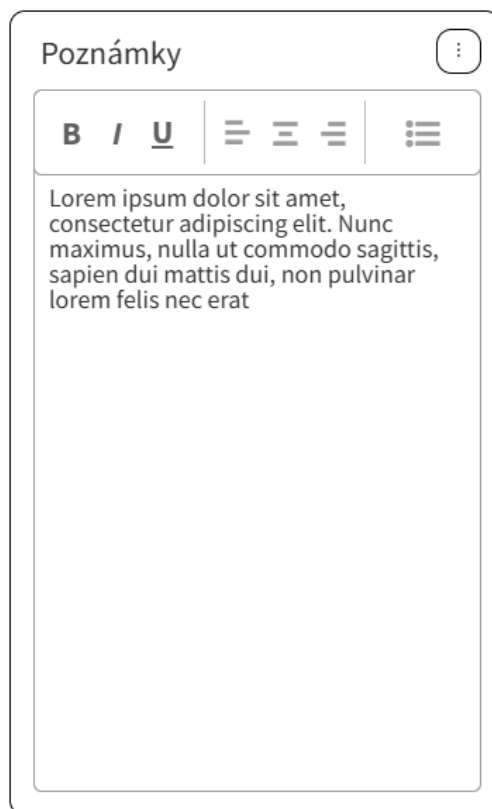


Obrázek 14 Návrh bloku „Kalendář“

*(zdroj: vlastní)*

#### 4.1.8 Blok „Poznámky“

Blok „Poznámky“ je určen pro rychlé vytváření a úpravy textových poznámek. V horní části okna jsou standardní nástroje pro formátování textu: můžete jej zvýraznit tučně, kurzívou nebo podtrhnout. K dispozici je také zarovnání textu podle levého, středového a pravého okraje a také možnost vytvořit seznam s odrážkami. Uživatel si tak strukturuje poznámky, zvýrazňuje důležité body a usnadňuje jejich vnímání.

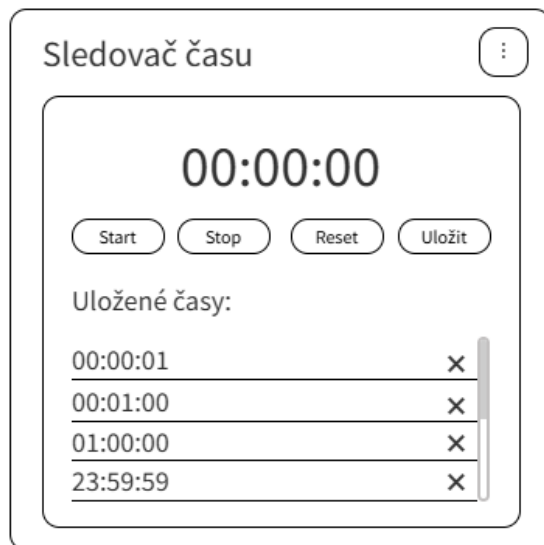


Obrázek 15 Návrh bloku „Poznámky“

*(zdroj: vlastní)*

#### 4.1.9 Blok „Sledovač času“

Blok „Sledovač času“ je určen pro pohodlné měření a ukládání časových intervalů. Uživatel spustí stopky tlačítkem „Start“, pozastaví odpočet tlačítkem „Stop“, může čas vynulovat tlačítkem „Reset“ a naměřený interval uloží tlačítkem „Uložit“. Všechny uložené časové intervaly jsou zobrazeny v seznamu níže. Uživatel může libovolnou uloženou hodnotu smazat kliknutím na odpovídající tlačítko vedle položky.

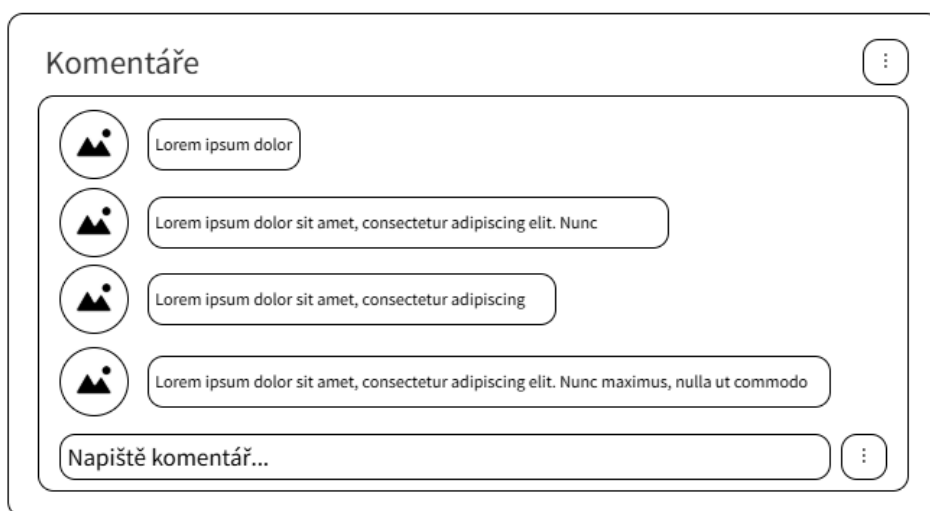


Obrázek 16 Návrh bloku „Sledovač času“

*(zdroj: vlastní)*

#### 4.1.10 Blok „Komentáře“

Blok „Komentáře“ umožňuje uživatelům zanechávat komentáře a diskutovat o detailech pracovního procesu. Zprávy se zobrazují s avatary a textem, seřazenými chronologicky pod sebou. Kromě jednoduchého textu má uživatel také možnost přidávat hypertextové odkazy na další bloky aplikace. K tomu klikne na tlačítko napravo od vstupního pole a vybere požadovaný blok ze seznamu. Po kliknutí na takový odkaz se odpovídající blok na 2 sekundy zvýrazní, což umožňuje rychle pochopit, o čem komentář je.

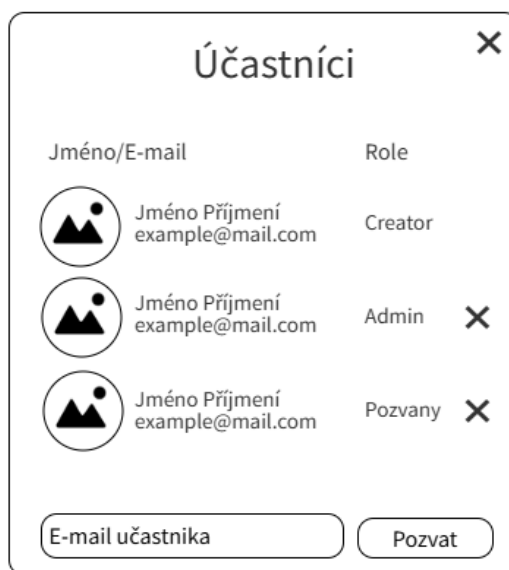


Obrázek 17 Návrh bloku „Komentáře“

*(zdroj: vlastní)*

#### 4.1.11 Menu „Účastníci“

Okno „Účastníci“ slouží ke správě složení účastníků pracovního prostoru. Seznam zobrazuje jména, e-maily a role všech připojených uživatelů. Každý účastník má uvedenou svou roli – například tvůrce, administrátor nebo pozvaný. Uživatel s příslušnými právy může pomocí tlačítka vpravo odebrat libovolného účastníka kromě tvůrce. Ve spodní části okna se nachází pole pro zadání e-mailu. Uživatel může zadat adresu a kliknutím na tlačítko „Pozvat“ odeslat pozvánku novému účastníkovi. Poté se v seznamu zobrazí se statusem „Pozvaný“.

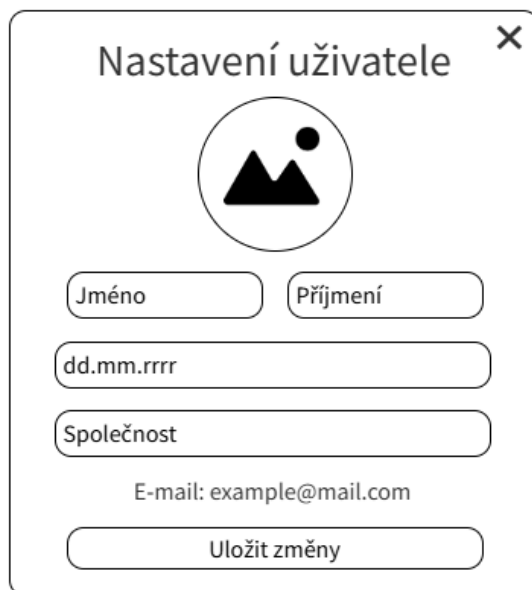


Obrázek 18 Návrh menu „Účastníci“

(zdroj: vlastní)

#### 4.1.12 Nastavení uživatele

Okno „Nastavení uživatele“ umožňuje úpravu osobních údajů uživatele. Zde můžete změnit jméno, příjmení, datum narození a název firmy. Pole pro e-mail se zobrazí, ale zůstává neupravitelné. Avatar v horní části je klikatelný – uživatel na něj může kliknout a nahrát nový obrázek. Po provedení změn je nutné kliknout na tlačítko „Uložit obrázky“, aby se aktualizované údaje uložily.



The image shows a user settings dialog box titled "Nastavení uživatele" with a close button (X) in the top right corner. Inside the dialog, there is a circular profile picture placeholder with a mountain icon. Below the profile picture are two input fields for "Jméno" and "Příjmení". Below these is a date input field with the placeholder "dd.mm.rrrr". Below the date field is a text input field for "Společnost". Below the company field, the email address "E-mail: example@mail.com" is displayed. At the bottom of the dialog is a button labeled "Uložit změny".

Obrázek 19 Návrh menu „Nastavení uživatele“

*(zdroj: vlastní)*

#### 4.1.13 Menu pozvánek

Okno „Pozvánky“ obsahuje seznam pozvánek do pracovních prostorů. U každé pozvánky se zobrazuje název pracovního prostoru, role, se kterou bude uživatel přidán, a jméno a e-mail pozývajícího. Uživatel může pozvánku přijmout kliknutím na tlačítko „Přijmout“ nebo ji odmítnout kliknutím na tlačítko „Odmítnout“. Po přijetí se odpovídající pracovní prostor zpřístupní v hlavním rozhraní.



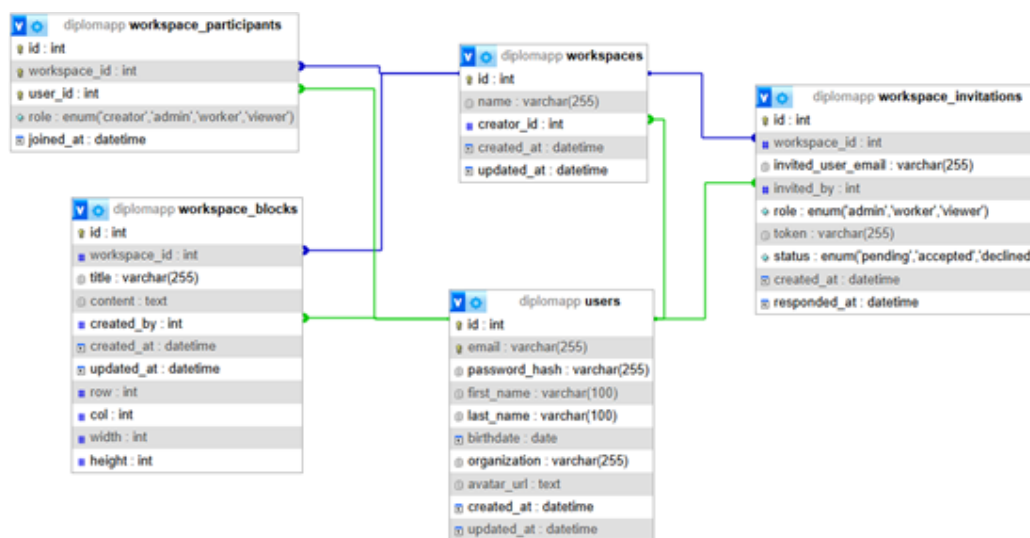
The image shows an invitations dialog box titled "Pozvánky" with a close button (X) in the top right corner. The dialog contains a list of two invitations. Each invitation entry includes a circular profile picture placeholder with a mountain icon, the name "Jméno Příjmení", the email "example@mail.com", the workspace name "Pracovní prostor 1" and "Pracovní prostor 2", and the role "Role: viewer". To the right of each invitation are two buttons: "Přijmout" and "Odmítnout".

Obrázek 20 Návrh menu pozvánek

*(zdroj: vlastní)*

## 4.2 Návrh datové struktury

Pro správný chod webové aplikace a pro správnou interakci mezi uživateli této aplikace bude potřeba databáze. Na základě budoucích cílů webové aplikace jsem vytvořil databázi pomocí online grafického rozhraní pro práci s databázemi Admin. Pomocí tohoto nástroje jsem vytvořil grafické znázornění mé databáze, kde jsou vidět všechny tabulky, jejich sloupce a propojení mezi nimi.



Obrázek 21 Datová struktura projektu

(zdroj: vlastní)

### 4.2.1 Tabulka users

Tabulka users slouží k ukládání základních informací o všech uživateli aplikace. Každý uživatel má přiřazen unikátní identifikátor a další údaje potřebné pro autentizaci a personalizaci prostředí.

Sloupce:

- id — primární klíč, jednoznačně identifikuje uživatele.
- email — e-mailová adresa uživatele, musí být unikátní.
- password\_hash — hash hesla, uložený bezpečně v zašifrované podobě.
- first\_name — jméno uživatele.
- last\_name — příjmení uživatele.
- birthdate — datum narození (volitelně).

- organization — název organizace, kde uživatel působí (volitelné pole).
- avatar\_url — cesta k profilovému obrázku.
- created\_at — datum a čas registrace uživatele.
- updated\_at — datum a čas poslední úpravy údajů.

#### 4.2.2 Tabulka workspaces

Tabulka workspaces představuje jednotlivé pracovní prostory. Každý prostor je samostatným kontejnerem pro obsah, ke kterému mají přístup konkrétní uživatelé. Každý pracovní prostor má svého tvůrce (vazba na users).

Sloupce:

- id — primární klíč.
- name — název pracovního prostoru.
- creator\_id — cizí klíč na users.id, určuje, kdo prostor vytvořil.
- created\_at — datum vytvoření prostoru.
- updated\_at — datum poslední změny.

#### 4.2.3 Tabulka workspace\_blocks

Tabulka workspace\_blocks uchovává informace o jednotlivých blocích umístěných v rámci pracovních prostorů. Každý blok může obsahovat například úkoly, poznámky, grafy nebo jiné interaktivní prvky.

Sloupce:

- id — primární klíč.
- workspace\_id — cizí klíč na workspaces.id, určuje, do kterého prostoru blok patří.
- title — název bloku.
- content — obsah bloku (uložený nejčastěji jako text nebo JSON).
- created\_by — cizí klíč na users.id, kdo blok vytvořil.
- created\_at — datum vytvoření bloku.

- `updated_at` — datum poslední úpravy.
- `row, col` — pozice bloku v gridové mřížce rozvržení.
- `width, height` — rozměry bloku v mřížce.

#### 4.2.4 Tabulka `workspace_invitations`

Tabulka `workspace_invitations` uchovává seznam pozvánek do pracovních prostorů. Slouží k tomu, aby uživatel mohl být pozván do pracovního prostoru. Každá pozvánka má přiřazený jedinečný token a stav.

Sloupce:

- `id` — primární klíč.
- `workspace_id` — cizí klíč na `workspaces.id`, určuje cílový pracovní prostor.
- `invited_user_email` — e-mail pozvaného uživatele.
- `invited_by` — cizí klíč na `users.id`, kdo pozvánku odeslal.
- `role` — role, která bude přidělena po přijetí pozvánky.
- `token` — jedinečný ověřovací kód pro přijetí pozvánky.
- `status` — stav pozvánky: `pending` (čeká), `accepted` (přijato), `declined` (odmítnuto).
- `created_at` — datum vytvoření pozvánky.
- `responded_at` — datum odpovědi na pozvánku (pokud existuje).

#### 4.2.5 Tabulka `workspace_participants`

Tabulka `workspace_participants` propojuje uživatele s pracovními prostory a definuje jejich roli. Jeden uživatel může být členem více prostorů s různými úrovněmi oprávnění.

Sloupce:

- `id` — primární klíč.
- `workspace_id` — cizí klíč na `workspaces.id`, označuje pracovní prostor.
- `user_id` — cizí klíč na `users.id`, označuje konkrétního uživatele.
- `role` — role uživatele v prostoru (`creator`, `admin`, `worker`, `viewer`).

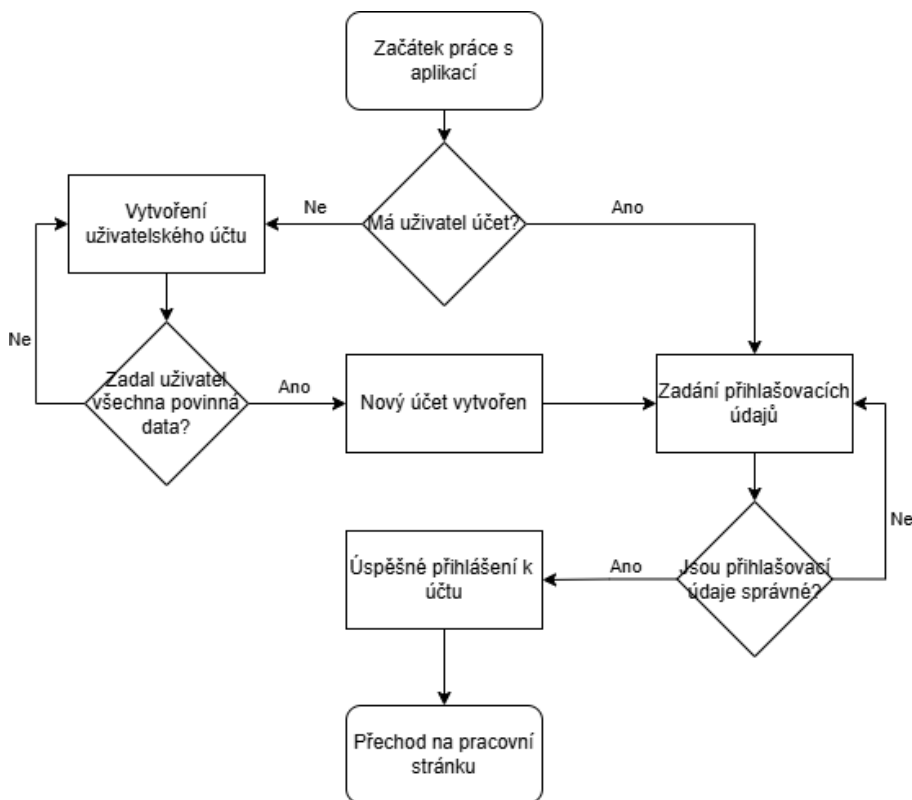
- `joined_at` — datum a čas připojení k prostoru.

### 4.3 Logika a scénáře použití

Než začnu s vývojem aplikace, musím si promyslet její logiku a chování z pohledu uživatele. Důležité je nejen implementovat sadu funkcí, ale také pochopit, jak přesně budou lidé se systémem interagovat. Za tímto účelem vytvořím případy užití – popisy typických uživatelských akcí v různých situacích. Tento přístup pomáhá předem identifikovat potenciální problémy a zajistit, aby aplikace skutečně řešila stanovené úkoly.

#### 4.3.1 Autentizační scénář

Diagram znázorňuje první kontakt uživatele s webovou aplikací. Po spuštění aplikace, pokud uživatel nemá účet, je přesměrován na formulář pro vytvoření účtu, kde vyplní požadovaná pole: e-mail, heslo, křestní jméno a příjmení. Pokud alespoň jedno z těchto polí zůstane prázdné, proces se vrátí na stejný formulář. Jakmile jsou zadány všechny požadované údaje, systém zaznamená zprávu „Vytvořen nový účet“ a přesměruje uživatele na autorizační formulář. Uživatel s existujícím účtem přeskočí krok registrace a přejde rovnou na přihlašovací formulář. Zde zadá svůj e-mail a heslo, načež systém zkontroluje jejich správnost. Nesprávná kombinace přihlašovacího jména a hesla vede k návratu na stejný formulář, což umožňuje opakovat pokus. Úspěšná kontrola vede ke stavu „Úspěšné přihlášení“, po kterém je uživatel automaticky přesměrován na hlavní pracovní stránku aplikace.



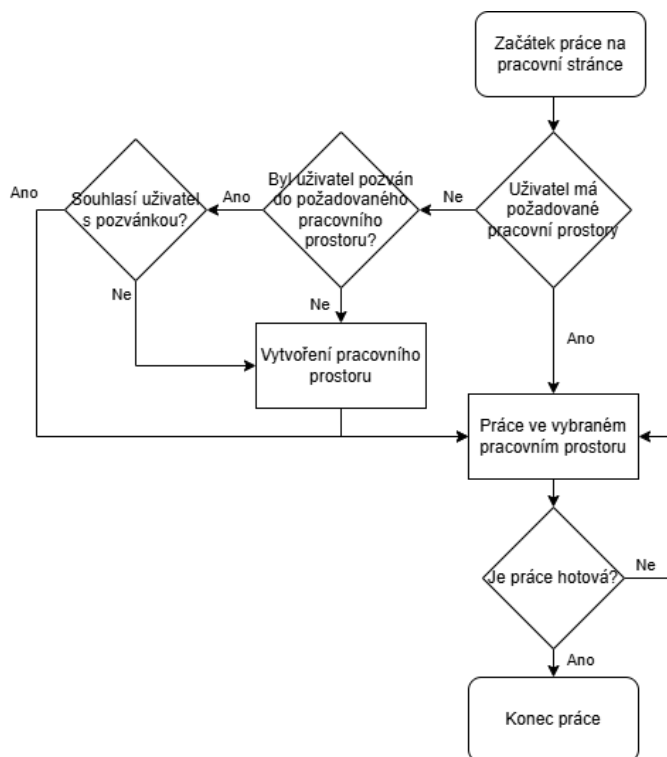
Obrázek 22 Autentizační scénář

(zdroj: vlastní)

#### 4.3.2 Scénář pro práce na pracovní stránce

Scénář začíná otevřením pracovní stránky uživatelem se seznamem jeho pracovních prostorů. Uživatel si prohlédne seznam a zkontroluje, zda požadovaný prostor již existuje. Pokud je odpověď ano, vše se scvrkne na běžnou práci ve vybraném prostoru: uživatel vyřeší zadané úkoly, dokončí je a ukončí práci.

Pokud požadovaný prostor neexistuje, jsou možné dvě větvení. První je, když má uživatel pozvánku. Přijme ji nebo odmítne: souhlas ho automaticky přidá do odpovídajícího prostoru. Druhé větvení nastane, pokud pozvánka neexistuje nebo je odmítnuta. Poté uživatel vytvoří nový pracovní prostor a poté v něm pokračuje v běžné práci. Diagram tedy pokrývá celý cyklus: kontrolu existence požadovaného prostoru, přijetí nebo odmítnutí pozvánky a v případě potřeby nezávislé vytvoření nového prostoru, po kterém se scénář vrací do fáze produktivní práce.



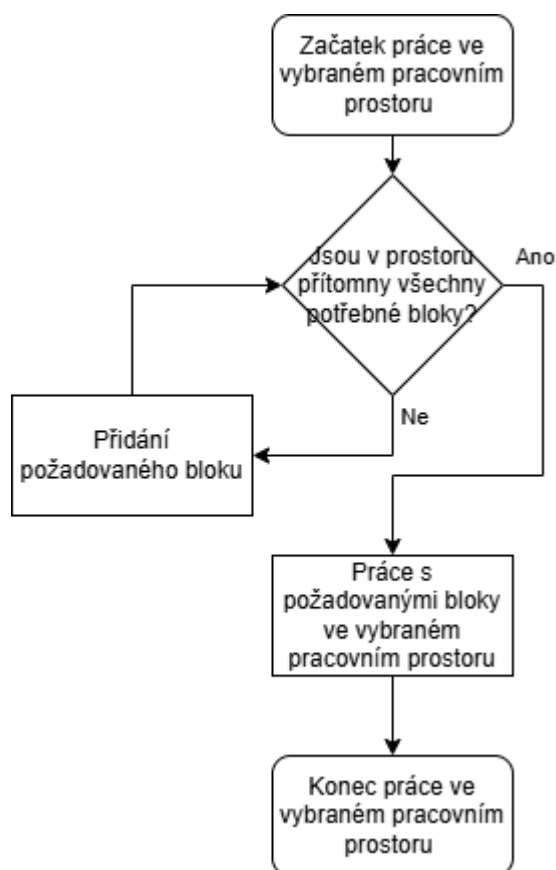
Obrázek 23 Scénář pro práci na pracovní stránce

(zdroj: vlastní)

#### 4.3.3 Scénář pro práci s bloky

Scénář práce s bloky začíná, když si uživatel vybere pracovní prostor pro práci. Nejprve vyhodnotí složení widgetů: zkontroluje, zda jsou přítomny všechny bloky potřebné pro aktuální úkol – například seznam úkolů, plán nebo kalendář. Pokud je požadovaná sada již přítomna, uživatel okamžitě začne pracovat: vyplňuje pole, upravuje data, interaguje s grafy.

Pokud v rozhraní chybí alespoň jeden z potřebných prvků, uživatel přidá odpovídající blok kliknutím na tlačítko „+ Přidat blok“. Po vložení se komponenta okamžitě zobrazí na pracovním prostoru a další činnost pokračuje s plnou sadou nástrojů. Po dokončení úprav uživatel ukončí relaci nebo přepne do jiného prostoru, čímž dokončí práci v aktuálním prostoru.



Obrázek 24 Scénář pro práce s bloky

(zdroj: vlastní)

#### 4.3.4 Scénář pro interakci s uživateli

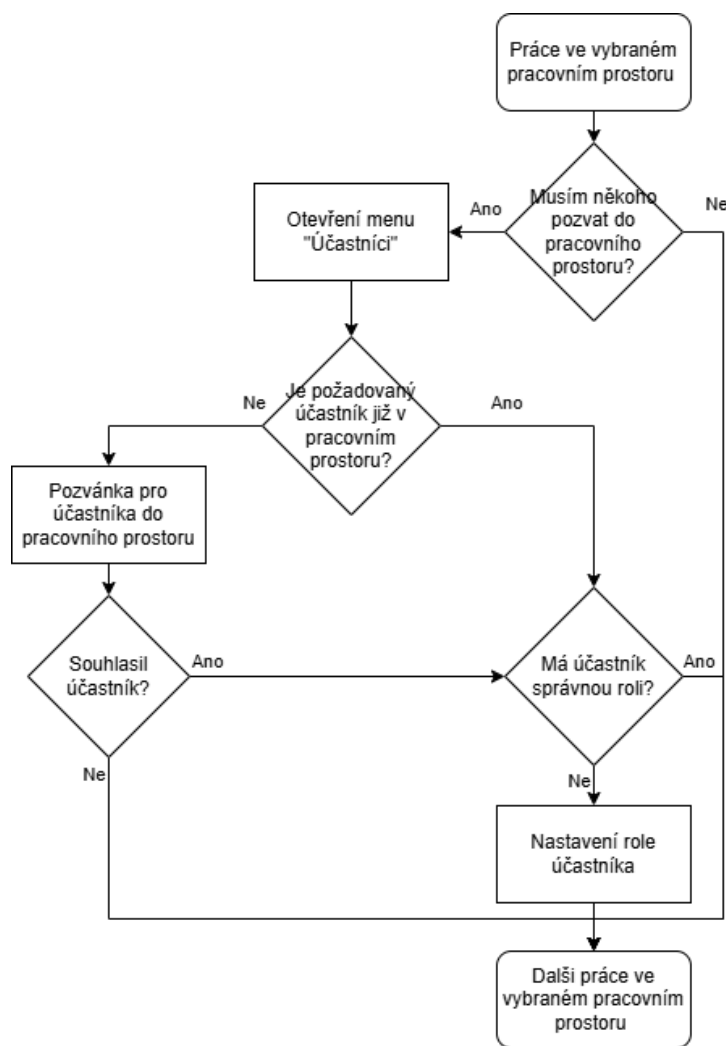
Uživatel při práci ve vybraném pracovním prostoru otevře nabídku „Účastníci“ pro správu týmu.

Pokud je potřeba pozvat novou osobu, ručně zadá svůj e-mail a klikne na tlačítko „Pozvat“. V tomto okamžiku systém vytvoří záznam v tabulce `workspace_invitations`, přiřadí pozvánce jedinečný token a nastaví stav na „čeká na vyřízení“, načež je zadanému uživateli odeslána pozvánka.

Když uživatel pozvánku přijme, je to zaznamenáno: stav se změní na „přijato“ a v tabulce `workspace_participants` se automaticky zobrazí řádek, který propojí nového účastníka s prostorem a okamžitě určí vybranou roli (například administrátor, pracovník nebo čtenář).

Pokud pozvaný klikne na „Odmítnout“, stav pozvánky se změní na „odmítnuto“; v tomto případě se v tabulce `workspace_participants` řádek účastníka nevytvoří. V případě potřeby může pozvánku znovu odeslat: systém vygeneruje nový token a znovu změní

záznam na stav „čeká na vyřízení“. Alternativní scénář nastává, když je požadovaná osoba již v seznamu účastníků – pak se fáze pozvání přeskočí a administrátor v případě potřeby okamžitě přistoupí ke změně pole role. Pokud je role účastníka pro danou práci také správná, pak práce ve vybraném pracovním prostoru jednoduše pokračuje.



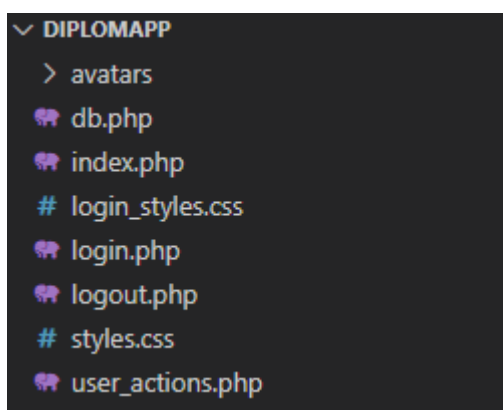
Obrázek 25 Scénář pro interakci s uživateli

(zdroj: vlastní)

## 5 IMPLEMENTACE APLIKACE

### 5.1 Struktura projektu

Aplikace je jeden projekt, jehož struktura je znázorněna na obrázku 24. Struktura projektu je poměrně jednoduchá: všechny serverové skripty a stylové listy se nacházejí v kořenovém adresáři DiplomApp a jediný podadresář avatars je určen pro ukládání stažených profilových obrázků.



Obrázek 26 Struktura projektu

(zdroj: vlastní)

#### 5.1.1 db.php

Soubor db.php je hlavní konfigurační skript zodpovědný za navázání připojení k databázi MySQL. Na začátku souboru jsou definovány čtyři klíčové parametry: adresa serveru (127.0.0.1), název databáze (diplomapp) a přihlašovací údaje uživatele (root / root pro lokální připojení). Dále je nastaveno požadované kódování utf8mb4 pro správné zpracování všech možných znaků, včetně vícebajtových.

Skript vytvoří objekt mysqli se zadaným nastavením a okamžitě nastaví vybrané kódování pomocí metody set\_charset(). Pokud nelze navázat připojení, je provádění programu přerušeno voláním metody die() s chybovou hláškou o připojení. Vzhledem k tomu, že soubor db.php je součástí všech ostatních souborů PHP prostřednictvím require\_once, jsou přístupové parametry uloženy na jednom místě: stačí je jednou změnit, aby se změny projevíly v celé aplikaci.

### 5.1.2 login.php

Soubor login.php kombinuje dvě funkce – ověření existujícího uživatele a kompletní proceduru počáteční registrace. Skript začíná pracovat s příkazem session\_start, aby okamžitě odfiltroval případy, kdy se uživatel již přihlásil: přítomnost identifikátoru v poli \$\_SESSION vede k okamžitému přesměrování na index.php. Při přístupu metodou GET se zobrazí stránka se dvěma záložkami: přihlašovací formulář s poli pro e-mail a heslo a rozšířený registrační formulář, kde se kromě několika řádků pro identifikaci hesla požaduje také křestní jméno, příjmení a e-mailová adresa, zatímco datum narození a název organizace zůstávají volitelné.

Po odeslání dat skript rozlišuje režim pomocí skrytého pole action. Pokud je hodnota login, je řádek uživatele vybrán e-mailem a zadané heslo je porovnáno s uloženým hashem. Pokud je nalezena úspěšná shoda, identifikátor, křestní jméno a příjmení se zapíše do relace a prohlížeč je přesměrován na hlavní stránku. V registračním režimu (action = register) skript postupně kontroluje, zda povinná pole nejsou prázdná, zda adresa odpovídá formátu, zda heslo má alespoň šest znaků a zda odpovídá jeho potvrzení. Poté je proveden dotaz, zda v tabulce uživatelů již taková e-mailová adresa existuje. Pokud neexistuje duplicitní adresa, vygeneruje se hash hesla a vytvoří se nový záznam. Na základě toho se na stránce zobrazí zpráva o úspěšné registraci a uživatel je vyzván k okamžitému přihlášení pomocí vytvořených přihlašovacích údajů. Pokud je e-mail zaneprázdněný nebo neproběhlo ověření, vrátí se stejná stránka se stručným vysvětlením příčiny chyby a zadané hodnoty se uloží, aby uživatel mohl opravit pouze problematická pole.

### 5.1.3 index.php

Soubor index.php je ústřední částí aplikace: sem je uživatel odeslán ihned po úspěšném přihlášení. Skript začíná obnovením relace, aby získal aktuální ID uživatele; pokud není k dispozici, je prohlížeč přesměrován na přihlašovací stránku, která umožňuje přístup k obsahu pouze těm, kteří se již autentizovali. Poté je zahrnut soubor db.php a aplikace načte seznam všech pracovních prostorů, ke kterým má uživatel přístup. Pro každý pracovní prostor se současně načte jeho název a uložená role člena.

Aktuální pracovní prostor je určen parametrem workspace\_id, který je předáván prostřednictvím řetězce dotazu; pokud parametr chybí, je vybrán první ze seznamu. Pro vybraný pracovní prostor skript provede další požadavek, tentokrát na tabulku s bloky, ze které se načtou souřadnice, velikosti a obsah každého widgetu. Výsledné pole je

transformováno do HTML struktury: bloky jsou umístěny uvnitř flexibilní mřížky a jejich obsah – seznamy úkolů, grafy, poznámky – interpretují klientské skripty. Samostatná větev kódu sleduje kliknutí na tlačítko „Přidat pracovní prostor“: po kontrole počtu existujících pracovních prostorů pro tohoto uživatele se vytvoří nový záznam a autor se automaticky stane jeho tvůrcem; poté se stránka okamžitě znovu načte s novým parametrem `workspace_id`, aby uživatel viděl nově vytvořenou oblast. Posledním krokem při spuštění šablony je zahrnutí globálních stylů, po kterém prohlížeč obdrží plně vytvořenou stránku, připravenou pro další interaktivní požadavky prostřednictvím `user_actions.php`.

Kromě zobrazení samotných pracovních prostorů obsahuje `index.php` dvě pomocná rozhraní. Prvním je dialogové okno „Členové“, které se otevře po kliknutí na odpovídající tlačítko; v něm si administrátor může prohlédnout seznam uživatelů, změnit jejich role nebo odeslat nové pozvánky. Druhým je plovoucí nabídka profilu, přístupná kliknutím na avatara: zde uživatel konfiguruje osobní údaje, nahraje nový avatar a v případě potřeby se odhlásí. Obě okna se vykreslují přímo v kódu `index.php` a otevírají se pomocí klientských skriptů bez nutnosti opětovného načítání stránky.

#### 5.1.4 `logout.php`

`logout.php` je malý utilita, která ukončuje uživatelskou relaci. Po volání funkce `session_start()` vymaže všechny proměnné relace pomocí `session_unset()` a smaže samotnou relaci pomocí příkazu `session_destroy()`. Ihned poté skript odešle hlavičku přesměrování `Location: login.php`, která prohlížeči umožní okamžitě přejít na přihlašovací stránku. V důsledku toho je aktuální relace kompletně uzavřena a uživatel je vrácen na místo opětovné autorizace.

#### 5.1.5 `user_actions.php`

Soubor `user_actions.php` plní dvojí funkci. Na jedné straně se jedná o skripty pro všechny operace, při kterých uživatelé vzájemně interagují: pozvánky do pracovních prostorů, přijetí nebo odmítnutí těchto pozvánek, přiřazení nebo změna rolí účastníků, odebrání kolegů z projektu. Na druhé straně skript obsluhuje i obvyklé „nezávislé“ akce uživatelů: vytváření a mazání pracovních prostorů, přidávání nebo konfigurace bloků, nahrávání avatara, aktualizace osobního profilu.

### 5.1.6 styles.css

styles.css je hlavní stylový soubor, který tvoří vizuální design všech stránek aplikace. Nastavuje hlavní barvy, písma a odsazení, popisuje strukturu prvků (boční menu, pracovní plocha, horní panel) a navrhuje interaktivní komponenty – tlačítka, formuláře, blokové karty. Díky tomuto souboru zůstává rozhraní jednotné a správně se škáluje na různých zařízeních.

### 5.1.7 login\_styles.css

Soubor login\_styles.css se používá výhradně na přihlašovací stránce. Tento soubor nastavuje gradientní pozadí pro celý výřez a vycentruje přihlašovací formulář svisle i vodorovně, přičemž kontejner stylizuje pomocí polí: bílé pozadí, zaoblené rohy, stín a pevná minimální šířka. Další pravidla definují velikosti a odsazení prvků formuláře a také barevné efekty při najetí myši na tlačítka. Soubor login\_styles.css tak izoluje design přihlašovací stránky od celkového stylového listu, což umožňuje změnit jeho design bez ovlivnění ostatních částí aplikace.

### 5.1.8 avatars/.htaccess

Soubor obsahuje direktivy Apache, které zajišťují bezpečné stahování adresáře avatars/:

```
php_flag engine off
RemoveHandler .php .phtml .php3 .php4 .php5 .php7 .phps
AddType text/plain .php .phtml .php3 .php4 .php5 .php7 .phps
```

Takže i když se útočník pokusí nahrát „avatar“ s příponou .php a poté k němu přistupovat přes prohlížeč, soubor se nespustí, ale zobrazí se jako prostý text. [30]

## 5.2 Ukázka finálního rozhraní

Finální vzhled aplikace se ukázal být téměř přesně stejný, jaký byl původně zamýšlen při tvorbě rozvržení vzhledu aplikace. Celá aplikace je provedena v jemném stylu bez ostrých rohů a v teplé oranžové barevné schémě. Aplikace vypadá hezky a srozumitelně.

### 5.2.1 Registrace

Uprostřed na oranžovém gradientním pozadí je umístěna bílý obdélníkový registrační formulář se zaoblenými rohy. Formulář obsahuje postupně: nadpis „Registrace“, sedm textových polí (E-mail\*, Datum\* s poznámkou „min. 6 znaků“, Datum narození\*, Datum\*, Datum narození\*, Datum narození s ikonou kalendáře, Organizace), tlačítko „Registrovat“

přes celou šířku s sotva znatelným gradientem a odkaz „← Registrace“. Povinná pole jsou označena hvězdičkou a v polích jsou uvedeny nápovědy.

The image shows a registration form titled "Registrace" (Registration) enclosed in an orange border. The form contains the following fields and elements:

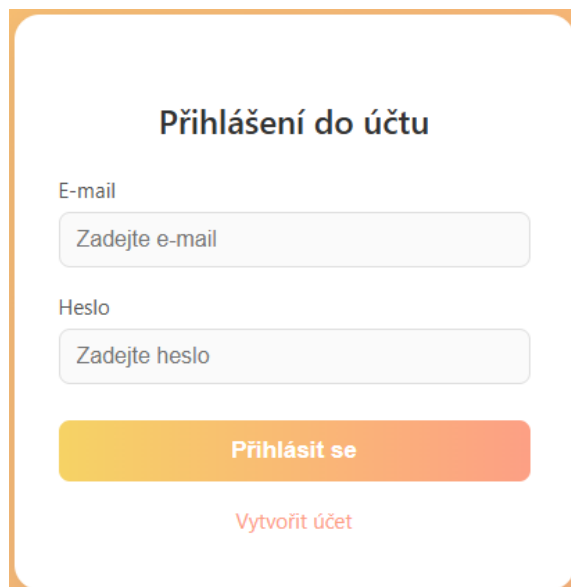
- E-mail\***: A text input field with the placeholder "Zadejte e-mail".
- Heslo\* (min. 6 znaků)**: A text input field with the placeholder "Zadejte heslo".
- Potvrďte heslo\***: A text input field with the placeholder "Zadejte heslo znovu".
- Jméno\***: A text input field with the placeholder "Zadejte jméno".
- Příjmení\***: A text input field with the placeholder "Zadejte příjmení".
- Datum narození**: A date input field with the placeholder "ДД . ММ . ГГГГ" and a calendar icon.
- Organizace**: A text input field with the placeholder "Zadejte organizaci".
- Registrovat se**: A large orange button with a gradient.
- [← Zpět na přihlášení](#): A blue link below the registration button.

Obrázek 27 Registrační formulář

(zdroj: vlastní)

### 5.2.2 Přihlášení do účtu

Uprostřed na oranžovém gradientním pozadí je bílý obdélník se zaoblenými rohy. Formulář obsahuje postupně: nadpis „Přihlášení do účtu“, textová pole „E-mail“ a „Heslo“, široké tlačítko „Přihlásit se“ a textový odkaz „Vytvořit účet“.



Přihlášení do účtu

E-mail

Zadejte e-mail

Heslo

Zadejte heslo

Přihlásit se

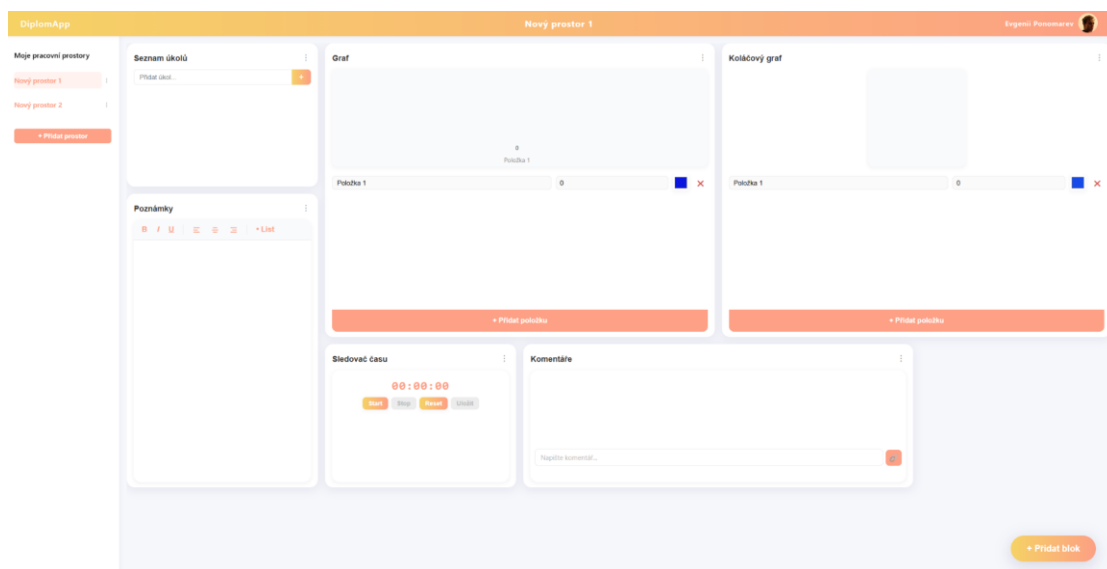
Vytvořit účet

Obrázek 28 Přihlašovací formulář

*(zdroj: vlastní)*

### 5.2.3 Pracovní stránka

Levou stranu obrazovky zabírá boční nabídka se seznamem pracovních prostorů: aktuální pracovní prostor je zvýrazněn a pod seznamem je tlačítko „+ Přidat prostor“ pro vytvoření nového. Kliknutím na tlačítko nabídky vedle názvu pracovního prostoru se otevře kontextová nabídka s možnostmi pro přejmenování, zobrazení účastníků a smazání pracovního prostoru. V horním oranžovém panelu vlevo je zobrazeno logo „DiplomApp“, uprostřed je název aktivního pracovního prostoru („Nový prostor 1“) a vpravo je uživatelské jméno a avatar. Kliknutím na avatara uživatele se otevře nabídka s položkami „Nastavení uživatele“, „Pozvánky“ a „Ukončit“. Hlavní oblast je vyplněna bloky různého účelu. V pravém dolním rohu obrazovky se nachází plovoucí tlačítko „+ Přidat blok“, které umožňuje přidat nový blok vybraného typu.



Obrázek 29 Pracovní stránka

*(zdroj: vlastní)*

#### 5.2.4 Blok „Seznam úkolů“

V horní části bloku „Seznam úkolů“ se v horním řádku zobrazuje velkým písmem jeho název. Vpravo se nachází tlačítko, které po kliknutí otevře kontextovou nabídku s možnostmi „Přejmenovat“ a „Smazat“. Pod názvem se nachází vstupní pole a napravo od něj tlačítko „+“ pro přidání písemného úkolu do seznamu. Níže je uveden seznam úkolů, jehož každý řádek obsahuje text úkolu a vpravo dvě ikony: zaškrtnutí pro označení splněného úkolu a křížek pro jeho smazání. Celý blok je navržen jako bílý obdélník s mírně zaoblenými rohy a podél pravého okraje je viditelná posuvní lišta, pokud je úkolů větší počet.



Obrázek 30 Blok „Seznam úkolů“

*(zdroj: vlastní)*

### 5.2.5 Blok „Graf“

V horní části bloku „Graf“ se v horním řádku zobrazuje velkým písmem jeho název. Vpravo se nachází tlačítko, které po kliknutí otevře kontextovou nabídku s možnostmi „Přejmenovat“ a „Smazat“. Pod ním je umístěn sloupcový graf, kde každá položka představuje jeden barevný sloupec; výška sloupce vizuálně znázorňuje jeho číselnou hodnotu a pod sloupcem se zobrazuje popisek s názvem položky. Pod grafem se nachází propojený seznam položek – každá řádka obsahuje název položky, číselné pole s její hodnotou a barevné tlačítko pro změnu barvy sloupce; vedle tohoto tlačítka je ikona pro odstranění položky. V dolní části bloku je široké oranžové tlačítko „+ Přidat položku“, kterým lze do seznamu a tím i do grafu vložit nový záznam.

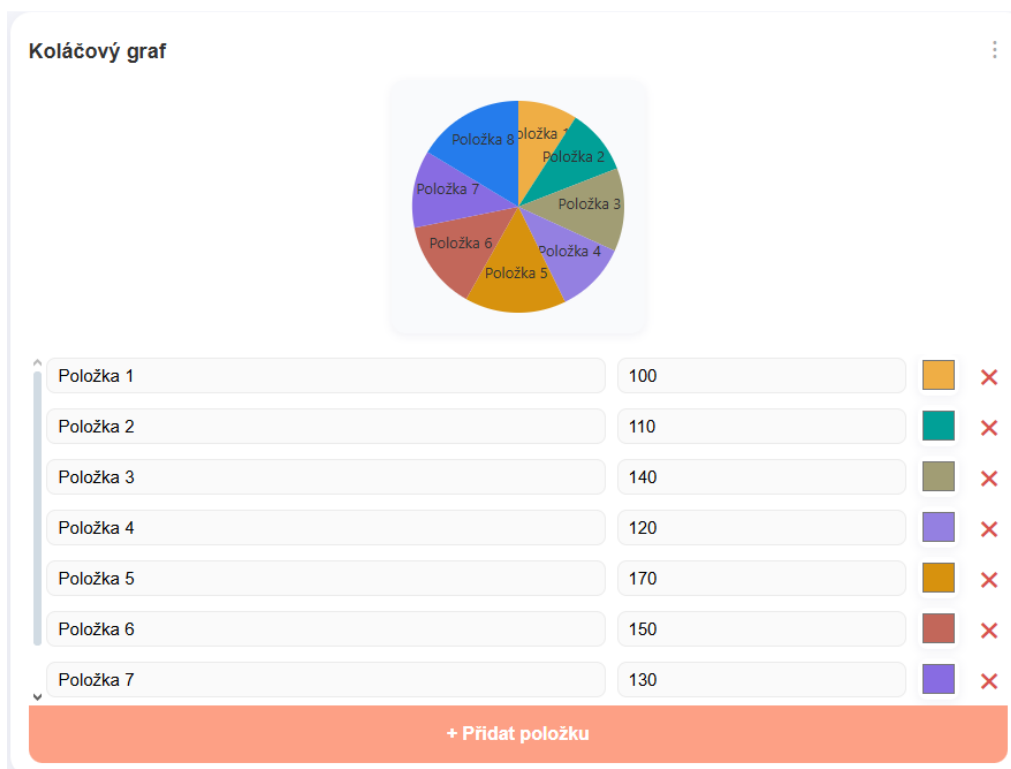


Obrázek 31 Blok „Graf“

*(zdroj: vlastní)*

### 5.2.6 Blok „Koláčový graf“

V horní části bloku „Koláčový graf“ se v horním řádku zobrazuje velkým písmem jeho název. Vpravo se nachází tlačítko, které po kliknutí otevře kontextovou nabídku s možnostmi „Přejmenovat“ a „Smazat“. Pod ním je umístěn koláčový graf, kde každá položka odpovídá jednomu barevnému výseku; velikost výseku vizuálně znázorňuje jeho číselnou hodnotu a přímo na výseku je uveden popisek s názvem položky. Pod samotným grafem se nachází propojený seznam položek – každá řádka obsahuje textové pole s názvem položky, vedle něj číselné pole s její hodnotou a barevné tlačítko pro změnu barvy výseku; vpravo od tohoto tlačítka je ikona „×“ pro smazání položky. V dolní části bloku je široké oranžové tlačítko „+ Přidat položku“, kterým lze do seznamu a tím i do grafu vložit nový záznam.

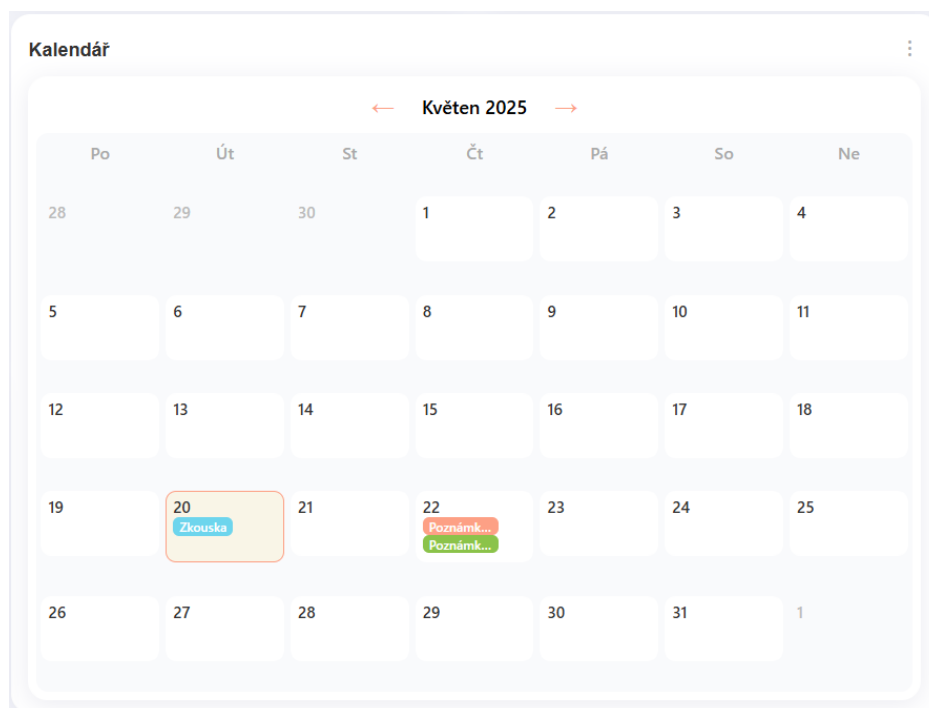


Obrázek 32 Blok „Koláčový graf“

(zdroj: vlastní)

### 5.2.7 Blok „Kalendář“

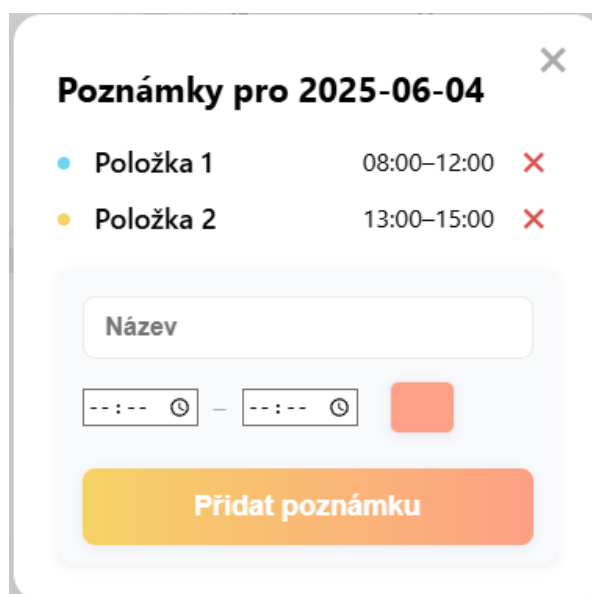
Blok „Kalendář“ umožňuje uživateli zobrazit data a zanechávat poznámky k jednotlivým dnům. V horní části bloku jsou tlačítka pro přepínání mezi měsíci a zobrazuje se aktuální měsíc a rok. Při výběru dne v kalendáři si uživatel může vytvořit nebo zobrazit již vytvořené poznámky k tomuto datu. Kalendář tak slouží jako praktický nástroj pro plánování a sledování událostí.



Obrázek 33 Blok „Kalendář“

*(zdroj: vlastní)*

Při kliknutí na den se otevře vyskakovací okno s nadpisem data. Obsahuje seznam existujících poznámek s možností je odstranit, pole pro zadání názvu nové poznámky, výběr časového intervalu (od–do) a volbu barvy. Ve spodní části okna je tlačítko „Přidat poznámku“ pro uložení nové položky.

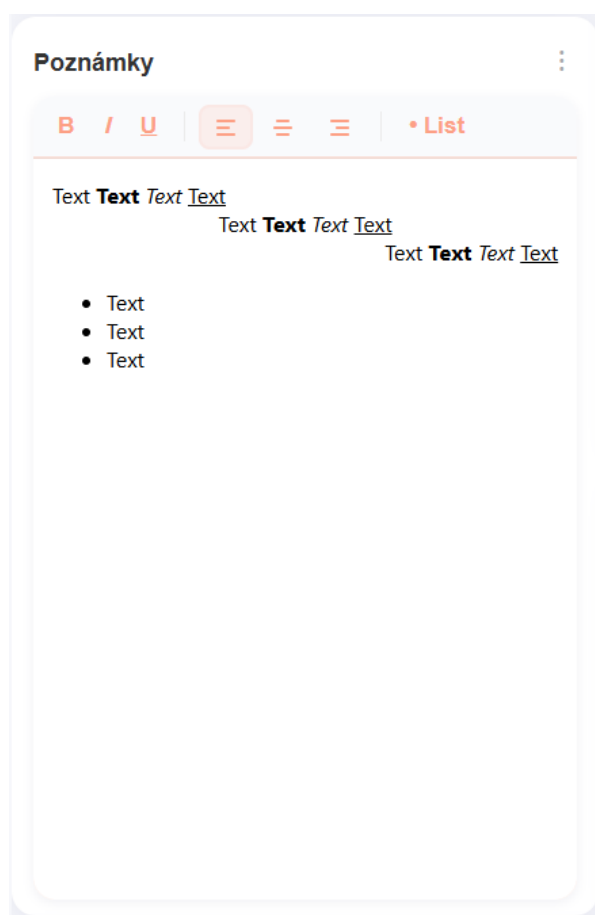


Obrázek 34 Blok „Kalendář“

*(zdroj: vlastní)*

### 5.2.8 Blok „Poznámky“

V horní části bloku „Poznámky“ se v horním řádku zobrazuje velkým písmem jeho název. Vpravo se nachází tlačítko, které po kliknutí otevře kontextovou nabídku s možnostmi „Přejmenovat“ a „Smazat“. Pod ním je umístěn řádek formátovacích nástrojů: tučné písmo, kurzíva, podtržení, zarovnání a přepínání do režimu seznamu. Samotná oblast pro vkládání textu představuje bílé prázdné pole se světle šedou obvodovou linkou a mírným zaoblením rohů. Uživatel zde napíše nebo vloží poznámky, které lze formátovat pomocí horního panelu nástrojů.



Obrázek 35 Blok „Poznámky“

(zdroj: vlastní)

### 5.2.9 Blok „Sledovač času“

V horní části bloku „Sledovač času“ se v horním řádku zobrazuje velkým písmem jeho název. Vpravo se nachází tlačítko, které po kliknutí otevře kontextovou nabídku s možnostmi „Přejmenovat“ a „Smazat“. Pod názvem je uprostřed digitální displej ukazující aktuální čas. Pod ním jsou čtyři tlačítka pro ovládání: „Start“, „Stop“, „Reset“ a „Uložit“. V

dolní části bloku je uveden nadpis „Uložené časy:“ a pod ním seznam zaznamenaných časů, zobrazených každá řádka s časovou hodnotou a červenou ikonou „×“ pro smazání. Při delším seznamu se vpravo objeví posuvník.

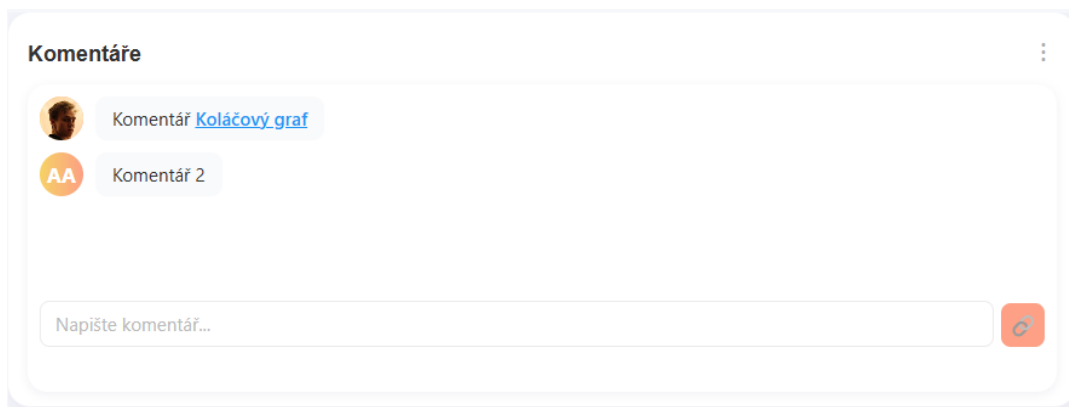


Obrázek 36 Blok „Sledovač času“

(zdroj: vlastní)

#### 5.2.10 Blok „Komentáře“

V horní části bloku „Komentáře“ se v horním řádku zobrazuje velkým písmem jeho název. Vpravo se nachází tlačítko, které po kliknutí otevře kontextovou nabídku s možnostmi „Přejmenovat“ a „Smazat“. Pod názvem jsou uvedeny jednotlivé komentáře – každý v samostatném poli s kulatým avatárem vlevo, za kterým následuje vlastní text komentáře. Pokud kurzorem najedete na kterýkoli avatar, zobrazí se nad ním popup s plným jménem uživatele a jeho e-mailem. Uživatelé mohou také odkazovat na konkrétní blok. Ve spodní části bloku je bílé vstupní pole a napravo od něj tlačítko pro odkazování na blok.

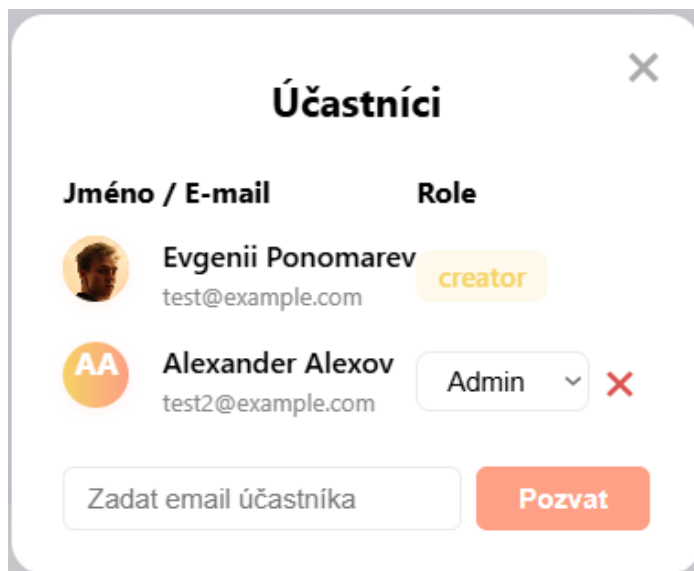


Obrázek 37 Blok „Komentáře“

*(zdroj: vlastní)*

### 5.2.11 Menu „Účastníci“

V horní části okna „Účastníci“ se zobrazuje název velkým písmem a vpravo ikona křížku pro zavření. Pod ním jsou dvě sloupce s hlavičkami „Jméno / E-mail“ a „Role“. V seznamu účastníků je každá osoba uvedena v samostatném řádku: vlevo kruhový avatar s nápisem plného jména a e-mailu, napravo štítek s její rolí. U dalších uživatelů je menu pro výběr role a červená ikona křížku pro odebrání z prostoru. Ve spodní části okna se nachází vstupní pole pro zadání e-mailu uživatele a vedle něj tlačítko „Pozvat“ pro odeslání pozvánky.

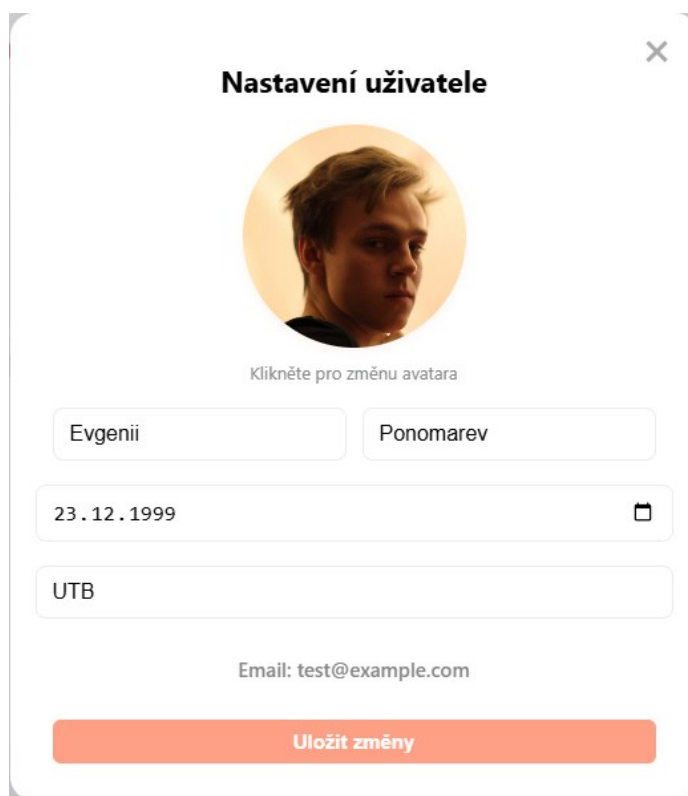


Obrázek 38 Okno „Účastníci“

*(zdroj: vlastní)*

### 5.2.12 Nastavení uživatele

V horní části okna „Nastavení uživatele“ se zobrazí název velkým písmem a v pravém rohu ikona křížku pro zavření. Pod ním je uprostřed kruhový avatar, pod kterým je nápis „Klikněte pro změnu avatara“. Následují dvě vedle sebe umístěná textová pole pro jméno a příjmení. Pod nimi je pole pro datum narození. Dále se nachází pole pro zadání organizace a pod ním šedě zobrazený e-mail, který není editovatelný. V dolní části okna je široké oranžové tlačítko „Uložit změny“ pro uložení úprav.

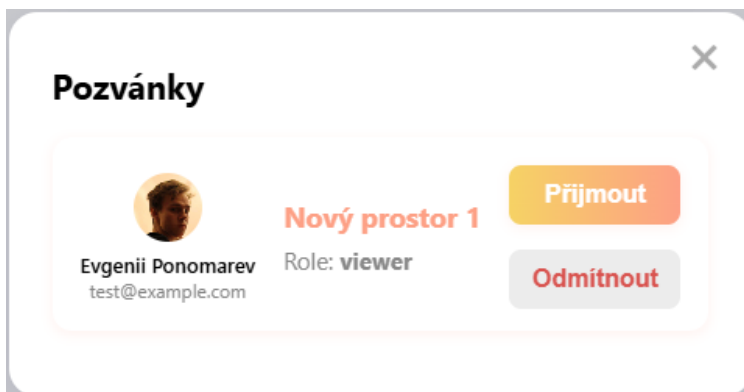


Obrázek 39 Okno „Nastavení uživatele“

(zdroj: vlastní)

### 5.2.13 Menu pozváněk

V horní části okna „Pozvánky“ je zobrazen název velkým písmem a vpravo ikona křížku pro zavření. Pod ním se nachází blok s informacemi o pozvánce: v levé části kruhový avatar a pod ním jméno odesílatele a jeho e-mail. Vpravo od toho je uveden název pracovního prostoru a role uživatele. Vedle těchto informací jsou dvě tlačítka: „Přijmout“ a „Odmítnout“.



Obrázek 40 Návrh menu pozvánek

*(zdroj: vlastní)*

## 5.3 Ochrana před zranitelnostmi

### 5.3.1 SQL injekce

Projekt provádí všechna volání databáze pomocí připravených příkazů `mysqli->prepare()`. Zástupné symboly jsou vázány pomocí metody `bind_param`, takže uživatelský vstup není nikdy přímo vkládán do textu dotazu. Přímé SQL injections jsou tedy v aktuální implementaci vyloučeny.

### 5.3.2 Bezpečné uložení hesel

Uživatelská hesla se neukládají v otevřeném textu. Během registrace se volá `password_hash()` (algoritmus BCRYPT) a během přihlášení — `password_verify()`. To splňuje moderní doporučení OWASP a eliminuje možnost obnovení původního hesla z databáze.

### 5.3.3 Zobrazování systémových chyb

V rané fázi vývoje měl soubor `user_actions.php` nahoře následující řádky:

```
ini_set('display_errors', 1);  
ini_set('display_startup_errors', 1);  
error_reporting(E_ALL);
```

Tyto direktivy by vypisovaly všechna varování a zprávy PHP přímo do prohlížeče, což by odhalilo vnitřní strukturu aplikace. Ve finální verzi byly tyto direktivy zcela odstraněny, takže i když dojde k výjimce, uživatel neobdrží technické podrobnosti.

### 5.3.4 Nesprávné zpracování nahraných souborů

Avatary se ukládají do adresáře avatars, který přijímá pouze obrázky s příponami .jpg a .png. Aby se dále zabránilo spuštění načtených stránek .php, je do tohoto adresáře umístěn konfigurační soubor Apache s následujícími direktivami:

```
php_flag engine off
RemoveHandler .php .phtml .php3 .php4 .php5 .php7 .phps
AddType text/plain .php .phtml .php3 .php4 .php5 .php7 .phps
```

První řádek zcela zakáže interpret PHP ve složce avatars, druhý odstraní obslužnou rutinu pro potenciálně nebezpečné přípony a třetí násilně deklaruje takové soubory jako prostý text. V důsledku toho, i když se útočník pokusí načíst skript pod rouškou obrázku, Apache jej naservíruje jako statický soubor bez spuštění kódu. Zranitelnost spojená se spuštěním nahraných souborů je tedy považována za uzavřenou.

### 5.3.5 Útoky CSRF

Finální verze projektu má základní mechanismus ochrany proti padělání požadavků mezi weby. Při vytvoření relace je kryptografickou funkcí random\_bytes() vygenerován unikátní token \$\_SESSION['csrf\_token']. Tento token je automaticky vložen jako skryté pole do všech kritických HTML formulářů zobrazených na stránce index.php.

Při každém POST požadavku soubor obslužné rutiny (user\_actions.php a další skripty, pokud je to nutné) porovnává přijatou hodnotu pole csrf\_token s tokenem uloženým v relaci. Neshoda má za následek okamžité odmítnutí požadavku a vrácení chybové zprávy. Tento přístup blokuje pokusy webu třetí strany o odeslání falešného formuláře jménem autorizovaného uživatele.

### 5.3.6 Cross-site scripting (XSS)

Všude, kde jsou uživatelská data vypisována na stránku, je aplikováno escapování pomocí htmlspecialchars (PHP) nebo funkce escapeHTML (JavaScript). Tím se zcela zabrání vkládání a spouštění škodlivého kódu prostřednictvím XSS. V důsledku toho se veškerá uživatelská data zobrazují pouze jako text, nikoli jako HTML nebo skript.

## 6 TESTOVÁNÍ APLIKACE

Při dokončování vývoje je důležité zajistit, aby aplikace nejen plnila deklarovanou funkcionalitu, ale také zůstala chráněna před typickými útoky. Tato kapitola proto představuje dvě doplňkové metody testování. Nejprve se provádí funkční testování: krok za krokem se reprodukuje klíčové scénáře. Poté se provádí bezpečnostní testování, zaměřené na praktické ověření implementovaných ochranných opatření proti SQL injections, CSRF, XSS a nesprávně zpracovaným nahrávaním souborů.

Toto dvoufázové testování – funkční a bezpečnostní – poskytuje komplexní posouzení kvality vyvinutého systému. Samostatné podkapitoly níže popisují metodiku, průběh a výsledky každého testu a také vyvozují závěry o tom, jak aplikace funguje.

### 6.1 Funkční testování

Pro funkční testování webové aplikace se musím pokusit reprodukovat pracovní scénáře uživatele s mou webovou aplikací a zjistit výsledky, které budou získány v důsledku interakce a provádění hlavních funkcí webové aplikace.

#### 6.1.1 Testování registrace

Pro otestování registrace otestuji dva scénáře: úspěšnou registraci, když si vytvoříme nový účet, a neúspěšnou registraci, když účet s takovým e-mailem v systému již existuje.

Tabulka 1 Testování úspěšné registrace

| Pole               | Popis                                                                                                                               |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| Cíl                | zaregistrovat nový účet                                                                                                             |
| Předpoklady        | Stránka login.php je otevřená v prohlížeči.<br>Tabulka users neobsahuje záznam s e-mailem test@example.com                          |
| Kroky              | 1. Přejít na záložku „Registrace“.<br>2. Vyplnit všechna povinná pole uvedenými údaji.<br>3. Kliknout na tlačítko „Registrovat se“. |
| Očekávaný výsledek | Systém vytvoří záznam o uživateli, zobrazí zprávu „Registrace úspěšná“ a prohlížeč je přesměrován na přihlašovací formulář.         |
| Skutečný výsledek  | Systém vytvořil záznam o uživateli, zobrazil zprávu „Registrace úspěšná“ a prohlížeč byl přesměrován na přihlašovací formulář.      |
| Stav               | Cíle bylo dosaženo                                                                                                                  |

(zdroj: vlastní)

Tabulka 2 Testování neúspěšné registrace

| Pole               | Popis                                                                                                                               |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| Cíl                | dostat chybu registrace                                                                                                             |
| Předpoklady        | Stránka login.php je otevřená v prohlížeči.<br>Tabulka users obsahuje záznam s e-mailem test@example.com                            |
| Kroky              | 1. Přejít na záložku „Registrace“.<br>2. Vyplnit všechna povinná pole uvedenými údaji.<br>3. Kliknout na tlačítko „Registrovat se“. |
| Očekávaný výsledek | Systém se nezaregistruje a vrátí chybu.                                                                                             |
| Skutečný výsledek  | Systém vrátil zprávu „Tento e-mail je již registrován“ a registrace se nezdařila.                                                   |
| Stav               | Cíle bylo dosaženo                                                                                                                  |

(zdroj: vlastní)

### 6.1.2 Testování autorizace

Pro autorizaci otestuji dva scénáře: úspěšnou autorizaci, kdy byla data zadána správně, a kdy je v údajích o účtu chyba.

Tabulka 3 Testování úspěšné autorizace

| Pole               | Popis                                                                                                                        |
|--------------------|------------------------------------------------------------------------------------------------------------------------------|
| Cíl                | Přihlásit se k vytvořenému účtu                                                                                              |
| Předpoklady        | Stránka login.php je otevřená v prohlížeči.<br>Tabulka users obsahuje záznam s e-mailem test@example.com a helsem qwel23     |
| Kroky              | 1. Přejít na záložku „Přihlášení do účtu“.<br>2. Vyplnit pole pro e-mail a heslo.<br>3. Kliknout na tlačítko „Přihlásit se“. |
| Očekávaný výsledek | Uživatel se přihlásí ke svému účtu a bude přesměrován na pracovní stránku.                                                   |
| Skutečný výsledek  | Uživatel se přihlásil ke svému účtu a byl přesměrován na pracovní stránku.                                                   |
| Stav               | Cíle bylo dosaženo                                                                                                           |

(zdroj: vlastní)

Tabulka 4 Testování neúspěšné autorizace

| Pole               | Popis                                                                                                                                  |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| Cíl                | Získat chybu přihlášení                                                                                                                |
| Předpoklady        | Stránka login.php je otevřená v prohlížeči. Tabulka users obsahuje záznam s e-mailem <u>test@example.com</u> a heslem qwe123           |
| Kroky              | 1. Přejít na záložku „Přihlášení do účtu“.<br>2. Vyplnit s hýbou v poli pro e-mail a heslo.<br>3. Kliknout na tlačítko „Přihlasit se“. |
| Očekávaný výsledek | Uživatel obdrží zprávu o chybě v zadaných údajích.                                                                                     |
| Skutečný výsledek  | Uživatel obdržel zprávu „Nesprávné heslo.“                                                                                             |
| Stav               | Cíle bylo dosaženo                                                                                                                     |

(zdroj: vlastní)

### 6.1.3 Testování práce s pracovními prostory

Pro otestování pracovních prostorů otestuji možnost přidat nový pracovní prostor, přejmenovat pracovní prostor, přepínat mezi pracovními prostory a mazat pracovní prostory.

Tabulka 5 Testování vytvoření pracovního prostoru

| Pole               | Popis                                                                            |
|--------------------|----------------------------------------------------------------------------------|
| Cíl                | Vytvořit nový pracovní prostor                                                   |
| Předpoklady        | Stránka index.php je otevřená v prohlížeči. Uživatel se přihlásil ke svému účtu. |
| Kroky              | 1. Stisknout tlačítko „+ Přidat prostor“.                                        |
| Očekávaný výsledek | V bočním menu se zobrazí nový pracovní prostor.                                  |
| Skutečný výsledek  | V bočním menu se objevil nový pracovní prostor s názvem „Pracovní prostor 1“.    |
| Stav               | Cíle bylo dosaženo                                                               |

(zdroj: vlastní)

Tabulka 6 Testování přejmenování pracovního prostoru

| Pole               | Popis                                                                                                                                                                   |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cíl                | Přejmenovat pracovní prostor                                                                                                                                            |
| Předpoklady        | Stránka index.php je otevřená v prohlížeči.<br>Uživatel se přihlásil ke svému účtu.<br>V bočním menu je alespoň jeden pracovní prostor.                                 |
| Kroky              | 1. Kliknout na tlačítko napravo od tlačítka pracovního prostoru.<br>2. V rozbalovací nabídce vyberte možnost „přejmenovat“<br>3. Napsat nový název pracovního prostoru. |
| Očekávaný výsledek | Název pracovního prostoru se změní.                                                                                                                                     |
| Skutečný výsledek  | Pracovní prostor se nyní nazývá „Nový název“                                                                                                                            |
| Stav               | Cíle bylo dosaženo                                                                                                                                                      |

(zdroj: vlastní)

Tabulka 7 Testování přepínání mezi pracovními prostory

| Pole               | Popis                                                                                                                                    |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| Cíl                | Přepnout na jiný pracovní prostor                                                                                                        |
| Předpoklady        | Stránka index.php je otevřená v prohlížeči.<br>Uživatel se přihlásil ke svému účtu.<br>V bočním menu je alespoň dva pracovních prostoru. |
| Kroky              | V bočním menu kliknout na tlačítko neaktivního pracovního prostoru                                                                       |
| Očekávaný výsledek | Systém zobrazí další pracovní prostor.                                                                                                   |
| Skutečný výsledek  | Systém zobrazuje pracovní prostor „Pracovní prostor 2“                                                                                   |
| Stav               | Cíle bylo dosaženo                                                                                                                       |

(zdroj: vlastní)

Tabulka 8 Testování smazání pracovního prostoru

| Pole               | Popis                                                                                                                                                            |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cíl                | Smazat pracovní prostor                                                                                                                                          |
| Předpoklady        | Stránka index.php je otevřená v prohlížeči.<br>Uživatel se přihlásil ke svému účtu.<br>V bočním menu je alespoň jeden pracovní prostor.                          |
| Kroky              | 1. Kliknout na tlačítko napravo od tlačítka pracovního prostoru.<br>2. V rozbalovací nabídce vybrat možnost „Smazat“<br>3. Potvrdit smazání pracovního prostoru. |
| Očekávaný výsledek | Pracovní prostor bude smazán.                                                                                                                                    |
| Skutečný výsledek  | Pracovní prostor byl smazán.                                                                                                                                     |
| Stav               | Cíle bylo dosaženo                                                                                                                                               |

(zdroj: vlastní)

#### 6.1.4 Testování bloků

Je nutné otestovat možnost přidávání bloků do pracovního prostoru, odstraňovat bloky z pracovního prostoru a také možnost práce s daty v těchto blocích.

Tabulka 9 Testování možnosti přidávání bloků

| Pole               | Popis                                                                                                                                                                  |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cíl                | Přidat do pracovního prostoru všechny typy bloků.                                                                                                                      |
| Předpoklady        | Stránka index.php je otevřená v prohlížeči.<br>Uživatel se přihlásil ke svému účtu.<br>Byl vybrán aktuální pracovní prostor.                                           |
| Kroky              | 1. Kliknout na tlačítko „+ Přidat blok“.<br>2. Vybrat jeden z dříve nepřidaných bloků<br>3. Opakovat, dokud nebudou do pracovního prostoru přidány všechny typy bloků. |
| Očekávaný výsledek | Všechny typy bloků budou umístěny na vybraném pracovním prostoru.                                                                                                      |
| Skutečný výsledek  | Všechny možné bloky se objevily na pracovním prostoru                                                                                                                  |
| Stav               | Cíle bylo dosaženo                                                                                                                                                     |

(zdroj: vlastní)

Tabulka 10 Testování schopnosti správně pracovat s bloky

| Pole               | Popis                                                                                                                                                                                  |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cíl                | Změna dat všech typů bloků bez chyb.                                                                                                                                                   |
| Předpoklady        | Stránka index.php je otevřená v prohlížeči.<br>Uživatel se přihlásil ke svému účtu.<br>Byl vybrán aktuální pracovní prostor.<br>Aktuální pracovní prostor obsahuje všechny typy bloků. |
| Kroky              | 1. Změnit data ve všech bloků.<br>2. Opakovat, dokud se data nezmění ve všech typech bloků.                                                                                            |
| Očekávaný výsledek | Všechna data v blocích se změni dle očekávání bez problémů a chyb.                                                                                                                     |
| Skutečný výsledek  | Všechna data v blocích byla změněna bez chyb.                                                                                                                                          |
| Stav               | Cíle bylo dosaženo                                                                                                                                                                     |

(zdroj: vlastní)

Tabulka 11 Testování schopnosti smazání bloků

| Pole               | Popis                                                                                                                                                                                |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cíl                | Smazat blok z pracovního prostoru.                                                                                                                                                   |
| Předpoklady        | Stránka index.php je otevřená v prohlížeči.<br>Uživatel se přihlásil ke svému účtu.<br>Byl vybrán aktuální pracovní prostor.<br>Aktuální pracovní prostor obsahuje aspoň jeden blok. |
| Kroky              | 1. Kliknout na tlačítko nabídky bloků vpravo.<br>2. V rozbalovací nabídce vybrat možnost „Smazat“.<br>3. Potvrdit smazání bloku.                                                     |
| Očekávaný výsledek | Blok bude z pracovního prostoru odstraněn.                                                                                                                                           |
| Skutečný výsledek  | Blok byl z pracovního prostoru odstraněn.                                                                                                                                            |
| Stav               | Cíle bylo dosaženo                                                                                                                                                                   |

(zdroj: vlastní)

### 6.1.5 Testování interakcí mezi lidmi

Abych otestoval interakce mezi lidmi, musím otestovat schopnost pozvat lidi do svého pracovního prostoru a schopnost reagovat na pozvání. To zahrnuje i možnost změnit si profil.

Tabulka 12 Testování pozvání nového účastníka do pracovního prostoru

| Pole               | Popis                                                                                                                                                                                                                              |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cíl                | Přihlásit nového účastníka do pracovního prostoru.                                                                                                                                                                                 |
| Předpoklady        | Stránka index.php je otevřená v prohlížeči.<br>Uživatel se přihlásil ke svému účtu.<br>Byl vybrán aktuální pracovní prostor.                                                                                                       |
| Kroky              | 1. Kliknout na tlačítko napravo od tlačítka pracovního prostoru.<br>2. V rozbalovací nabídce vybrat možnost „Účastníci“<br>3. V otevřené nabídce zadat e-mail účastníka.<br>4. Kliknout na tlačítko „Pozvat“ pro odeslání pozvánky |
| Očekávaný výsledek | Účastníkovi bude zaslána pozvánka.                                                                                                                                                                                                 |
| Skutečný výsledek  | Objevila se zpráva „Pozvánka byla odeslána!“ a pozvánka byla odeslána.                                                                                                                                                             |
| Stav               | Cíle bylo dosaženo                                                                                                                                                                                                                 |

(zdroj: vlastní)

Tabulka 13 Testování pozvánky pro účastníka, který není v systému

| Pole               | Popis                                                                                                                                                                                                                                               |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cíl                | Zkusit pozvat účastníka, který není v systému.                                                                                                                                                                                                      |
| Předpoklady        | Existuje uživatel s e-mailem <u>test2@example.com</u><br>Stránka index.php je otevřená v prohlížeči.<br>Uživatel se přihlásil ke svému účtu.<br>Byl vybrán aktuální pracovní prostor.                                                               |
| Kroky              | 1. Kliknout na tlačítko napravo od tlačítka pracovního prostoru.<br>2. V rozbalovací nabídce vybrat možnost „Účastníci“<br>3. V otevřené nabídce zadat e-mail účastníka který neexistuje.<br>4. Kliknout na tlačítko „Pozvat“ pro odeslání pozvánky |
| Očekávaný výsledek | Objeví se zpráva, že takový účastník neexistuje.                                                                                                                                                                                                    |
| Skutečný výsledek  | Objevila se zpráva "Uživatel s tímto emailem neexistuje" a pozvánka nebyla odeslána.                                                                                                                                                                |
| Stav               | Cíle bylo dosaženo                                                                                                                                                                                                                                  |

(zdroj: vlastní)

Tabulka 14 Testování přijetí pozvánky od někoho

| Pole               | Popis                                                                                                                                                     |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cíl                | Přijmout pozvání do pracovního prostoru někoho jiného.                                                                                                    |
| Předpoklady        | Stránka index.php je otevřená v prohlížeči.<br>Uživatel se přihlásil ke svému účtu.<br>Je pozvánka od jiného uživatele                                    |
| Kroky              | 1. Kliknout na tlačítko vpravo nahoře.<br>2. V rozbalovací nabídce vybrat možnost „pozvánky“.<br>3. V zobrazené pozvánce kliknout na tlačítko „Přijmout“. |
| Očekávaný výsledek | Pozvánka bude přijata a pracovní prostor se zobrazí v seznamu.                                                                                            |
| Skutečný výsledek  | Pozvánka byla přijata, objevila se zpráva o přijetí pozvánky a pracovní prostor se objevil v seznamu.                                                     |
| Stav               | Cíle bylo dosaženo                                                                                                                                        |

(zdroj: vlastní)

Tabulka 15 Testování odmítnutí pozvánky od někoho

| Pole               | Popis                                                                                                                                                      |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cíl                | Odmítnout pozvání do pracovního prostoru někoho jiného.                                                                                                    |
| Předpoklady        | Stránka index.php je otevřená v prohlížeči.<br>Uživatel se přihlásil ke svému účtu.<br>Je pozvánka od jiného uživatele                                     |
| Kroky              | 1. Kliknout na tlačítko vpravo nahoře.<br>2. V rozbalovací nabídce vybrat možnost „pozvánky“.<br>3. V zobrazené pozvánce kliknout na tlačítko „Odmítnout“. |
| Očekávaný výsledek | Pozvánka bude odmítnuta.                                                                                                                                   |
| Skutečný výsledek  | Pozvánka byla odmítnuta, objevila se zpráva o odmítnutí pozvánky.                                                                                          |
| Stav               | Cíle bylo dosaženo                                                                                                                                         |

(zdroj: vlastní)

Tabulka 16 Testování změn dat profilů uživatele

| Pole               | Popis                                                                                                                                                                                                            |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cíl                | Upravit podrobnosti profilu.                                                                                                                                                                                     |
| Předpoklady        | Stránka index.php je otevřená v prohlížeči. Uživatel se přihlásil ke svému účtu.                                                                                                                                 |
| Kroky              | 1. Kliknout na tlačítko vpravo nahoře.<br>2. V rozbalovací nabídce vybrat možnost „Nastavení uživatele“.<br>3. V otevřené nabídce změnit uživatelské údaje: jméno, příjmení, datum narození, avatar, společnost. |
| Očekávaný výsledek | Uživatelská data budou úspěšně změněna.                                                                                                                                                                          |
| Skutečný výsledek  | Všechna uživatelská data byla úspěšně změněna.                                                                                                                                                                   |
| Stav               | Cíle bylo dosaženo                                                                                                                                                                                               |

(zdroj: vlastní)

#### 6.1.6 Testování rolí

Potřebuji otestovat, zda jsou funkce správně přístupné pro všechny role v jednom pracovním prostoru.

Tabulka 17 Testování role creator

| Pole               | Popis                                                                                                                                                                                                                      |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cíl                | Otestovat si roli creator.                                                                                                                                                                                                 |
| Předpoklady        | Stránka index.php je otevřená v prohlížeči. Uživatel se přihlásil ke svému účtu. Uživatel má v pracovním prostoru roli creator.                                                                                            |
| Kroky              | 1. Otestovat si schopnost spravovat pracovní prostor.<br>2. Otestovat si schopnost vytvářet nové bloky<br>3. Otestovat si schopnost pracovat s bloky.<br>4. Otestovat si schopnost spravovat účastníky pracovního prostoru |
| Očekávaný výsledek | Účastník může plně spravovat pracovní prostor a účastníky, vytvářet nové bloky a měnit data v blocích.                                                                                                                     |
| Skutečný výsledek  | Všechny funkce pro roli creator jsou k dispozici.                                                                                                                                                                          |
| Stav               | Cíle bylo dosaženo                                                                                                                                                                                                         |

(zdroj: vlastní)

Tabulka 18 Testování role admin

| Pole               | Popis                                                                                                                                                                                                                      |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cíl                | Otestovat si roli admin.                                                                                                                                                                                                   |
| Předpoklady        | Stránka index.php je otevřená v prohlížeči.<br>Uživatel se přihlásil ke svému účtu.<br>Uživatel má v pracovním prostoru roli admin.                                                                                        |
| Kroky              | 1. Otestovat si schopnost spravovat pracovní prostor.<br>2. Otestovat si schopnost vytvářet nové bloky<br>3. Otestovat si schopnost pracovat s bloky.<br>4. Otestovat si schopnost spravovat účastníky pracovního prostoru |
| Očekávaný výsledek | Účastník může změnit název pracovního prostoru, může pracovní prostor opustit, změnit role účastníků, vytvářet nové bloky a měnit data v blocích.                                                                          |
| Skutečný výsledek  | Všechny funkce pro roli admin jsou k dispozici.                                                                                                                                                                            |
| Stav               | Cíle bylo dosaženo                                                                                                                                                                                                         |

(zdroj: vlastní)

Tabulka 19 Testování role worker

| Pole               | Popis                                                                                                                                                                                                                      |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cíl                | Otestovat si roli worker.                                                                                                                                                                                                  |
| Předpoklady        | Stránka index.php je otevřená v prohlížeči.<br>Uživatel se přihlásil ke svému účtu.<br>Uživatel má v pracovním prostoru roli worker.                                                                                       |
| Kroky              | 1. Otestovat si schopnost spravovat pracovní prostor.<br>2. Otestovat si schopnost vytvářet nové bloky<br>3. Otestovat si schopnost pracovat s bloky.<br>4. Otestovat si schopnost spravovat účastníky pracovního prostoru |
| Očekávaný výsledek | Účastník nemůže spravovat pracovní prostor a účastníky, může pracovní prostor opustit, nemůže vytvářet nové bloky ale může měnit data v blocích.                                                                           |
| Skutečný výsledek  | Všechny funkce pro roli worker jsou k dispozici.                                                                                                                                                                           |
| Stav               | Cíle bylo dosaženo                                                                                                                                                                                                         |

(zdroj: vlastní)

Tabulka 20 Testování role viewer

| Pole               | Popis                                                                                                                                                                                                                      |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cíl                | Otestovat si roli viewer.                                                                                                                                                                                                  |
| Předpoklady        | Stránka index.php je otevřená v prohlížeči.<br>Uživatel se přihlásil ke svému účtu.<br>Uživatel má v pracovním prostoru roli viewer.                                                                                       |
| Kroky              | 1. Otestovat si schopnost spravovat pracovní prostor.<br>2. Otestovat si schopnost vytvářet nové bloky<br>3. Otestovat si schopnost pracovat s bloky.<br>4. Otestovat si schopnost spravovat účastníky pracovního prostoru |
| Očekávaný výsledek | Účastník si může pouze prohlížet data v blocích pracovního prostoru a může pracovní prostor opustit.                                                                                                                       |
| Skutečný výsledek  | Všechny funkce pro roli viewer jsou k dispozici.                                                                                                                                                                           |
| Stav               | Cíle bylo dosaženo                                                                                                                                                                                                         |

(zdroj: vlastní)

### 6.1.7 Výsledky funkčního testování

Během funkčního testování byly ověřeny všechny hlavní scénáře interakce s uživatelem: vytvoření účtu, přihlášení do systému, správa pracovního prostoru (přidávání, výběr, mazání), operace s různými bloky, odesílání a zpracování pozvánek, změna rolí účastníků a aktualizace profilu. Pro každý scénář byly zaznamenány očekávané a skutečné výsledky. Nebyly zjištěny žádné nesrovnalosti, což nám umožňuje dojít k závěru, že implementovaná funkcionální plně splňuje stanovené požadavky a funkce plánované ve fázi návrhu.

## 6.2 Bezpečnostní testování

Abych otestoval bezpečnost aplikace, musím se pokusit reprodukovat scénáře hrozeb pro mou aplikaci a podívat se na výsledky, které z těchto testů vyplynou.

### 6.2.1 Testování ochrany proti SQL-injekcím

Abych otestoval obranu proti SQL-injekcím, zkusím si sám píchnout injekci.

Tabulka 21 Testování ochrany proti SQL-injekcím

| Pole               | Popis                                                                                                   |
|--------------------|---------------------------------------------------------------------------------------------------------|
| Cíl                | Ověřit, že vstupy ošetřují SQL injekci a dotazy používají připravené výrazy.                            |
| Předpoklady        | Stránka index.php je otevřená v prohlížeči. Uživatel není přihlášen.                                    |
| Kroky              | 1. Do pole E-mail vložit ' OR 1=1 --.<br>2. Do pole Heslo vložit libovolný text<br>3. Odeslat formulář. |
| Očekávaný výsledek | Přihlášení nebude úspěšné, zobrazí se obecná chyba „Neplatné přihlašovací údaje“.                       |
| Skutečný výsledek  | Prohlížeč odmítl vstup – hláška o neplatné adrese; žádný dotaz k DB odeslán.                            |
| Stav               | Cíle bylo dosaženo                                                                                      |

(zdroj: vlastní)

### 6.2.2 Testování bezpečného uložení hesel

Pro otestování bezpečnosti ukládání hesel je nutné ujasnit si, že heslo registrovaného uživatele je v databázi uloženo jako hash.

Tabulka 22 Testování bezpečného uložení hesel

| Pole               | Popis                                                                                |
|--------------------|--------------------------------------------------------------------------------------|
| Cíl                | Ověřit, že hesla jsou ukládána hashovaná a nikoli v prostém textu.                   |
| Předpoklady        | Přístup k DB (tabulka users).<br>Uživatel „test@example.com“ je právě zaregistrován. |
| Kroky              | 1. Otevřít tabulku users v phpMyAdmin.<br>2. Zkontrolovat sloupec password_hash.     |
| Očekávaný výsledek | Hodnota začíná prefixem \$2y\$. Neobsahuje čitelný řetězec hesla.                    |
| Skutečný výsledek  | Heslo je uloženo jako řetězec \$2y\$10\$..., délka 60 znaků.                         |
| Stav               | Cíle bylo dosaženo                                                                   |

(zdroj: vlastní)

### 6.2.3 Testování skrytí interních chyb

Pro testování skrytí interních chyb je nutné simulovat situaci chyby připojení a podívat se na popis chyby připojení.

Tabulka 23 Testování skrytí interních chyb

| Pole               | Popis                                                                                              |
|--------------------|----------------------------------------------------------------------------------------------------|
| Cíl                | Zajistit, že interní výjimky / chyby DB se nezobrazují koncovému uživateli.                        |
| Předpoklady        | Dočasně přejmenován soubor db.php na db2.php.                                                      |
| Kroky              | 1. Otevřít login.php v prohlížeči.                                                                 |
| Očekávaný výsledek | Stránka zobrazí generickou hlášku „Server error“, bez stack-trace a detailů o souborových cestách. |
| Skutečný výsledek  | Zobrazila se stránka 500 s obecným textem, interní data nejsou odhalena.                           |
| Stav               | Cíle bylo dosaženo                                                                                 |

(zdroj: vlastní)

#### 6.2.4 Testování validace uploadu avatara

Abych otestoval validaci nahrání avatara, musím zkusit nahrát soubor avatar.php se spustitelným kódem místo avatara.

Tabulka 24 Testování validace nahrání souboru

| Pole               | Popis                                                                                                                         |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------|
| Cíl                | Provéřit, zda aplikace odmítá nebezpečné soubory při uploadu.                                                                 |
| Předpoklady        | Dočasně přejmenován soubor db.php na db2.php.                                                                                 |
| Kroky              | 1. Připravit soubor test.php s obsahem <code>&lt;?php echo 123; ?&gt;</code> .<br>2. Vybrat tento soubor a potvrdit nahrání.. |
| Očekávaný výsledek | Aplikace odmítne soubor s hláškou „Pouze PNG nebo JPG obrázek“. Soubor se neuloží.                                            |
| Skutečný výsledek  | Upload byl zamítnut, na serveru se soubor neobjevil.                                                                          |
| Stav               | Cíle bylo dosaženo                                                                                                            |

(zdroj: vlastní)

#### 6.2.5 Testování ochrany proti CSRF

Abych otestoval ochranu CSRF, musím zachytit jakýkoli platný POST požadavek z aplikace, odstranit nebo nahradit v něm CSRF token a požadavek znovu odeslat - server je povinen jej odmítnout.

Tabulka 25 Testování ochrany proti CSRF

| Pole               | Popis                                                                                                                                                         |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cíl                | Ověřit, že POST požadavky vyžadují platný CSRF token.                                                                                                         |
| Předpoklady        | Uživatel je přihlášen a má otevřený prohlížeč.                                                                                                                |
| Kroky              | 1. Vytvořit externí HTML formulář, který posílá POST na index.php s pole add_workspace=1, ale bez tokenu.<br>2. Odeslat formulář v kontextu přihlášené relace |
| Očekávaný výsledek | Server vrátí JSON { "success": false, "error": "Invalid token" }, workspace se nevytvoří.                                                                     |
| Skutečný výsledek  | Server odpověděl Invalid token, žádný záznam v DB.                                                                                                            |
| Stav               | Cíle bylo dosaženo                                                                                                                                            |

(zdroj: vlastní)

### 6.2.6 Testování odolnosti proti XSS

Abych otestoval ochranu proti XSS, vložím do vstupního pole určeného pro uživatelský text řetězec obsahující JavaScriptový kód. Aplikace musí tento řetězec při zobrazení správně escapovat a zobrazit jej pouze jako obyčejný text, aniž by se kód spustil.

Tabulka 26 Testování ochrany proti XSS

| Pole               | Popis                                                                                                                               |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| Cíl                | Ověřit, že uživatelský vstup je escapován a nelze spustit JS kód.                                                                   |
| Předpoklady        | Uživatel je přihlášen a má právo přidat poznámku.                                                                                   |
| Kroky              | 1. Do obsahu poznámky vložit <code>&lt;script&gt;alert("xss")&lt;/script&gt;</code> .<br>2. Uložit poznámku a znovu načíst stránku. |
| Očekávaný výsledek | Text poznámky se zobrazí doslova, žádné okno alert se neotevře.                                                                     |
| Skutečný výsledek  | Skript nebyl vykonán, text byl escapován.                                                                                           |
| Stav               | Cíle bylo dosaženo                                                                                                                  |

(zdroj: vlastní)

### 6.2.7 Výsledky bezpečnostního testování

Během bezpečnostních testů byly ověřeny všechny klíčové hrozby (SQL injections, CSRF, XSS, nahrávání škodlivých souborů a odhalování systémových chyb). Ve všech případech aplikace útoky správně odmítla nebo data vygenerovala v bezpečné, chráněné podobě. Nebyly zjištěny žádné kritické zranitelnosti, což potvrzuje, že implementované ochranné mechanismy splňují doporučení OWASP a umožňuje nám považovat systém za připravený k nasazení v produkčním prostředí.

## ZÁVĚR

V rámci bakalářské práce byla vyvinuta webová aplikace pro řízení týmových projektů, která implementuje nejoblíbenější funkce nezbytné pro efektivní týmovou práci. Byla provedena analýza stávajících řešení na trhu (Trello, Asana, Jira a další), která nám umožnila určit silné a slabé stránky konkurenčních aplikací a zohlednit je při návrhu vlastního systému.

Vývoj zahrnoval kompletní cyklus tvorby softwaru: od analýzy požadavků a návrhu architektury až po implementaci uživatelského rozhraní, nastavení databáze a informační bezpečnost. Jako hlavní technologie byly použity HTML, CSS, PHP, JavaScript a MySQL.

Praktická část projektu demonstruje vytvoření funkčního rozhraní zahrnujícího registraci a autorizaci uživatelů, správu pracovního prostoru, sledování úkolů, interakci účastníků a vizualizaci dat. Pozornost byla věnována i ochraně před běžnými zranitelnostmi.

Funkční a bezpečnostní testování potvrdilo správný chod všech komponent aplikace a splnění stanovených požadavků. Výsledkem provedené práce byl vytvoření prototypu webové aplikace, připravené k dalšímu rozvoji a rozšíření funkcionality.

## SEZNAM POUŽITÉ LITERATURY

- [1] LER Studio. *Co je webová aplikace a k čemu slouží*. Online. Lerstudio.cz, 2025. Dostupné z: <https://lerstudio.cz/blog/co-je-webova-aplikace-a-k-cemu-slouzi> [cit. 2025-03-09]
- [2] INTELIVITA. *Types of Web Applications*. Online. Intelivita.com, 2025. Dostupné z: <https://www.intelivita.com/blog/types-of-web-applications/> [cit. 2025-03-09]
- [3] BRITANNICA. *Client-server architecture*. Online. Britannica.com, 2025 [cit. 2025-03-10]. Dostupné z: <https://www.britannica.com/technology/client-server-architecture>
- [4] IBM CORPORATION. *What Is Three-Tier Architecture?* Online. IBM Think Topics, 2025. Dostupné z: <https://www.ibm.com/think/topics/three-tier-architecture> [cit. 2025-03-24]
- [5] FOWLER, Martin. *Patterns of Enterprise Application Architecture*. Boston: Addison-Wesley, 2002. ISBN 0-321-12742-0.
- [6] Sheldon, Robert. *What is model-view-controller (MVC)?* Online. TechTarget, 2023. Dostupné z: <https://www.techtarget.com/whatis/definition/model-view-controller-MVC> [cit. 2025-03-24]
- [7] Smith, Josh. Patterns – WPF Apps With The Model-View-ViewModel Design Pattern. *MSDN Magazine*, únor 2009, sv. 24, č. 02, Online. Dostupné z: <https://learn.microsoft.com/en-us/archive/msdn-magazine/2009/february/patterns-wpf-apps-with-the-model-view-viewmodel-design-pattern> [cit. 2025-03-25].
- [8] MOZILLA CORPORATION'S. *http request methods*. Online. Mozilla.org, 2025. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Methods> [cit. 2025-03-27].
- [9] FIELDING, Roy T. *Principled Design of the Modern Web Architecture*. UCI.edu, Online. 2002 Dostupné z: <https://www.ics.uci.edu/~taylor/documents/2002-REST-TOIT.pdf> [cit. 2025-03-27].
- [10] Atlassian. „What Is Trello: Learn Features, Uses & More.“ *Trello*, Online. Dostupné z: <https://trello.com/tour> [cit. 2025-04-05].

- [11] Asana inc., Understand Asana's core features. Online. Asana.com, 2025. Dostupně z: <https://asana.com/ru/features> [cit. 2025-04-05].
- [12] MICROSOFT. Začínáme s aplikací Microsoft Teams. Online. Microsoft.com, 2025. Dostupně z: <https://support.microsoft.com/cs-cz/office/začínáme-s-aplikací-microsoft-teams-b98d533f-118e-4bae-bf44-3df2470c2b12> [cit. 2025-04-05].
- [13] JIRA. Great outcomes start with Jira. Online. atlassian.com, 2025. Dostupně z: <https://www.atlassian.com/software/jira> [cit. 2025-04-09].
- [14] MONDAY. Ваша лучшая рабочая платформа. Online. Monday.com, 2025. Dostupně z: <https://monday.com/lang/ru> [cit. 2025-04-09].
- [15] SKILLFAKTORY MEDIA. Язык разметки. Online. Skillfactory.ru, 2024. Dostupně z: <https://blog.skillfactory.ru/glossary/yazyk-razmetki/> [cit. 2025-04-09].
- [16] LUBBERS, Peter; ALBERS, Brian; SALIM, Frank. 2011. *HTML5: programujeme moderní webové aplikace*. Brno: Computer Press. ISBN 978-80-251-3539-6.
- [17] MOZILLA CORPORATION'S. *CSS: Cascading Style Sheets*. Online. Mozilla.org, 2025. Dostupně z: <https://developer.mozilla.org/en-US/docs/Web/CSS> [cit. 2025-04-10].
- [18] DUCKETT, Jon. *HTML and CSS: Design and Build Websites*. Hoboken: John Wiley & Sons, 2011. ISBN 978-1-118-00818-8.
- [19] PHP GROUP. *PHP Manual – Úvod* Online. PHP.net, 2025. Dostupné z: <https://www.php.net/> [cit.2025-04-11]
- [20] TurnKey Linux. „LAMP Stack Appliance.“ *TurnKey Linux Library*, Online. Dostupné z: <https://www.turnkeylinux.org/lamp> [2025-04-11].
- [21] WP Web Infotech. *How PHP Still Powers the Web in 2025: Stats and Use Cases*. Medium, Online. Dostupné z: <https://medium.com/@wpwebinfotech/how-php-still-powers-the-web-in-2025-stats-and-use-cases-bbf135b37118> [2025-04-12].
- [22] HARTINGER, David. Lekce 1 - Úvod do JavaScriptu. Online. Itnetwork.cz, 2024. Dostupně z: <https://www.itnetwork.cz/javascript/zaklady/javascript-tutorial-uvod-do-javascriptu-nepochopeny-jazyk> [cit. 2025-04-13]
- [23] MUNZNER, Tamara. *Visualization Analysis and Design*. 2nd ed. Boca Raton: CRC Press, 2014. ISBN 978-1-4987-5971-7

- [24] Q-Success. „Historical trends in the usage statistics of client-side programming languages for websites.“ *W3Techs.com*, Online. Dostupné z: [https://w3techs.com.translate.google/technologies/history\\_overview/client\\_side\\_language/all?\\_x\\_tr\\_sl=en&\\_x\\_tr\\_tl=ru&\\_x\\_tr\\_hl=ru&\\_x\\_tr\\_pto=rq](https://w3techs.com.translate.google/technologies/history_overview/client_side_language/all?_x_tr_sl=en&_x_tr_tl=ru&_x_tr_hl=ru&_x_tr_pto=rq) [2025-04-16].
- [25] NARAMORE, Elizabeth. *Vytváříme webové aplikace v PHP5, MySQL a Apache*. Praha: Grada Publishing, 2009. ISBN 978-80-251-1073-7
- [26] Renaps. „Oracle MySQL Managed Services.“ *Renaps*, Online. Dostupné z: <https://renaps.com/en/services/oracle-mysql-Managed-Services> [2025-04-17].
- [27] APACHE SOFTWARE FOUNDATION. The Apache HTTP Server Project. Online. Apache.org, 2025. Dostupné z: <https://httpd.apache.org/> [cit. 2025-04-20].
- [28] GEEKSFORGEEKS. What is Apache Server? Online. GeeksforGeeks.org, 2024. Dostupné z: <https://www.geeksforgeeks.org/what-is-apache-server/> [cit. 2025-04-20].
- [29] PHPMYADMIN DEVEL TEAM. phpMyAdmin – Official Homepage. Online. phpMyAdmin.net, 2025. Dostupné z: <https://www.phpmyadmin.net> [cit. 2025-04-20].
- [30] OPENAI. Online. Dostupné z: <https://chatgpt.com/>. [cit. 2025-04-27].

**SEZNAM OBRÁZKŮ**

|                                                       |    |
|-------------------------------------------------------|----|
| Obrázek 1 Jednoduchá klient-server aplikace .....     | 15 |
| Obrázek 2 Schéma MVC .....                            | 16 |
| Obrázek 3 Domovská stránka Trello .....               | 19 |
| Obrázek 4 Domovská stránka Asana .....                | 20 |
| Obrázek 5 Domovská stránka Microsoft Teams.....       | 21 |
| Obrázek 6 Domovská stránka Jira .....                 | 22 |
| Obrázek 7 Domovská stránka Monday.....                | 23 |
| Obrázek 8 Návrh registračního formuláře .....         | 33 |
| Obrázek 9 Návrh přihlašovacího formuláře.....         | 33 |
| Obrázek 10 Návrh pracovní stránky .....               | 34 |
| Obrázek 11 Návrh bloku „Seznam úkolů“ .....           | 35 |
| Obrázek 12 Návrh bloku „Graf“ .....                   | 36 |
| Obrázek 13 Návrh bloku „Koláčový graf“ .....          | 37 |
| Obrázek 14 Návrh bloku „Kalendář“ .....               | 38 |
| Obrázek 15 Návrh bloku „Poznámky“ .....               | 39 |
| Obrázek 16 Návrh bloku „Sledovač času“ .....          | 40 |
| Obrázek 17 Návrh bloku „Komentáře“ .....              | 40 |
| Obrázek 18 Návrh menu „Účastníci“ .....               | 41 |
| Obrázek 19 Návrh menu „Nastavení uživatele“ .....     | 42 |
| Obrázek 20 Návrh menu pozvánek.....                   | 42 |
| Obrázek 21 Datová struktura projektu.....             | 43 |
| Obrázek 22 Autentizační scénář .....                  | 47 |
| Obrázek 23 Scénář pro práci na pracovní stránce ..... | 48 |
| Obrázek 24 Scénář pro práci s boky .....              | 49 |
| Obrázek 25 Scénář pro interakci s uživateli .....     | 50 |
| Obrázek 26 Struktura projektu.....                    | 51 |
| Obrázek 27 Registrační formulář.....                  | 55 |
| Obrázek 28 Přihlašovací formulář .....                | 56 |
| Obrázek 29 Pracovní stránka .....                     | 57 |
| Obrázek 30 Blok „Seznam úkolů“ .....                  | 58 |
| Obrázek 31 Blok „Graf“ .....                          | 59 |
| Obrázek 32 Blok „Koláčový graf“ .....                 | 60 |
| Obrázek 33 Blok „Kalendář“ .....                      | 61 |
| Obrázek 34 Blok „Kalendář“ .....                      | 61 |

|                                             |    |
|---------------------------------------------|----|
| Obrázek 35 Blok „Poznámky“ .....            | 62 |
| Obrázek 36 Blok „Sledovač času“ .....       | 63 |
| Obrázek 37 Blok „Komentáře“ .....           | 64 |
| Obrázek 38 Okno „Účastníci“ .....           | 64 |
| Obrázek 39 Okno „Nastavení uživatele“ ..... | 65 |
| Obrázek 40 Návrh menu pozvánek .....        | 66 |

**SEZNAM TABULEK**

|                                                                           |    |
|---------------------------------------------------------------------------|----|
| Tabulka 1 Testování úspěšné registrace .....                              | 68 |
| Tabulka 2 Testování neúspěšné registrace.....                             | 69 |
| Tabulka 3 Testování úspěšné autorizace .....                              | 69 |
| Tabulka 4 Testování neúspěšné autorizace.....                             | 70 |
| Tabulka 5 Testování vytvoření pracovního prostoru.....                    | 70 |
| Tabulka 6 Testování přejmenování pracovního prostoru .....                | 71 |
| Tabulka 7 Testování přepínání mezi pracovními prostory .....              | 71 |
| Tabulka 8 Testování smazání pracovního prostoru .....                     | 72 |
| Tabulka 9 Testování možnosti přidávání bloků.....                         | 72 |
| Tabulka 10 Testování schopnosti správně pracovat s bloky.....             | 73 |
| Tabulka 11 Testování schopnosti smazání bloků .....                       | 73 |
| Tabulka 12 Testování pozvání nového účastníka do pracovního prostoru..... | 74 |
| Tabulka 13 Testování pozvánky pro účastníka, který není v systému .....   | 74 |
| Tabulka 14 Testování přijetí pozvánky od někoho.....                      | 75 |
| Tabulka 15 Testování odmítnutí pozvánky od někoho.....                    | 75 |
| Tabulka 16 Testování změn dat profilů uživatele.....                      | 76 |
| Tabulka 17 Testování role creator .....                                   | 76 |
| Tabulka 18 Testování role admin .....                                     | 77 |
| Tabulka 19 Testování role worker .....                                    | 77 |
| Tabulka 20 Testování role viewer .....                                    | 78 |
| Tabulka 21 Testování ochrany proti SQL-injekcím .....                     | 79 |
| Tabulka 22 Testování bezpečného uložení hesel.....                        | 79 |
| Tabulka 23 Testování skrytí interních chyb .....                          | 80 |
| Tabulka 24 Testování validace nahrání souboru .....                       | 80 |
| Tabulka 25 Testování ochrany proti CSRF .....                             | 81 |
| Tabulka 26 Testování ochrany proti XSS.....                               | 81 |

## SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

|        |                                           |
|--------|-------------------------------------------|
| API    | Application Programming Interface         |
| ASP    | Active Server Page                        |
| COBRA  | Common Object Request Broker Architecture |
| CSV    | Comma-Separated Values                    |
| CSS    | Cascading Style Sheets                    |
| CSRF   | Cross-Site Request Forgery                |
| DBMS   | DataBase Management System                |
| HTML   | HyperText Markup Language                 |
| IBM    | International Business Machines           |
| JIT    | Just-It-Time                              |
| JSON   | JavaScript Object Notation                |
| LAMP   | Linux, Apache, MySQL, PHP                 |
| MathML | Mathematical Markup Language              |
| MPA    | Multi-Page Application                    |
| MVC    | Model-View-Controller                     |
| MVVM   | Model-View-ViewModel                      |
| SQL    | Structured Query Language                 |
| NoSQL  | Not Only SQL                              |
| OWASP  | Open Web Application Security Project     |
| PDF    | Portable Document Format                  |
| PHP    | PHP: Hypertext Preprocessor               |
| QBE    | Query By Example                          |
| REST   | Representational State Transfer           |
| SOAP   | Simple Object Access Protocol             |
| SPA    | Single-Page Application                   |

|       |                                          |
|-------|------------------------------------------|
| SVG   | Scalable Vector Graphics                 |
| W3C   | World Wide Web Consortium                |
| WPF   | Windows Presentation Foundation          |
| XAMPP | Cross-Platform, Apache, MySQL, PHP, Perl |
| XML   | eXtensible Markup Language               |
| XSS   | Cross-Site Scripting                     |

**SEZNAM PŘÍLOH**

Příloha P I: příloha.zip

Příloha P II: SQL dotaz pro databázi

## **PŘÍLOHA P I: PŘÍLOHA.ZIP**

Příloha obsahuje zip archiv se zdrojovými kódy pro aplikace.

Archiv obsahuje složku DiplomApp která obsahuje soubory login.php, login\_styles.css, index.php, styles.css, logout.php, user\_actions.php a db.php, pro aplikaci a složku avatars pro načítání avatarů se souborem .htaccess.

## PŘÍLOHA P II: SQL DOTAZ PRO DATABÁZI

```
CREATE DATABASE IF NOT EXISTS diplomapp;
```

```
USE diplomapp;
```

```
CREATE TABLE users (
```

```
    id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    email VARCHAR(255) NOT NULL UNIQUE,
```

```
    password_hash VARCHAR(255) NOT NULL,
```

```
    first_name VARCHAR(100) NOT NULL,
```

```
    last_name VARCHAR(100) NOT NULL,
```

```
    birthdate DATE,
```

```
    organization VARCHAR(255),
```

```
    avatar_url TEXT,
```

```
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
```

```
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE  
    CURRENT_TIMESTAMP
```

```
);
```

```
CREATE TABLE workspaces (
```

```
    id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    name VARCHAR(255) NOT NULL,
```

```
    creator_id INT NOT NULL,
```

```
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
```

```
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE  
    CURRENT_TIMESTAMP,
```

```
    FOREIGN KEY (creator_id) REFERENCES users(id)
```

```
);
```

```
CREATE TABLE workspace_participants (
```

```
    id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    workspace_id INT NOT NULL,
```

```
    user_id INT NOT NULL,
```

```
    role ENUM('creator', 'admin', 'worker', 'viewer') NOT NULL,
```

```
    joined_at DATETIME DEFAULT CURRENT_TIMESTAMP,
```

```
    FOREIGN KEY (workspace_id) REFERENCES workspaces(id),
```

```
    FOREIGN KEY (user_id) REFERENCES users(id)
```

```
);
```

```
CREATE TABLE workspace_blocks (
```

```
    id INT AUTO_INCREMENT PRIMARY KEY,
```

```

workspace_id INT NOT NULL,
title VARCHAR(255),
content TEXT,
created_by INT NOT NULL,
created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
`row` INT,
`col` INT,
width INT,
height INT,
FOREIGN KEY (workspace_id) REFERENCES workspaces(id),
FOREIGN KEY (created_by) REFERENCES users(id)
);

```

```

CREATE TABLE workspace_invitations (
id INT AUTO_INCREMENT PRIMARY KEY,
workspace_id INT NOT NULL,
invited_user_email VARCHAR(255) NOT NULL,
invited_by INT NOT NULL,
role ENUM('admin', 'worker', 'viewer') NOT NULL,
token VARCHAR(255) NOT NULL,
status ENUM('pending', 'accepted', 'declined') NOT NULL DEFAULT
'pending',
created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
responded_at DATETIME,
FOREIGN KEY (workspace_id) REFERENCES workspaces(id),
FOREIGN KEY (invited_by) REFERENCES users(id)
);

```