

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Webová aplikace pro pokročilé sdílení souborů v
univerzitním prostředí



2023

Vedoucí práce:
Mgr. Markéta Trnečková, Ph.D.

Karolína Vorlíčková

Studijní program: Informační technologie,
kombinovaná forma

Bibliografické údaje

Autor: Karolína Vorlíčková
Název práce: Webová aplikace pro pokročilé sdílení souborů v univerzitním prostředí
Typ práce: bakalářská práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby: 2023
Studijní program: Informační technologie, kombinovaná forma
Vedoucí práce: Mgr. Markéta Trnečková, Ph.D.
Počet stran: 62
Přílohy: elektronická data v úložišti katedry informatiky
Jazyk práce: český

Bibliographic info

Author: Karolína Vorlíčková
Title: Web application for advanced file sharing in a university environment
Thesis type: bachelor thesis
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense: 2023
Study program: Information Technologies, combined form
Supervisor: Mgr. Markéta Trnečková, Ph.D.
Page count: 62
Supplements: electronic data in the storage of department of computer science
Thesis language: Czech

Anotace

Tato bakalářská práce se zaměřuje na vytvoření webové aplikace pro sdílení učebních materiálů v prostředí univerzity. V rámci implementace práce byl kladen důraz na spolehlivé ověření identity uživatele, mechanismus kontroly přístupu a rolí uživatelů, notifikace (emailová upozornění a push notifikace), možnosti přidávání komentářů k souborům a samotné sdílení souborů (včetně nahrávání, stahování, prohlížení a mazání). Pro implementaci aplikace byly využity aktuálně moderní webové technologie, konkrétně framework React pro frontendovou část a .NET Core framework pro backendovou část. Tato kombinace technologií umožní vytvoření uživatelsky přívětivého a rychlého rozhraní a efektivního zpracování dat. Výsledkem této práce je funkční moderní webová aplikace, která poskytuje uživatelům univerzitního prostředí efektivní prostředek pro sdílení učebních materiálů.

Synopsis

This bachelor's thesis focuses on the development of a web application for sharing educational materials within a university environment. During the implementation, emphasis was placed on reliable user identity verification, access control mechanisms and user roles, notifications (email alerts and push notifications), the ability to add comments to files, and the actual file sharing functionality (including uploading, downloading, viewing, and deleting). The application was implemented using state-of-the-art web technologies, specifically the React framework for the frontend and .NET Core framework for the backend. This technology combination enables the creation of a user-friendly and responsive interface, as well as efficient data processing. The outcome of this thesis is a functional and modern web application that provides users within a university environment with an effective means of sharing educational materials.

Klíčová slova: Webová aplikace; Sdílení souborů; Datové úložiště; Cloudové úložiště; Přístupová práva; Uživatelské rozhraní; Multiplatformní aplikace; Kontrola přístupu; Správa uživatelů; Synchronizace souborů

Keywords: Web application; File sharing; Data storage; Cloud storage; Access rights; User interface; Cross-platform applications; Access control; User management; File synchronization

Nejprve bych chtěla vyjádřit svou vděčnost vedoucí práce, paní Mgr. Markétě Trnečkové Ph.D., za její odborné vedení a věcnou zpětnou vazbu. Dále bych ráda poděkovala své rodině, přátelům a kolegům za jejich neustálou podporu, povzbuzování a pochopení během mého studia.

Odevzdáním tohoto textu jeho autor/ka místopřísežně prohlašuje, že celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

Obsah

1	Úvod	9
2	Motivace	10
3	Analýza současného stavu	11
3.1	Platformy pro sdílení souborů	11
3.1.1	OwnCloud / Nextcloud	11
3.1.2	Pydio	12
3.2	Neoficiální univerzitní portály	14
3.2.1	Fektuška	14
3.2.2	Fituška	15
3.2.3	Primát.cz	15
3.3	Vzdělávací platforma Moodle	16
3.4	Sociální sítě	17
3.4.1	Facebookové skupiny	17
3.4.2	Discord	18
3.4.3	Microsoft Teams	18
3.5	Aktuální trendy webových technologií	19
3.5.1	Serverless Architektura	19
3.5.2	Rozvoj Moderního Front-Endu	20
3.5.3	Single Page Aplikace (SPA)	21
3.5.4	Progresivní Webové Aplikace (PWA)	21
3.6	Autentizace uživatelů	21
3.6.1	Zastaralé možnosti autentizace	22
3.6.2	Moderní způsoby autentizace	23
3.6.2.1	Microsoft Azure Active Directory	23
3.7	Shrnutí	24
4	Vývoj vlastní aplikace	25
4.1	Analýza Požadavků	25
4.2	Návrh aplikace	26
4.2.1	Návrh architektury	26
4.2.2	Návrh uživatelského rozhraní	26
4.2.3	Návrh databázové struktury	27
4.3	Použité technologie	28
4.3.1	Front-End	28
4.3.1.1	React	28
4.3.1.2	React router	29
4.3.1.3	Redux	29
4.3.1.4	SASS	30
4.3.1.5	TypeScript	31
4.3.2	Back-End	31
4.3.2.1	.NET Core 7	32

4.3.2.2	Odesílání emailů pomocí Outlook.com a Smtpc- lient	33
4.3.3	Databázová vrstva	34
4.3.3.1	Entity Framework Core	34
4.3.3.2	SQLite	35
4.3.4	Integrace Azure Active Directory	35
4.3.4.1	UPOL Tenant	36
4.4	Vývojové prostředí	36
4.4.1	JetBrains Rider	37
4.5	Produkční prostředí	37
4.5.1	Možnosti produkčního prostředí	39
4.5.2	Microsoft Azure App Service	39
4.6	Funkce aplikace	40
4.6.1	Procházení souborů	40
4.6.2	Komentování	40
4.6.3	Reakce uživatelů	41
4.6.4	Notifikace	42
4.6.5	Uživatelské role	43
5	Budoucí rozšíření	44
6	Závěr	45
A	Uživatelská příručka	47
A.1	Přihlašovací obrazovka	47
A.2	Rozložení stránek	47
A.3	Nástěnka	48
A.4	Detail kurzu	48
A.5	Detail souboru	50
A.6	Sekce komentářů	50
A.7	Novinky	50
A.8	Profil	51
A.9	Detail uživatele	51
B	Implementační manuál	53
B.1	Nasazení do Azure Web Cloud	53
B.2	Implementace na OS Windows	54
B.3	Implementace na MacOS	55
C	Obsah odevzdané přílohy	57
	Seznam zkratk	58
	Literatura	59

Seznam obrázků

1	Ukázka uživatelského rozhraní ownCloud/NextCloud	12
2	Ukázka uživatelského rozhraní Pydio [14]	13
3	Ukázka uživatelského rozhraní Fektušky	14
4	Ukázka uživatelského rozhraní Primat.cz včetně zobrazené reklamy	15
5	Ukázka uživatelského rozhraní Moodle	16
6	Serverless architektura Microsoft Azure [33]	19
7	Wireframe uživatelského rozhraní	27
8	Implementovaná autentizace pomocí Microsoft Azure	36
9	Nastavení tenantů v Azure AD	38
10	Vývojové prostředí JetBrains Rider	38
11	Procházení souborů v kurzu	40
12	Komentáře	41
13	Reakce uživatelů	42
14	Notifikace	42
15	Uživatelské role	43
16	Přihlašovací obrazovka	47
17	Modální okno	48
18	Nástěnka	49
19	Detail kurzu	49
20	Detail souboru	50
21	Novinky	51
22	Profil	52
23	Nastavení deploymentu v Azure AD	53

Seznam zdrojových kódů

1	Ukázka kódu v Reactu	28
2	Ukázka definování cest v React Routeru	29
3	Ukázka použití Reduxu	30
4	Ukázka stylování v SASS	30
5	Ukázka definování typů v TypeScriptu	31
6	Endpoint pro vytvoření kurzu	32
7	Příklad konfigurace a odesílání emailů	33
8	Objektový model kurzu	34
9	Nastavení Azure AD v souboru appsettings.json	37

1 Úvod

V dnešním rychle se rozvíjejícím světě moderních technologií a digitálního prostředí se vzdělávání stává stále více propojeným s digitálními možnostmi. Internet se etabloval jako primární zdroj informací a studijních materiálů, což otevírá dveře ke kontinuálnímu učení kdykoli a kdekoli. Tento dynamický trend směřuje k rostoucí potřebě kvalitních a moderních webových platform, které umožní studentům pohodlnější a efektivnější způsob sdílení vzdělávacích materiálů.

Cílem této bakalářské práce je vytvořit výkonnou webovou platformu, jež odpovídá nárokům dnešního stavu vědy a techniky. Tato inovativní platforma bude měnit pohled na způsob, jakým se studijní materiály sdílí, umožňujíc uživatelům snadno a efektivně sdílet a interagovat na kurzy a studijní materiály. Při návrhu této platformy budou využity aktuálně nejmodernější technologie a výsledkem bude uživatelsky intuitivní, přívětivé a rychlé rozhraní.

Práce bude prováděna systematickým postupem, zahrnujícím analýzu současného stavu v oblasti, detailní přezkum existujících nástrojů a platform, následnou komplexní analýzu požadavků pro novou webovou platformu, pečlivý návrh efektivního řešení, kritický výběr optimálních technologií pro implementaci, samotnou realizaci platformy a konečně komplexní zhodnocení dosažených výsledků.

Tato práce je rozdělena do dvou hlavních kapitol a několika podkapitol. Kapitola 3 se bude věnovat analýze aktuálního stavu vědy a techniky. Kapitola 4 se poté zaměří na samotný vývoj aplikace. Podkapitola 4.1 bude obsahovat analýzu požadavků, podkapitola 4.2 bude věnována návrhu řešení, podkapitola 4.3 se bude zabývat popisem použitých technologií, podkapitola 4.5 se bude věnovat nasazení do produkčního prostředí, podkapitola 4.6 se bude zabývat popisem funkcí aplikace. Kapitola 5 bude obsahovat další možná rozšíření.

2 Motivace

Při rozhodování o tématu mé bakalářské práce jsem hledala inspiraci v oblasti, která by mi nejen přinášela osobní zájem, ale také přispívala společnosti. Během mého studia na Univerzitě Palackého v Olomouci jsem si všimla, že sdílení souborů a spolupráce mezi studenty často představuje výzvu. Existující systémy často nedostatečně splňují potřeby univerzitního prostředí a často jsou zastaralé a nepřehledné [1].

Tato situace mě motivovala k napsání bakalářské práce na téma "Webová aplikace pro pokročilé sdílení souborů v univerzitním prostředí". Chtěla jsem vytvořit moderní a efektivní řešení, které by usnadnilo a zlepšilo proces sdílení informací, dokumentů a materiálů mezi studenty na univerzitě.

Mým cílem bylo vytvořit webovou aplikaci, která by umožňovala snadné nahrávání, organizaci a sdílení souborů v rámci jednotlivých kurzů. Aplikace měla poskytovat uživatelům přehledný a intuitivní prostředek pro správu a vyhledávání souborů, přičemž měla být flexibilní a přizpůsobitelná potřebám konkrétní univerzity.

Jeden z klíčových motivátorů mého zájmu byla možnost aktivně přispět k modernizaci a digitalizaci univerzitního prostředí, a to za využití moderních webových technologií a konceptů jako .NET, React, Redux nebo SASS. Věřím, že moderní technologie a aplikace mohou významně přispět ke zlepšení výuky, komunikace a spolupráce mezi studenty. Mé přání bylo vytvořit nástroj, který by umožňoval efektivní výměnu informací a podporoval interakci a spolupráci mezi všemi účastníky univerzitního procesu.

V neposlední řadě mým motivátorem bylo dosažení co nejvyšší možné úrovně zabezpečení. Z tohoto důvodu jsem se rozhodla zavést [Single Site On \(SSO\)](#), což je v současné době považováno za nejmodernější a nejbezpečnější metodu přihlašování.

Věřím, že tato bakalářská práce bude mít praktické využití i po obhajobě. Touto prací jsem se naučila pracovat s velmi zajímavými moderními technologiemi, které věřím, mi poskytly velmi solidní základ pro mou budoucí kariéru.

3 Analýza současného stavu

Pro účely mé bakalářské práce jsem prováděla analýzu aktuálního stavu platform pro sdílení souborů, jelikož velmi málo platform je určeno přímo pro univerzitní prostředí. Při této analýze jsem se zaměřila na existující systémy a postupy využívané studenty při sdílení informací a materiálů.

Zjistila jsem, že většina univerzit v současnosti využívá různé platformy a nástroje pro sdílení souborů, jako jsou e-learningové systémy [2], cloudové úložiště [3, 4] nebo rozesílání materiálů pomocí e-mailu. Nicméně, tyto systémy dle mého zjištění často neposkytují dostatečné možnosti pro efektivní správu a organizaci souborů, a to zejména v případě většího množství kurzů a uživatelů.

Vzhledem k tomu, že právě studentům tyto více či méně funkční systémy nevyhovují, velmi často volí neoficiální platformy pro přístup ke studijním materiálům, jako je například platforma Primát [5], platforma Fituška [6] (pro studenty Fakulty informatiky VUT v Brně), platforma Fektuška [7] (pro studenty Fakulty elektrotechniky a komunikačních technologií VUT v Brně) nebo čím dále více oblíbené Facebookové skupiny či skupiny na platformě Discord [8].

Další výzvou, kterou jsem identifikovala, je nedostatek intuitivnosti a uživatelské přívětivosti existujících systémů. Některé platformy jsou složité na ovládnání a uživatelé mají problémy s nalezením konkrétních souborů nebo správou přístupových práv.

Dále jsem zaznamenala problémy týkající se zastaralosti a nedostatečného záplatování bezpečnostních zranitelností, které neodpovídají moderním technologickým standardům a nedokáží vyhovět rostoucím bezpečnostním potřebám univerzitního prostředí.

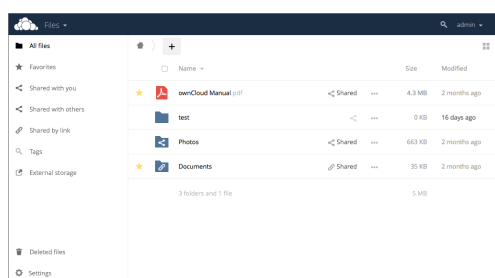
Na základě těchto zjištění jsem dospěla k závěru, že existuje potřeba pro vývoj a implementaci nové webové aplikace, která by efektivně zvládala sdílení souborů v univerzitním prostředí. Tato aplikace by měla být snadno použitelná, přehledná a flexibilní, aby vyhovovala potřebám studentů.

3.1 Platformy pro sdílení souborů

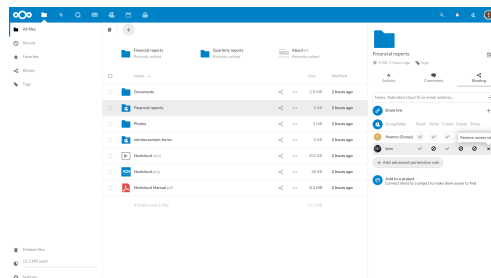
V této kapitole jsou popsány platformy pro sdílení souborů, které hrají klíčovou roli v moderním digitálním prostředí. Tyto platformy umožňují uživatelům snadno sdílet, ukládat a přistupovat k různým typům souborů, ať už jde o dokumenty, obrázky, videa nebo další multimediální obsah. Po přečtení této kapitoly budou mít čtenáři ucelený přehled o dané problematice a budou znát nejběžnější platformy, se kterými se můžou na internetu běžně setkat.

3.1.1 OwnCloud / Nextcloud

OwnCloud[9] je open-source cloudová platforma, která umožňuje uživatelům ukládat, sdílet a spravovat své soubory a data online. Jeho vznik se datuje do roku 2010, kdy byl vyvinut v reakci na rostoucí poptávku po soukromých a bezpečných cloudových řešeních. OwnCloud nabízí uživatelům široké spektrum



(a) Uživatelské rozhraní OwnCloud



(b) Uživatelské rozhraní NextCloud

Obrázek 1: Ukázka uživatelského rozhraní ownCloud/NextCloud

funkcí, jako je synchronizace souborů, sdílení složek, kalendáře, kontakty a možnost online úpravy dokumentů. Důležitým prvkem je také možnost hostování vlastního serveru, což umožňuje uživatelům úplnou kontrolu nad svými daty. Ukázka uživatelského rozhraní je vidět na obrázku 1a.

NextCloud[10] je open-source cloudová platforma, která vznikla v roce 2016 jako odnož projektu OwnCloud[11] kvůli neshodám vývojářů ohledně strategického směřování projektu. NextCloud je postaven na základech projektu OwnCloud, ale přináší mnoho vylepšení a nových funkcí. Uživatelé mohou očekávat podobné funkce jako u OwnCloud, jako je sdílení souborů, kalendáře a kontakty. Dalšími přidanými funkcemi jsou například rozšířená podpora pro mobilní zařízení, integrace s dalšími cloudovými službami a lepší zabezpečení dat. Ukázka uživatelského rozhraní je vidět na obrázku 1b.

Rozhodování mezi OwnCloud a NextCloud bude záviset na individuálních potřebách a požadavcích uživatelů. Ti, kteří preferují stabilní a osvědčené řešení, mohou zůstat u OwnCloud. Naopak ti, kteří hledají inovativní a moderní funkce, mohou dát přednost NextCloud. V každém případě, obě platformy poskytují uživatelům kontrolu nad jejich daty a umožňují bezpečné sdílení a spolupráci s ostatními uživateli.

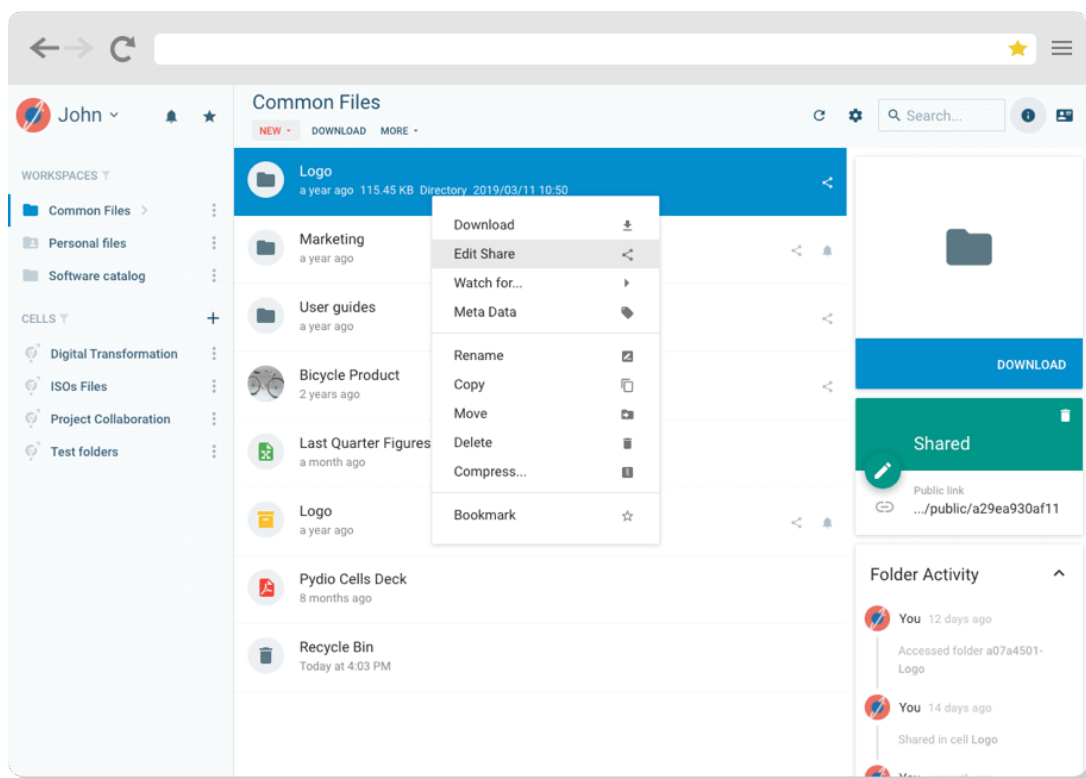
OwnCloud například umožňuje využívat provozovatel národní infrastruktury pro vědu a výzkum Cesnet[12, 13]. S NextCloud je možné se setkat například na katedře informatiky, PF, UPOL.

3.1.2 Pydio

Pydio Cells[15] je moderní a výkonná platforma pro sdílení souborů, která nabízí pokročilé funkce a flexibilitu. Tato open-source aplikace umožňuje uživatelům snadno nahrávat, spravovat a sdílet soubory prostřednictvím webového rozhraní. Zdrojové kódy jsou k dispozici na platformě Github[16].

Jedním z hlavních přínosů platformy Pydio Cells je její uživatelsky přívětivé prostředí (viz. obrázek 2), které je intuitivní a snadno ovladatelné. Uživatelé mají přehled o svých souborech a mohou je organizovat do složek a podadresářů, čímž usnadňují správu a vyhledávání potřebných informací.

Pydio Cells také poskytuje rozsáhlé možnosti nastavení přístupových práv



Obrázek 2: Ukázka uživatelského rozhraní Pydio [14]

a sdílení souborů. Uživatelé mohou definovat přesné oprávnění pro jednotlivé osoby nebo skupiny, což zajišťuje bezpečnost a ochranu citlivých dat. Dále je možné sdílet soubory s externími uživateli pomocí odkazů nebo pozvánek, což usnadňuje spolupráci a komunikaci s lidmi mimo univerzitní prostředí.

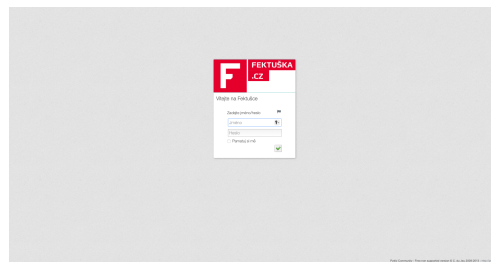
Další výhodou Pydio Cells je jeho rozšiřitelnost a možnost přizpůsobení. Díky otevřenému zdrojovému kódu je možné provádět rozšíření a úpravy aplikace dle specifických potřeb např. univerzity. Pydio tak může být integrován s dalšími univerzitními systémy a technologiemi, což usnadňuje propojení s existujícími infrastrukturami.

Pydio Cells je však velmi složitá platforma, vyžadující znalost velkého množství technologií k tomu, aby bylo možné zdrojový kód upravovat k potřebám organizace nebo univerzity. Jeho back-endová část je napsána v jazyku Go a front-endová část v jazyku React. Zejména použití jazyka Go (který vychází z jazyka C) dělá z této platformy velmi složitě upravitelnou aplikaci.

Ještě před tím, než vzniklo Pydio Cells, existovala verze Pydio, která byla postavena na klasických webových technologiích, jako je [PHP: Hypertext Preprocessor \(PHP\)](#) na back-endu a JavaScript na front-endu. Tato verze byla mnohem lépe upravitelná a zejména bylo mnohem jednodušší tuto aplikaci nahrát například na komerční hostingové služby. Verze Pydio Cells vyžaduje běh v Docker[17] Containeru. To způsobuje problémy s přístupem, kdy buď musí na Pydio Cells být dedikovaná veřejná IP adresa, nebo musí být před Pydio Cells předřazen re-



(a) Přihlašovací obrazovka



(b) Uživatelské rozhraní Pydio

Obrázek 3: Ukázka uživatelského rozhraní Fektušky

verzní proxy server. Obě varianty jsou ale náročné a je velmi složité najít vhodný komerční hosting. Z tohoto pohledu se nabízí jako nejjednodušší varianta pronájem prostoru na oficiálních stránkách Pydio Cells, avšak při větším množství uživatelů je využití této varianty extrémně finančně náročné.

3.2 Neoficiální univerzitní portály

Tato sekce se zabývá neoficiálními platformami, které založili a spravují studenti z příslušných škol a fakult. Již samotná existence těchto platform a portálů pouze dokazuje, jak je systém školství v tomto směru špatně nastaven. Pokud by byl nastaven správně, tyto platformy by totiž vůbec neměly důvod vzniknout.

3.2.1 Fektuška

Fektuška je neoficiální fórum studentů Fakulty elektrotechniky a komunikačních technologií Vysokého učení technického v Brně (FEKT VUT) [18]. Využití této starší verze Pydio (jak je vidět na obrázku 3b) je však v tomto případě nevhodné, jelikož tato verze je spravována pouze komunitně a nereaguje tak pružně na nově nalezené zranitelnosti, jako oficiální verze. Verze nasazená na Fektušce je podle dostupných informací z roku 2013, což je v čase psaní této práce 10 let stará verze.

Je důležité zdůraznit, že i když se jedná o neoficiální verzi fóra, přihlašování do Fektušky je integrováno s VUT Centrálním přihlašovacím systémem (SSO) v rámci Microsoft Azure. To znamená, že studenti a zaměstnanci mají možnost použít své stejné přihlašovací údaje jako pro přístup do ostatních informačních systémů VUT. Tímto je zajištěna vyšší úroveň bezpečnosti, neboť SSO minimalizuje potřebu uchovávání uživatelských účtů a hesel. Více o tomto způsobu autentizace je popsáno v kapitole 3.6.

Pydio na Fektušce umožní studentům FEKT VUT sdílet soubory, spolupracovat na projektech a komunikovat efektivněji v rámci školního prostředí.

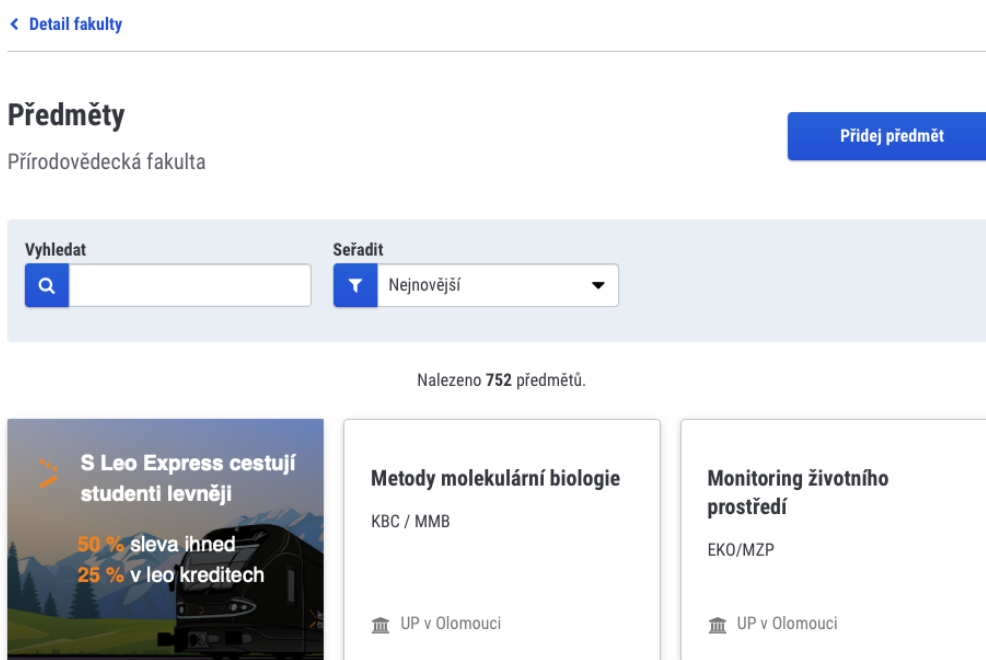
3.2.2 Fituška

Fituška je neoficiální fórum pro studenty Fakulty informačních technologií Vysokého učení technického v Brně (VUT FIT) [19]. Toto fórum nese podpis firmy Grow Job Institute [20], což naznačuje, že vzniklo pravděpodobně z iniciativy bývalých studentů této fakulty.

Co se týče technologií, na nichž je toto fórum postaveno, je možné odhalit některé informace ze zdrojového kódu. Aplikace je pravděpodobně napsána v jazyce PHP, pravděpodobně na základě platformy PHPbb.

PHPbb[21] je open-source fórum napsané v programovacím jazyce PHP. Jedná se o jedno z nejpopulárnějších fór, které poskytuje uživatelům prostředí pro diskuzi, sdílení informací, zakládání témat a komunikaci v různých diskusních fórech. PHPbb má dlouhou historii a je vysoce oblíbené mezi komunitami, jako jsou studenti, programátoři nebo nadšenci do počítačových technologií.

3.2.3 Primát.cz



Obrázek 4: Ukázka uživatelského rozhraní Primat.cz včetně zobrazené reklamy

Primát[5] je projekt soukromé firmy Primat.cz s.r.o., který je zapsán v veřejném rejstříku Justice. Tato firma je vedena Maximem Gorjainovem a Adamem Koželou. Primát představuje komerční webový portál určený pro studenty, sloužící k sdílení studijních materiálů. Hlavní způsob, jakým portál generuje příjmy pro svůj provoz, spočívá v zobrazování reklam, jak je vidět na Obrázku 4.

Podle informací dostupných na Justice.cz byl tento portál založen v roce 2009.

Od té doby prošel několika verzemi, jak dokumentuje Wayback Machine¹ [22]. Primát se tak postupně vyvíjel a měnil, aby lépe vyhovoval potřebám uživatelů.

Aktuální verze Primáta je dostupná pro kohokoliv, kdo se zaregistruje. Ověřování uživatelů probíhá pouze na základě ověření emailové adresy. Platforma samotná ze své podstaty nemůže mít přehled o identitě svých uživatelích.

3.3 Vzdělávací platforma Moodle

My courses

Course overview

The screenshot displays the 'My courses' section of a Moodle user interface. At the top, there are filters for 'In progress', a search bar, 'Sort by course name', and a 'Card' view selector. Below these are six course cards arranged in a 2x3 grid. Each card features a representative image, the course title, the community name, and a progress percentage. The courses shown are: 'Mindful course creation' (50% complete), 'Moodle and Mountaineering' (0% complete), 'Activity Examples' (no progress shown), 'Celebrating Cultures' (27% complete), 'Class and Conflict in World Cinema' (50% complete), and 'Critical Thinking: Develop your skills' (66% complete).

Course Title	Community	Progress
Mindful course creation	Mount Orange Community	50% complete
Moodle and Mountaineering	Physical Education	0% complete
Activity Examples	Moodle Resources	-
Celebrating Cultures	Society and Environment	27% complete
Class and Conflict in World Cinema	Art and Media	50% complete
Critical Thinking: Develop your skills	Society and Environment	66% complete

Obrázek 5: Ukázka uživatelského rozhraní Moodle

Moodle[23] je open-source vzdělávací platforma, která slouží jako nástroj pro moderní studium a online výuku. Jeho název je zkratkou pro “Modular Object-Oriented Dynamic Learning Environment“ (Modulární objektově orientované dynamické vzdělávací prostředí) a byl poprvé uveden v roce 2002. Od té doby se stal jedním z nejvíce využívaných systémů pro správu výuky a e-learning. Naplno se využil potenciál této platformy během světové pandemie koronaviru, kdy

¹Wayback Machine je internetová služba, která ukládá a uchovává archivy webových stránek z minulosti, umožňující uživatelům prohlížet historické verze webů a sledovat jejich vývoj v čase.

Moodle zaznamenal rekordní počty registrací, stažení a studentských přístupů [24].

Moodle nabízí širokou škálu funkcí, které usnadňují interakci mezi studenty a učiteli v online prostředí. Mezi klíčové prvky patří možnost zakládání a správy kurzu, nahrávání studijních materiálů, zadávání úkolů, online testování a komunikace v rámci diskuzních fór. Díky své modularitě a otevřené architektuře umožňuje Moodle snadné rozšiřování funkcí pomocí různých doplňků a rozšíření dle potřeb učitelů a studentů.

Jednou z nejcennějších vlastností Moodle je jeho flexibilita. Učitelé mohou přizpůsobit výuku podle svých individuálních potřeb a preferencí a studenti mají možnost učit se svým tempem. Zajišťuje také sledování pokroku studentů, což umožňuje učitelům lépe pochopit, jak studenti studují a kde je potřeba poskytnout dodatečnou podporu. Moodle také podporuje interakci a spolupráci mezi studenty a umožňuje vytvářet skupinové projekty a diskutovat o tématu v reálném čase.

V dnešní době je Moodle využíván na mnoha vzdělávacích institucích po celém světě – od škol a univerzit až po korporátní vzdělávací programy. Podle aktuálních oficiálních statistik[25] v době psaní této práce platformu využívalo více než 154 tisíc organizací a firem, spolu s více než 377 miliony aktivních uživatelů, čímž se Moodle stává nejrozšířenější vzdělávací platformou na světě[26].

3.4 Sociální sítě

Sociální sítě se staly v dnešní digitální éře nedílnou součástí našeho každodenního života a postupně nacházejí své místo i ve vzdělávání. Tyto platformy se ukázaly jako inovativní nástroj pro interaktivní vyučování a zlepšení komunikace mezi studenty a učiteli. Integrace sociálních sítí do vzdělávacího procesu nabízí řadu výhod i nevýhod[27].

3.4.1 Facebookové skupiny

Facebookové skupiny mají v současné době značný vliv na vzdělávání a přinášejí mnoho pozitivních aspektů. Tyto online komunitní platformy umožňují vytváření zaměřených a interaktivních skupin spojených s konkrétním vzdělávacím tématem, předmětem nebo institucí. Díky nim se otevírá nový rozměr spolupráce, sdílení informací a učení mezi studenty. Ve starším článku z roku 2013 s názvem “Facebook Groups as LMS: A Case Study“ od akademických autorů Hagit Meishar-Tal, Gila Kurtz a Efrat Pieterse z Izraele je zdůrazněno, že využívání facebookových skupin představuje přínos, protože umožňuje studentům udržovat kontakt i po ukončení kurzu, což má pozitivní vliv na jejich kariérní a sociální rozvoj[28].

Nicméně, je také důležité brát v úvahu některé potenciální negativní vlivy. Například není výjimkou, že studenti mimo oficiální skupiny zakládají neoficiální skupiny s cílem vyloučit učitele nebo část studentů, což může vést k pocitu

vyčlenění ze skupiny. Navíc velké skupiny se mohou dělit na menší a menší, kde je obtížné, ne-li nemožné, obsah monitorovat.

Dalším faktorem je skutečnost, že Facebookové skupiny mohou způsobit rozptýlení studentů a snížit jejich produktivitu, pokud nejsou řádně řízeny nebo se stanou místem pro nevhodnou komunikaci. Je proto důležité, aby učitelé aktivně monitorovali a řídili obsah a průběh diskuzí v rámci skupiny.

3.4.2 Discord

Discord[29], původně vyvinutý jako platforma pro komunikaci mezi hráči počítačových her, se postupem času stal velice často užívaným nástrojem v univerzitním prostředí[30]. Jednou z hlavních výhod využití Discordu ve vzdělávání je jeho schopnost usnadnit komunikaci mezi studenty a vyučujícími. Díky kombinaci textových, hlasových a video chatů mohou účastníci rychle sdílet informace, diskutovat o studijních tématech a projektech a spolupracovat na úkolech.

Další výhodou je jeho flexibilita a dostupnost. Discord je přístupný z různých zařízení, což umožňuje studentům přistupovat k diskuzím a materiálům kdykoliv a odkudkoliv. To může být zejména pro studenty v online vzdělávacím prostředí velmi užitečné. Organizace skupin a týmů je další významnou výhodou Discordu. Platforma umožňuje vytvářet různé kanály pro různé účely, což usnadňuje strukturování a organizaci diskuzí. Každý kanál může být věnován konkrétní třídě, projektu nebo tématu.

Nicméně je také třeba brát v úvahu některé nevýhody. Jelikož byl Discord původně vytvořen pro komunikaci v rámci her, existuje riziko, že studenti se mohou stát závislími na platformě a trávit na ní nadměrně času na neakademických aktivitách. Další nevýhodou může být nedostatečná formalita komunikace. Používání Discordu může vést k uvolněnější a méně formální komunikaci, což může být nevhodné v některých akademických situacích.

Omezená struktura a schopnost organizace je další aspekt, který by mohl být problémový v univerzitním prostředí. Discord není vždy ideálním nástrojem pro organizaci složitých projektů a správu velkého množství informací. Řada vysokých škol v ČR se s tímto problémem pere tak, že školní spolky zakládají a spravují kanály pro studenty před začátkem školního roku, aby se zamezilo tříštění studentů do více kanálů a podskupin.

Nutno podotknout, že soukromí a ochrana osobních údajů studentů může být na Discordu problematická. Protože platforma byla původně navržena pro zábavu, nemusí poskytovat dostatečnou úroveň zabezpečení a ochrany dat. V současné době je nejčastěji využívaným způsobem ověření studenta kombinace školního emailového účtu a ověřovacího emailu.

3.4.3 Microsoft Teams

V uplynulých letech, s významným zřetelem k pandemii COVID-19, která výrazně ovlivnila způsob výuky a komunikaci, se Microsoft Teams[31] osvědčil jako

neodmyslitelný prvek moderního vzdělávání. Tato platforma, původně koncipovaná pro podnikové účely, rychle našla své místo i ve vzdělání, kde se stala účinným nástrojem propojujícím studenty, pedagogy a výukový obsah v digitálním prostoru. Důsledkem toho byly nové příležitosti pro komunikaci, spolupráci a vedení výuky, což se stalo nejen reakcí na aktuální situaci, ale také budoucím směřováním moderní vzdělávací praxe. Schůzky prostřednictvím této platformy vzrostly během pandemie o více než polovinu[32], což jen dokazuje, jak zásadním nástrojem se Teams stal ve výukovém procesu.

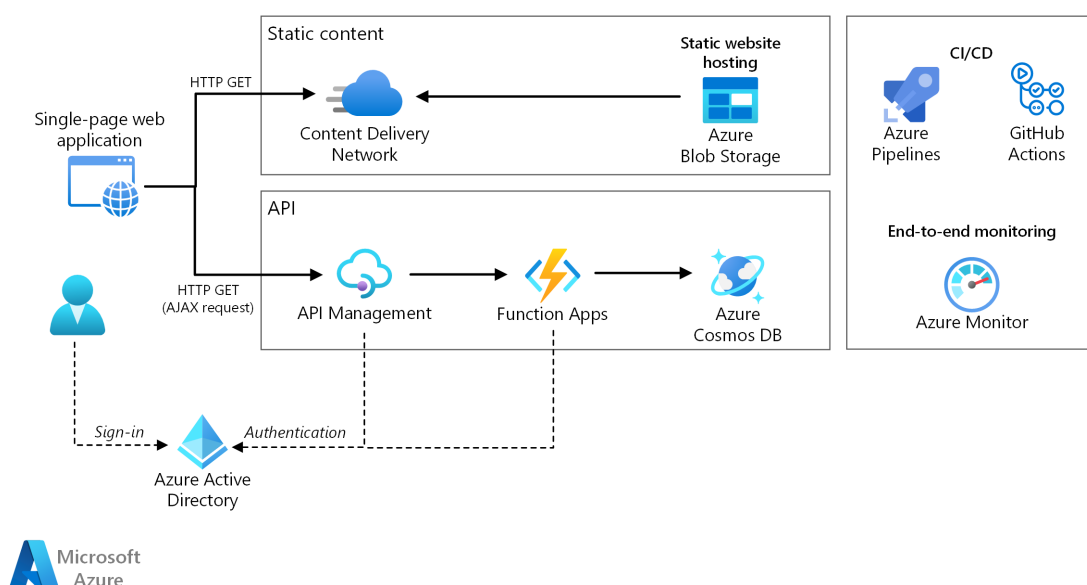
Microsoft Teams nabízí širokou škálu funkcí, které podporují interaktivitu a komunikaci mezi účastníky. Skrze chatovací kanály a diskuse mohou studenti spolu i se svými učiteli komunikovat, sdílet nápady a řešit otázky mimo učební dobu. Tím se podporuje kontinuální tok informací a vytváří se prostor pro kreativní myšlení.

Dalším významným přínosem je možnost organizace a správy výukových materiálů. Učitelé mohou vytvářet a sdílet obsah, jako jsou prezentace, dokumenty a úkoly, přímo na platformě. To usnadňuje přístup k materiálům a studenti tak mají k dispozici veškeré potřebné informace na jednom místě.

3.5 Aktuální trendy webových technologií

V dynamickém světě webových technologií neustále dochází k inovacím a posouvání hranic možností. Tato kapitola se detailně zabývá několika klíčovými trendy, které v současnosti formují a ovlivňují vývoj webových technologií.

3.5.1 Serverless Architektura



Obrázek 6: Serverless architektura Microsoft Azure [33]

Serverless architektura je dalším významným trendem, který mění způsob, jakým vývojáři budují a nasazují své aplikace. Tento přístup k vývoji umožňuje psát kód a nasazovat aplikace bez starostí o konfiguraci a správu serverů. Cloudoví poskytovatelé jako [Amazon Web Services \(AWS\)](#), Microsoft Azure[34] a Google Cloud nabízejí služby serverless výpočtu, jako jsou [AWS Lambda](#)[35] nebo [Azure Functions](#)[36].

Serverless architektura umožňuje vývojářům koncentrovat se na psaní kódu a vytváření funkcí, které reagují na určité události. Cloudový poskytovatel se pak stará o spuštění a škálování těchto funkcí dle aktuální potřeby. To výrazně usnadňuje nasazování, škálování a správu aplikací, a zároveň může vést k úsporám nákladů na provoz [37].

3.5.2 Rozvoj Moderního Front-Endu

Moderní webové aplikace procházejí vývojem směrem k interaktivním a vizuálně atraktivním uživatelským rozhraním. Tento trend vede k rostoucímu využití JavaScriptových frameworků a knihoven, například [React](#)[38], [Angular](#)[39] a [Vue.js](#)[40]. Tyto nástroje umožňují vytvoření pokročilých uživatelských rozhraní s animacemi, dynamickým načítáním obsahu a rychlou odezvou. To usnadňuje implementaci složitých funkcí, jako je real-time vyhledávání, filtrace obsahu a další interakce, které zlepšují UX (User Experience, neboli uživatelskou zkušenost).

Souběžně s vývojem JavaScriptových nástrojů se zlepšuje také nástrojová sada pro vývoj, testování a ladění front-endového kódu. Nástroje jako [Webpack](#)[41], [Babel](#)[42] a [ESLint](#)[43] umožňují efektivní práci a zajišťují vysokou kvalitu kódu. Webpack umožňuje balit moduly do souborů, které jsou optimální pro nasazení, Babel umožňuje překládat moderní JavaScript do podporovaných verzí prohlížečů a ESLint slouží k udržení konzistence a správné syntaxe kódu.

V současné době existuje také důraz na responzivní design, který zajišťuje, že webová aplikace se správně zobrazí na různých zařízeních a obrazovkách. [Progressivní Webová Aplikace \(PWA\)](#) nabízí možnost kombinovat vlastnosti webových stránek a nativních aplikací, což umožňuje offline přístup a lepší výkon. Automatizované testování, kontinuální integrace a nasazení ([Continuous Integration \(CI\)](#)/[Continuous Deployment \(CD\)](#)) pomáhají zajistit kvalitu kódu a efektivní aktualizace aplikace, jak je vidět na [Obrázku 6](#).

V neposlední řadě je důležité zmínit také význam [Search Engine Optimization \(SEO\)](#) a rychlost načítání webových stránek. Techniky jako [lazy loading](#)² a optimalizace obrázků pomáhají zrychlit načítání stránek a zlepšit uživatelský zážitek.

²Koncept lazy loadingu se v informatice vztahuje k optimalizaci načítání zdrojů, dat nebo komponent v softwarových systémech. Tento přístup se zaměřuje na načítání zdrojů až v okamžiku, kdy jsou skutečně potřebné, což má za následek zrychlení načítání aplikace a úsporu systémových zdrojů.

3.5.3 Single Page Aplikace (SPA)

Single Page Application (SPA) je moderní přístup k vývoji webových aplikací, který zásadně mění způsob, jakým uživatelé interagují s online obsahem. SPA odstraňují tradiční model načítání celých stránek tím, že načítají pouze jednu stránku při prvním načtení a následně provádějí asynchronní³ načítání obsahu na pozadí [44]. Tento přístup výrazně zvyšuje rychlost a plynulost uživatelského zážitku, protože webová stránka zůstává aktivní a reaguje na uživatelské interakce bez nutnosti opakovaného načítání celé stránky. Díky architektuře SPA mohou uživatelé plynule přecházet mezi různými částmi aplikace, aniž by zažili tradiční zpoždění spojené s načítáním nových stránek.

SPA využívají moderní webové technologie, jako je JavaScript, AJAX (asynchronní JavaScript a XML), a SPA frameworky (např. Angular, React, nebo Vue.js, které jsou podrobněji popsány v kapitole 3.5.2). Tyto nástroje umožňují vývojářům vytvářet složitá uživatelská rozhraní s interaktivními prvky a efekty.

Nicméně, při použití SPA je třeba zvládnout i některé výzvy, jako je správa stavu aplikace, složitá SEO optimalizace (problém s inicializací na straně klienta, omezená indexace obsahu, dynamická změna **Uniform Resource Locator (URL)** adresy, atd.), a podpora pro uživatele s vypnutým JavaScriptem (vytvoření záložního obsahu bez použití JavaScriptu).

3.5.4 Progresivní Webové Aplikace (PWA)

PWA představují zásadní trend ve vývoji webových technologií, který spojuje přednosti webových stránek s výhodami nativních mobilních aplikací. PWA nabízejí uživatelům možnost interakce s aplikací offline, což znamená, že uživatelé mohou stále používat aplikaci i bez aktivního internetového připojení. Díky mechanismům pro caching a načítání obsahu na pozadí se PWA načítají rychle a reagují ihned na interakce uživatele.

Kromě toho mají PWA schopnost se “nainstalovat“ na domovskou obrazovku uživatele. To znamená, že uživatel může mít ikonu PWA přímo mezi ostatními aplikacemi na svém zařízení, bez ohledu na operační systém či výrobce telefonu. Tato vlastnost zvyšuje viditelnost a dostupnost webových aplikací.

3.6 Autentizace uživatelů

Autentizace uživatelů v rámci webové aplikace je klíčovým prvkem zabezpečení, který zajišťuje, že pouze oprávnění uživatelé mají přístup k chráněným funkcím a datům. Při použití chráněné části webové aplikace se uživatel musí identifikovat a ověřit svou identitu, aby mohl využívat služby poskytované touto aplikací.

³Asynchronismus je pojem v informatice, který se týká způsobu, jakým se provádí úkoly nebo operace, které nemusí být vykonávány po sobě. Namísto toho jsou tyto úkoly prováděny nezávisle na sobě, což umožňuje pokračování v dalších operacích bez čekání na dokončení daného úkolu.

Správně implementovaná autentizace uživatelů je klíčová pro zajištění ochrany dat a soukromí uživatelů, minimalizaci rizika útoků a vytvoření důvěry vůči webové aplikaci. Je důležité pravidelně aktualizovat autentizační metody a sledovat bezpečnostní trendy, aby byla stále udržována vysoká úroveň ochrany.

3.6.1 Zastaralé možnosti autentizace

Autentizace pomocí tradičního uživatelského jména a hesla byla dlouhou dobu hojně využívána, ale postupem času se ukázala jako zastaralý způsob ověření identity uživatele. Tento přístup má několik vážných negativ, které zpochybňují jeho účinnost a bezpečnost[45]:

- **Bezpečnost** Hlavním problémem uživatelských jmen a hesel je jejich náchylnost k útokům. Uživatelé často volí slabá hesla nebo je používají na více místech, což zvyšuje riziko úspěšných útoků hackerů. Útočníci mohou také využít různé metody, jako je útok hrubou silou nebo slovníkové útoky, aby získali neoprávněný přístup.
- **Krádež citlivých údajů** Uživatelská jména a hesla mohou být snadno ukradena v případě úniku dat z webových stránek. Tento problém je stále aktuální a týká se velkého množství uživatelů.
- **Uživatelská nepřívětivost** Uživatelé si musí pamatovat různá uživatelská jména a hesla pro různé webové stránky, což může být zatěžující a nepohodlné. Tato situace může vést k volbě jednoduchých hesel nebo dokonce k ukládání údajů na papírky nebo do webového prohlížeče, čímž se ještě více narušuje bezpečnost. Kromě bezpečnostních aspektů má tradiční autentizace formou jména a hesla vliv i na uživatelskou spokojenost. Neustálé zadávání a pamatování si přihlašovacích údajů může uživatele frustrovat.

Jedním z možných řešení tohoto problému je využití správce hesel (Password Manager). Nicméně i tato možnost má své nevýhody. Správce hesel nemusí vždy garantovat dostatečně silná hesla pro každou službu, a také vyžaduje mimořádně robustní hlavní heslo (master heslo), které slouží k ochraně ostatních hesel. Dochází také k riziku ztráty všech hesel v případě ztráty hlavního hesla. Synchronizace mezi různými zařízeními může být problematická a existuje reálná pravděpodobnost existence potenciálních softwarových chyb nebo slabých šifrovacích algoritmů, které mohou být v budoucnosti zneužity. V neposlední řadě jsou password manažeři méně oblíbení mezi uživateli, než by bylo potřeba z hlediska bezpečnosti [46].

V důsledku těchto negativních aspektů se mnoho moderních webových aplikací obrací k pokročilejším metodám autentizace, které nabízejí vyšší bezpečnost a uživatelskou pohodlnost. Autentizace pomocí jména a hesla je stále používána, ale často je kombinována s dalšími vrstvami ochrany, jako je dvoufaktorová autentizace nebo biometrické metody. Je třeba však zmínit, že i tyto dvoufaktorové metody mohou představovat určitou zátěž pro uživatele. Pro použití dvoufaktorové autentizace musí uživatelé často mít nainstalované speciální mobilní

aplikace, zvládnout získat ověřovací [Short Message Service \(SMS\)](#) nebo mít k dispozici hardwarový klíč (token) [47].

3.6.2 Moderní způsoby autentizace

Moderní způsoby autentizace přinášejí inovativní a efektivní přístup, jak dosáhnout bezpečnosti přihlašování a ochrany osobních údajů. Klasické heslo, byť stále užitečné, již nestačí k zajištění důkladné bezpečnosti. Proto se vývoj zaměřuje na vytváření sofistikovanějších metod, které umožňují bezpečný a pohodlný přístup k online službám.

Dvoufaktorová autentizace je krok správným směrem k vyšší bezpečnosti přihlašování. Mimo běžného hesla vyžaduje od uživatele další autentizační prvek, jako je jednorázový kód z mobilní aplikace nebo zpráva [SMS](#). Tím se výrazně ztěžuje možnost neoprávněného přístupu i v případě, že by heslo bylo ohroženo [47].

Kromě toho existuje řada moderních autentizačních metod, které zjednodušují a zefektivňují proces přihlašování. OAuth⁴ je jedním z takových standardů, umožňujícím aplikacím získat omezený přístup k uživatelským datům na základě povolení, aniž by musel uživatel sdílet své hlavní přihlašovací údaje. Toho využívají velké firmy jako Google, Apple nebo Microsoft v rámci svých integrovaných služeb. Toto jednotné přihlašování je poté možné integrovat do vlastních aplikací pomocí [SSO](#), které umožňuje uživatelům využít jedinou sadu přihlašovacích údajů pro přístup k různým službám. To eliminuje potřebu pamatovat si mnoho různých hesel a zjednodušuje celý proces.

Posledním nástrojem v této oblasti je možnost přihlášení pomocí sociálních sítí, což umožňuje uživatelům využít své údaje z populárních platforem pro rychlý a pohodlný přístup na různé webové stránky. Tato metoda funguje na podobném principu jako SSO a nabízí další možnost, jak minimalizovat množství hesel a zjednodušit proces autentizace.

3.6.2.1 Microsoft Azure Active Directory

Microsoft Azure [Active Directory \(AD\)](#)[34] je cloudová služba pro správu identit a přístupových práv vyvinutá společností Microsoft. Tato služba slouží k centralizovanému řízení identit uživatelů, správě přístupových práv a zabezpečení aplikací, a to jak v cloudu, tak i v lokálních IT prostředích organizace.

Azure [AD](#) poskytuje robustní funkce pro jednotné přihlašování ([SSO](#)), což umožňuje uživatelům přihlásit se pouze jednou a získat přístup k různým aplikacím a službám bez nutnosti opakovaného zadávání přihlašovacích údajů. To zvyšuje pohodlí pro uživatele a zároveň zlepšuje bezpečnost tím, že snižuje riziko vytvoření slabých hesel.

⁴OAuth je otevřený standard pro autentizaci a autorizaci, který umožňuje aplikacím získat omezený přístup k účtům uživatelů na jiných webových službách, aniž by musely sdílet citlivá hesla.

Azure AD také umožňuje správu identit a přístupových práv různými způsoby, včetně jednoduchého přidávání a odebrání uživatelů, nastavování skupin a rolí pro řízení přístupu k různým zdrojům a aplikacím. Díky integraci s dalšími službami Microsoft Azure a nástroji pro správu zabezpečení nabízí Azure AD uživatelům možnost víceúrovňové autentizace, správu zařízení a monitorování aktivit.

3.7 Shrnutí

Sledování aktuálních trendů ve vývoji moderních webových aplikací je klíčovým prvkem pro udržení konkurenceschopnosti a zajištění optimálního uživatelského zážitku. V dnešní době, kdy se technologie rychle vyvíjí, je nezbytné držet krok s inovacemi a přizpůsobit se novým požadavkům uživatelů.

Příklon ke starým a překonaným technologiím může znamenat vážné omezení schopnosti aplikace efektivně reagovat na potřeby uživatelů. Staré technologie mohou být zranitelné vůči bezpečnostním hrozbám, pomalé nebo neslučitelné s moderními zařízeními a prohlížeči. Používání zastaralých nástrojů může vést k obtížím při údržbě, rozšiřování a aktualizacích aplikace, což zase může způsobit vyšší náklady a ztrátu konkurenceschopnosti. Kromě toho je důležité si uvědomit, že technologický vývoj může být neúprosný a rychlý. Co dnes může být považováno za moderní, se může za krátkou dobu stát zastaralým a nepoužitelným. V takovém případě se může ocitnout situace, kdy aplikace, postavená na zastaralých technologiích, dosáhne svého limitu a bude nutné provést zásadní změny. Přepsání celé aplikace do novějších technologií může být nákladné a časově náročné, čímž může dojít k podstatnému zpomalení rozvoje a ztrátě konkurenční výhody. Je proto moudré nejen sledovat aktuální trendy, ale také být připraven(a) přizpůsobit se rychlým změnám a včas investovat do moderního technologického základu aplikace.

Sledování aktuálních trendů umožňuje využít nové technologie a postupy, které mohou výrazně zvýšit efektivitu, rychlost a kvalitu vývoje. Moderní frameworky a knihovny často nabízejí lepší nástroje pro správu stavu, routování, práci s daty či optimalizaci výkonu. Rovněž mohou zahrnovat pokročilé nástroje pro testování, automatizaci a kontinuální integraci, což vše přispívá k rychlejšímu a stabilnějšímu vývoji. Svět moderního vývoje webu se neustále mění a vyvíjí. Trvání na zastaralých technologiích a odmítání nových trendů může brzdit inovace a růst.

4 Vývoj vlastní aplikace

Tato sekce bakalářské práce se zaměřuje na vhled do procesu vývoje vlastní aplikace, přičemž předkládá klíčové aspekty a rozhodnutí, která přispívají k úspěšné realizaci této webové aplikace. S využitím moderních technologií, návrhových postupů a průpravných analytických fází se tato sekce snaží osvětlit nejen samotný technický proces vývoje, ale také hlediska a přístupy, které směřují k vytvoření efektivní, uživatelsky přívětivé webové aplikace s přidanou hodnotou.

4.1 Analýza Požadavků

Většina požadavků na aplikaci vychází z úvodního zadání této práce. Nicméně, aplikace by měla splňovat i následující funkční požadavky, které přesahují rámec základního zadání:

- Zajištění bezpečného přihlašování uživatelů,
- Možnost komentovat a hodnotit soubory,
- Implementace efektivního vyhledávání souborů a kurzů,
- Vytvoření notifikačního systému pro zásadní události.

V kontextu nefunkčních požadavků, které se odvodily z analýzy aktuálního stavu, jsou definovány následující body:

- Zajištění nepřetržité dostupnosti aplikace s minimálními plánovanými výpadky,
- Ochrana citlivých uživatelských dat,
- Zajištění rychlé odezvy na uživatelské interakce,
- Podpora různých typů souborů a formátů.

Kromě výše zmíněných požadavků byly také stanoveny specifické technické požadavky, které budou v rámci programování aplikace zohledněny:

- Zavedení moderního způsobu deploymentu,
- Implementace automatizovaného procesu deploymentu,
- Navržení architektury aplikace s využitím serverless technologie.

Tyto informace poskytují vhled do potřeb a očekávání, která jsou spojena s budoucí webovou aplikací. Na základě zkoumání funkčních, nefunkčních a technických požadavků je nyní jasný přehled o tom, co by tato platforma měla dosáhnout a jak by měla optimálně fungovat. Tyto cenné informace nám poskytují solidní základ pro další fázi návrhu a vývoje, kdy se budou systematicky převádět tyto požadavky do konkrétních prvků a funkcionalit, které budou uživatelé moci aktivně využívat.

4.2 Návrh aplikace

Z technického pohledu návrh aplikace představuje definování architektury, komponent, datových toků a technických postupů, které tvoří základ pro vývoj a implementaci úspěšné aplikace. Tento krok je klíčový pro zajištění efektivity, škálovatelnosti a udržitelnosti aplikace v průběhu celého životního cyklu.

4.2.1 Návrh architektury

Návrh architektury aplikace je rozdělen do několika klíčových vrstev, které spolu navzájem spolupracují. Na straně frontendu byl zvolen moderní a oblíbený JavaScriptový framework React. Díky jeho komponentní architektuře jsou rozhraní rozdělena do znovupoužitelných a samostatných komponent. Aplikace je poté strukturována do logických sekcí, přičemž každá sekce obsahuje související komponenty, styly a logiku.

Na straně backendu je využit framework .NET, který poskytuje robustní základnu pro správu dat, řízení toku aplikace a komunikaci se serverem. Architektura aplikace je koncipována s ohledem na principy RESTful API⁵, díky nimž je umožněno propojení frontendu a backendu prostřednictvím jednotného a konzistentního rozhraní.

Zásadním aspektem architektury je řízení stavu. K tomuto účelu je využíván správce stavu Redux pro React, čímž je umožněno centrálně uchovávat a spravovat stav aplikace. Tím je zajištěno konzistentní chování aplikace i v situacích s komplikovanými interakcemi.

4.2.2 Návrh uživatelského rozhraní

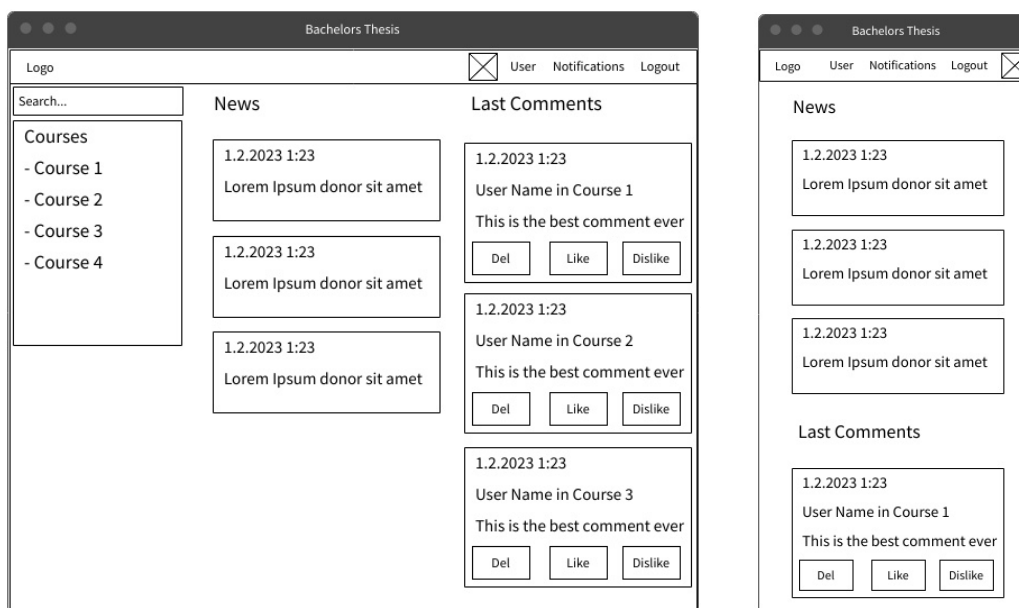
Jako klíčový krok v procesu návrhu rozhraní byl vytvořen wireframe, což je vizuální náčrt, který ilustruje umístění jednotlivých prvků na obrazovce.

Desktopová verze byla zaměřena na vytvoření přehledného a uživatelsky přívětivého rozhraní. Wireframe obsahuje strategicky umístěné prvky, jako jsou hlavní navigační menu, vyhledávací lišta a klíčové funkce aplikace. Byl zohledněn také responsivní design, aby bylo zajištěno optimální zobrazení na různých velikostech obrazovky.

Verze pro mobilní telefony byla zaměřena na vytvoření kompaktního a intuitivního designu. Wireframe pro tuto verzi klade důraz na snadný přístup k hlavním funkcím aplikace prostřednictvím navigačního menu či tlačítek. Byla zohledněna ergonomie a uspořádání prvků pro pohodlné používání na menších displejích.

Wireframe pro obě verze aplikace je vidět na obrázku 7.

⁵RESTful API (Representational State Transfer) je architektonický styl pro návrh webových služeb a rozhraní, který usnadňuje komunikaci mezi různými softwarovými systémy pomocí standardních HTTP metod.



(a) Počítač

(b) Mobilní telefon

Obrázek 7: Wireframe uživatelského rozhraní

4.2.3 Návrh databázové struktury

Při návrhu databázové struktury byl kladen důraz na vytvoření efektivního a spolehlivého schématu, které bude schopno zajistit veškerou funkcionalitu. Nejprve byla provedena pečlivá analýza potřeb aplikace a byly definovány datové entity (modely). Na základě této analýzy byly identifikovány klíčové tabulky a vztahy mezi nimi, což umožnilo vytvořit koherentní datový model, který je popsán níže:

- **Course** - Tabulka obsahující seznam všech kurů
- **CourseFile** - Tabulka obsahující soubory v jednotlivých kurzech
- **Reaction** - Tabulka obsahující informace o reakcích (like/dislike) k jednotlivým souborům a komentářům
- **Log** - Tabulka ukládající všechny události aplikace
- **Comment** - Tabulka komentářů
- **News** - Tabulka novinek
- **Notification** - Tabulka nastavených notifikací
- **User** - Tabulka uživatelů

Návrh probíhal s cílem minimalizovat redundanci a zajistit konzistenci dat. Každá tabulka byla navržena s ohledem na klíčová pole a primární klíče, jež zajišťují integritu dat a umožňují efektivní dotazování.

4.3 Použité technologie

V této kapitole se zaměřím na detailní popis použitých technologií při vývoji webové aplikace. Moderní webový vývoj vyžaduje široké spektrum nástrojů a technologií, které umožňují tvorbu interaktivních a uživatelsky přívětivých webových stránek a aplikací. Od front-endových frameworků a knihoven, přes [Cascading Style Sheets \(CSS\)](#) preprocesory a responzivní design, až po back-endové technologie, jako je .NET a databázové systémy, je výběr správných technologií klíčový pro zajištění bezproblémového provozu moderní webové aplikace.

4.3.1 Front-End

Front-end vývoj je klíčovým pilířem současné podoby webových aplikací. Moderní technologie a trendy v oblasti front-endu neustále mění způsob, jakým interagujeme s webovými stránkami a aplikacemi. Od responzivního designu až po pokročilé JavaScriptové knihovny, jako je React.js[38] a Vue.js[40], front-end vývojáři se neustále snaží vytvářet esteticky působivá, rychlá a uživatelsky přívětivá uživatelská rozhraní. Statické generátory, typová kontrola (TypeScript), [PWA](#) a optimalizace pro tmavý režim jsou jen některé z aspektů, které formují moderní front-end.

4.3.1.1 React

```
1 <h2>Courses</h2>
2 {userState?.user?.isAdmin &&
3   <button className="Button mb-1" onClick={() =>
4     openAddNewCourseModal() }>
5     <img src={addIcon} alt="Add icon" />
6     Add course
7   </button>
8 }
9 <div className="SearchInput">
10   <input
11     type="text"
12     value={filter}
13     onChange={event => setFilter(event.target.value.toLowerCase())
14     }
15     placeholder='Filter courses..'
16   />
17   <button onClick={() => setFilter("")}>
18     <img src={closeIcon} alt="Close icon" width="16" height="16"
19     />
20   </button>
21 </div>
```

Zdrojový kód 1: Ukázka kódu v Reactu

React je populární JavaScriptová knihovna pro tvorbu uživatelských rozhraní vyvinutá společností Facebook [38]. V průběhu času se stala klíčovým nástrojem pro moderní front-end vývoj. Jeho hlavním cílem je umožnit vývojářům vytvářet komponenty, které mohou být znovupoužity a snadno se kombinují k sestavení komplexních uživatelských rozhraní. Díky virtuálnímu [Document Object Model \(DOM\)](#) a účinnému algoritmu vykreslování je React schopný zajišťovat vysoký výkon a rychlé odezvy na uživatelské interakce. Další výhodou Reactu je jeho rozšířenost, obrovská komunita vývojářů a dostupnost mnoha doplňkových knihoven a nástrojů, které zvyšují jeho efektivitu a flexibilitu při tvorbě moderních webových aplikací. Ve zdrojovém kódu 1 můžete vidět příklad vytvoření obsahu bočního menu v aplikaci.

4.3.1.2 React router

```
1 {
2   path: RoutesList.news.url,
3   element: <News />,
4 },
5 {
6   path: RoutesList.log.url,
7   element: <Log />,
8 },
9 {
10  path: RoutesList.course.url,
11  element: <Course />,
12  children: [
13    {
14      path: RoutesList.file.url,
15      element: <File />,
16    }
17  ]
18 }
```

Zdrojový kód 2: Ukázka definování cest v React Routeru

React Router je knihovna pro React, která umožňuje vytvářet komplexní jednostránkové aplikace s více stránkami [48], jak můžete vidět ve zdrojovém kódu 2. Přestože React sám o sobě slouží k tvorbě uživatelských rozhraní, nemá vestavěnou podporu pro správu navigace mezi různými stránkami. Tato knihovna poskytuje vývojářům nástroje a komponenty pro definování navigace, obsluhu [URL](#) parametrů, dynamické routování a další funkce, které jsou klíčové pro správné fungování více stránek v React aplikacích.

4.3.1.3 Redux

Redux je stavový kontejner pro JavaScript aplikace, který usnadňuje správu stavu a jeho synchronizaci mezi komponentami. Je založen na jednosměrném

```

1  useEffect(() => {
2    dispatch(fetchCourses())
3  }, [])
4
5  function openAddNewCourseModal() {
6    dispatch(modalOpen({
7      type: 'addCourse'
8    }))
9  }

```

Zdrojový kód 3: Ukázka použití Reduxu

datovém toku, kde akce vyvolané uživatelem nebo systémem vedou k modifikaci stavu uloženého v centrálním úložišti nazývaném Store. Redukce pak definují, jakým způsobem se akce překládají do změn stavu. Tento architektonický vzor usnadňuje ladění, testování a škálování aplikace, což je zvláště užitečné pro rozsáhlé a komplexní projekty. Přestože může být inicializace a konfigurace Reduxu na začátku náročná, jeho výhody spočívají v tom, že poskytuje jasný způsob, jak sledovat a měnit stav aplikace a zajišťuje konzistenci mezi komponentami bez ohledu na jejich hierarchii. Příklad volání dispatch funkce, která aktualizuje data v Reduxu, můžete vidět ve zdrojovém kódu [3](#).

4.3.1.4 SASS

```

1  .Button {
2    padding: 0.5rem 0.75rem;
3
4    background-color: $light_blue;
5    color: $white;
6    box-shadow: 0px 8px 15px rgba(0, 0, 0, 0.1);
7    border: none;
8    border-radius: 7px;
9    font-size: 1rem;
10   cursor: pointer;
11
12   &:hover, &.active {
13     background-color: $dark_blue;
14   }
15 }

```

Zdrojový kód 4: Ukázka stylování v SASS

[Syntactically Awesome Style Sheets \(SASS\)](#) je nadstavba klasických CSS, která přináší pokročilé možnosti pro psaní a organizaci kaskádových stylů webových stránek. [SASS](#) umožňuje využívat proměnné, vnořování pravidel, mixinů

a funkce, což zjednodušuje a zpřehledňuje proces tvorby stylů. Tím usnadňuje práci s velkými projekty, uspořádání stylů do logických bloků a zároveň zvyšuje znovupoužitelnost kódu. Styl jednoduchého tlačítka s použitím proměnných a zanořování si můžete prohlédnout ve zdrojovém kódu 4. Po napsání SASS kódu je nutné jej přeložit na běžný CSS kód pro použití ve webových prohlížečích. To lze provést buď prostřednictvím nástroje na kompilaci, nebo přímo v rámci automatizace webového vývoje.

4.3.1.5 TypeScript

```
1 export type NewsType = {
2   newsID: number,
3   dateAdded: Date,
4   content: string
5 }
6
7 export type CommentType = {
8   commentID: number,
9   commentText: string,
10  dateAdded: Date,
11  userID: number,
12  courseFileID: number,
13  courseID: number,
14  categoryName: string,
15  likes: number,
16  dislikes: number,
17  reacted: number,
18  username: string
19 }
```

Zdrojový kód 5: Ukázka definování typů v TypeScriptu

TypeScript je nadstavba jazyka JavaScript, která přidává statické typování a další pokročilé funkce do vývoje webových aplikací v JavaScriptu [49]. Tím umožňuje vývojářům identifikovat a opravit chyby spojené s datovými typy již během vývoje, což vede k implementačně čistějším a bezpečnějším zdrojovým kódům. TypeScript také přináší vylepšenou inteligentní našeptávací funkcionálnítu, což zrychluje psaní kódu a zvyšuje jeho čitelnost. Díky tomu je TypeScript oblíbeným nástrojem mezi vývojáři, kteří chtějí maximalizovat produktivitu a minimalizovat chyby ve svých projektech postavených na JavaScriptu. Příklad definování typů najdete ve zdrojovém kódu 5.

4.3.2 Back-End

Základem pro spolehlivou a efektivní funkčnost moderních webových aplikací je kvalitní back-endová infrastruktura. Back-end představuje skrytou, ale zásadní

část aplikace, která zajišťuje zpracování dat, logiku aplikace a komunikaci s databází. Jeho úloha je nejen zajistit správné fungování uživatelských požadavků, ale také zabezpečit data a zajišťovat rychlou odezvu.

Architektura back-endu zahrnuje výběr technologií, jako jsou programovací jazyky, frameworky a databázové systémy, které jsou nezbytné pro konkrétní potřeby aplikace. Kvalitní back-end by měl být škálovatelný, což znamená, že by měl být schopen zpracovávat rostoucí množství uživatelských požadavků bez ztráty výkonu. Důležitou součástí je také zabezpečení, kde se implementují mechanismy pro ochranu dat, autentizaci a autorizaci uživatelů.

4.3.2.1 .NET Core 7

.NET Core 7 představuje očekávaný příchod nové éry vývoje softwaru, což je další významný milník v kontinuálním vývoji frameworku .NET. .NET Core 7 přináší rozšíření stávajících funkcionalit a přináší nové nástroje, které usnadní vývoj moderních, výkonných a spolehlivých webových aplikací.

```
1 app.MapPost("api/course/add", async (StudyDb db, Course course,
2     HttpContext context) =>
3     {
4         var user = db.Users.SingleOrDefault(u => u.Email == context.
5             User.FindFirstValue("preferred_username"));
6         if (user is null) return Results.NotFound();
7
8         course.DateAdded = DateTime.Now;
9         course.UserID = user.UserID;
10        course.NotificationSet = false;
11        await db.Courses.AddAsync(course);
12        await db.SaveChangesAsync();
13
14        var log = new Log();
15        log.UserID = user.UserID;
16        log.Event = LogEvent.CourseAdded;
17        log.DateAdded = DateTime.Now;
18        log.CourseID = course.CourseID;
19        log.Read = false;
20        await db.Logs.AddAsync(log);
21
22        await db.SaveChangesAsync();
23        return Results.Created($"{"/course/{course.CourseID}", course);
24    }
25 ).RequireAuthorization();
```

Zdrojový kód 6: Endpoint pro vytvoření kurzu

.NET Core 7 přináší zdokonalené nástroje pro práci s daty a databázemi, což umožní vývojářům rychleji a efektivněji pracovat s velkými objemy dat. Nové

verze knihoven a podpora moderních databázových systémů usnadní implementaci komplexních datových vrstev. S novými funkcemi a vylepšeními, které .NET Core 7 přináší, se očekává, že tento framework bude hrát klíčovou roli ve vývoji moderních webových aplikací.

V aplikaci byla použita Minimal API metoda definování endpointů, která eliminuje zbytečný kód a zjednodušuje tvorbu HTTP API nebo webových aplikací. Tato nová funkce vede k čistějšímu a srozumitelnějšímu kódu. Jedná se o novinku, která byla zavedena v .NET 6. Příklad takto definovaného endpointu můžete vidět na obrázku 6.

4.3.2.2 Odesílání emailů pomocí Outlook.com a SmtplibClient

```
1 var smtpClient = new SmtplibClient("smtp-mail.outlook.com")
2 {
3     Port = 587,
4     Credentials = new NetworkCredential("vorlka00@upol.cz", "
5         xxxxxxxxxxx"),
6     EnableSsl = true,
7 };
8 smtpClient.Send(
9     "vorlka00@upol.cz",
10    subscribedUser.Email,
11    "UPSHARE: New comment",
12    "New comment added in course " + comment.CategoryName + ".");
```

Zdrojový kód 7: Příklad konfigurace a odesílání emailů

V rámci aplikace existuje možnost nastavit notifikace pro soubory a kurzy, což umožňuje uživatelům být vždy v obraze ohledně nových událostí. Tato funkce je zvláště užitečná v situacích, kdy dochází ke klíčovým změnám, například když je přidán nový komentář k souboru nebo je nahrán nový materiál do kurzu.

Když se v aplikaci stane nějaká relevantní událost, jako je zmíněné přidání komentáře nebo nový soubor, notifikace jsou okamžitě zobrazeny uživatelům, kteří mají zapnutou tuto funkci. Tyto notifikace se diskrétně objeví v pravém horním rohu uživatelského rozhraní, což umožňuje rychlý a snadný přístup k důležitým informacím.

Pro případ, že uživatel není v daný okamžik online, byla zavedena další záložní možnost. Konkrétně se jedná o automatické odesílání notifikací prostřednictvím emailu. Tímto způsobem se zajišťuje, že uživatel nezmešká žádné důležité události, i když není připojen k aplikaci. Pro účel bakalářské práce jsem zvolila univerzitní [Simple Mail Transfer Protocol \(SMTP\)](#) Outlook server, který poskytuje spolehlivý a bezpečný způsob odesílání elektronických zpráv.

Emailový server je klíčovým prvkem infrastruktury elektronické pošty, která umožňuje uživatelům posílat a přijímat zprávy prostřednictvím internetu. Tyto

servery komunikují pomocí standardních protokolů, jako je [SMTP](#) pro odesílání a [Post Office Protocol 3 \(POP3\)](#) nebo [Internet Message Access Protocol \(IMAP\)](#) pro příjem zpráv.

Použitá `Smtplib` knihovna je napsána v programovacím jazyce `C#`. Umožňuje jednoduchou implementaci odesílání emailů pomocí protokolu [SMTP](#) a poskytuje jednoduché rozhraní pro nastavení odesílatele, příjemce, předmětu a obsahu zprávy. `Smtplib` je součástí `.NET Frameworku` a je široce využíván v `C#` aplikacích pro automatizované zasílání notifikací, ověření uživatelů a další komunikační účely. Příklad konfigurace a odesílání emailů je ilustrován ve zdrojovém kódu 7.

4.3.3 Databázová vrstva

Databázová vrstva v `.NET` aplikaci má za úkol zajistit komunikaci mezi samotnou aplikací a databázovým systémem. Tato vrstva umožňuje ukládání, získávání a aktualizaci dat v databázi prostřednictvím vhodných dotazů a transakcí. Díky oddělení databázové logiky od zbytku aplikace je možné dosáhnout lepšího řízení, efektivity a škálovatelnosti celého systému. Tím umožňuje databázová vrstva `.NET` aplikaci efektivně pracovat s daty a zajišťuje, že informace jsou uloženy a spravovány spolehlivě a bezpečně.

4.3.3.1 Entity Framework Core

```
1 using Microsoft.EntityFrameworkCore;
2
3 namespace BachelorThesis.Models
4 {
5     public class Course
6     {
7         public int CourseID { get; set; }
8         public int UserID { get; set; }
9         public string Title { get; set; }
10        public string Short { get; set; }
11        public DateTime DateAdded { get; set; }
12        public bool NotificationSet { get; set; }
13    }
14 }
```

Zdrojový kód 8: Objektový model kurzu

[Entity Framework Core \(EF Core\)](#) je open-source knihovna vyvinutá společností Microsoft, která poskytuje [Objektově Relační Mapování \(ORM\)](#) pro `.NET` aplikace. `EF Core` umožňuje vývojářům pracovat s databázemi pomocí objektového modelu jako je ve zdrojovém kódu 8, což usnadňuje a zrychluje práci s daty a zjednodušuje jejich manipulaci. Tato knihovna podporuje různé databázové systémy, včetně Microsoft SQL Server, MySQL, SQLite a další. Díky své

jednoduchosti a flexibilitě je EF Core oblíbenou volbou pro řešení datové vrstvy v moderních .NET aplikacích.

4.3.3.2 SQLite

SQLite je lehký, samostatný a serverově nezávislý relační databázový systém, který umožňuje ukládání dat ve strukturovaných tabulkách. Jeho hlavní výhody zahrnují snadnou implementaci, nízkou paměťovou náročnost a malou velikost souboru databáze. SQLite není závislý na externím serveru, což zjednodušuje nasazení a správu. Díky tomu je vhodný zejména pro jednoduché a menší projekty, kde není potřeba rozsáhlých databázových schémat.

Mezi nevýhody SQLite se řadí nižší výkon v porovnání s plnohodnotnými databázovými serverovými řešeními, zvláště při provádění náročných operací s velkým objemem dat. Dalším omezením je nedostatečná podpora pro složité mezitabulkové vztahy, což může omezit flexibilitu při modelování komplexních datových struktur.

Při migraci na pokročilejší databázové řešení je nutné v případě tradičního programování řešit celou řadu problémů. To zahrnuje vhodné upravení SQL dotazů, přizpůsobení databázového schématu a migraci dat. Důležitým krokem je provádění testů, které ověří funkčnost aplikace po provedených změnách. Průběžné zálohování dat je během celého procesu nezbytné.

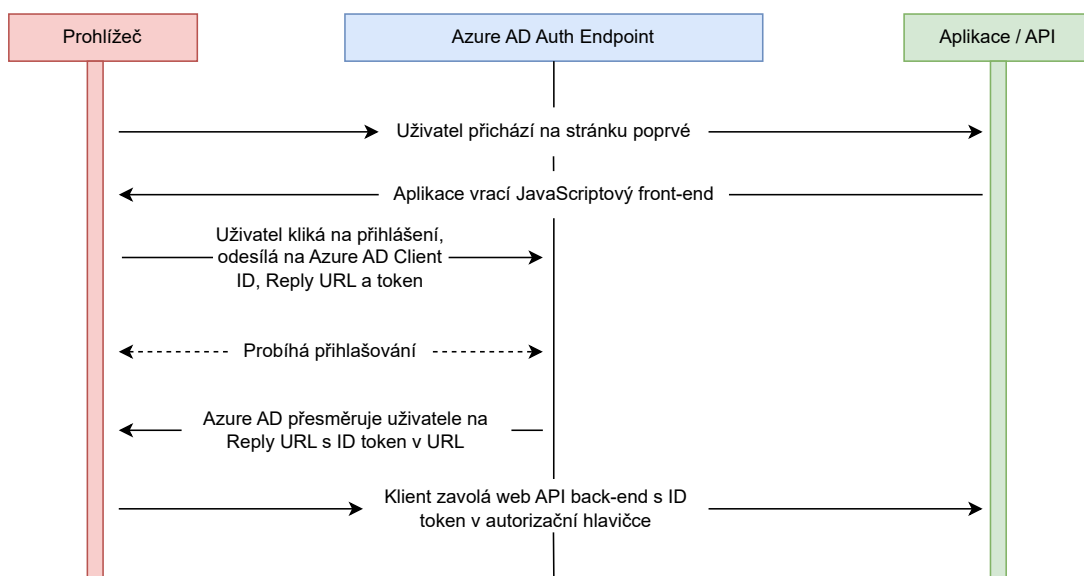
Pokud je použito [ORM](#), proces migrace je výrazně usnadněn. Při využití datových modelů je potřeba pouze zajistit kompatibilitu datových typů a struktur mezi starou a novou databází, zatímco všechny další kroky jsou zajištěny pomocí [ORM](#).

4.3.4 Integrace Azure Active Directory

[Azure Active Directory \(Azure AD\)](#) je technologie, která umožňuje uživatelům přihlásit se jednou a poté automaticky získat přístup k různým službám a aplikacím bez nutnosti opakovaného přihlašování. Způsob přihlášení je v aplikaci řešen na back-endové úrovni. Nejprve je nutné získat identifikační údaje aplikace registrací v portálu Azure. Tyto údaje jsou poté v aplikaci uloženy v souboru `appsettings.json` viz. zdrojový kód [9](#).

Proces přihlášení je velmi jednoduchý. Uživatel klikne na tlačítko přihlásit se a front-end pošle požadavek na back-end, který přesměruje uživatele na přihlašovací stránku Microsoftu. Po úspěšném přihlášení je uživatel přesměrován zpět do aplikace, do prohlížeče je uložena cookie⁶ se základními údaji a back-end si uloží aktuálně přihlášeného uživatele. Celý proces je znázorněn na obrázku [8](#). Pokud se uživatel přihlásil poprvé, nové informace jsou uloženy do databáze. Díky cookie se uživatel nemusí po určitou dobu do aplikace znovu přihlašovat. Při odhlášení je cookie smazána z prohlížeče a smažou se data o aktuálně přihlášeném uživateli.

⁶Cookie je malý textový soubor uložený na počítači uživatele, který slouží k ukládání informací o interakci s webovou stránkou a umožňuje stránce si uchovat určité údaje pro pozdější použití.



Obrázek 8: Implementovaná autentizace pomocí Microsoft Azure

4.3.4.1 UPOL Tenant

Každá organizace, která používá [Azure AD](#), má svůj vlastní tenant. Tenant obsahuje informace o uživateli, skupinách, aplikacích, politikách a dalších identitních a přístupových datech pro danou organizaci. Tenanty jsou oddělené, což znamená, že data a nastavení jedné organizace jsou izolována od ostatních, což zajišťuje bezpečnost a ochranu soukromí dat.

Každé registrované aplikaci v [Azure AD](#) lze nastavit přihlášení přes pouze jednoho tenanta nebo lze umožnit přístup všem uživatelům s Microsoft účty. Pro vývojáře může dávat smysl omezit přístup pouze pro jednu konkrétní organizaci. Tou může být univerzita nebo firma. Vše se nastavuje přímo v portálu Azure při registraci aplikace viz. obrázek 9.

4.4 Vývojové prostředí

Moderní webové aplikace v .NET získaly nový rozměr díky integraci nástrojů a technologií jako jsou GitHub Actions a produkční prostředí Azure.

Při vývoji v .NET se často využívá Visual Studio nebo JetBrains Rider, které poskytují bohaté nástroje pro vývoj a ladění kódu. Samotný kód může být verzován na platformě GitHub, kde se využívá Git pro sledování změn.

GitHub Actions umožňuje automatizovat různé části vývojového procesu, jako je sestavení, testování a nasazení do Microsoft Azure cloudu. Při každém nahrání změn do repozitáře mohou být spuštěny definované akce (workflow), které provádějí kontrolu kvality kódu, kompilaci, spouštění testů a případně nasazení do produkčního prostředí.

Pro produkční nasazení a provoz moderní webové aplikace v .NET se často využívá cloudová platforma Azure. Aplikace mohou být nasazeny na virtuálních

```

1 {
2   "AzureAd": {
3     "Instance": "https://login.microsoftonline.com/",
4     "ClientId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
5     "TenantId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
6     "ClientSecret": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
7     "ClientCertificates": [
8     ],
9     "CallbackPath": "/signin-oidc",
10    "Domain": "https://bachelorsthesis.azurewebsites.net/"
11  },
12  "DownstreamApi": {
13    "BaseUrl": "https://graph.microsoft.com/v1.0",
14    "Scopes": "https://graph.microsoft.com/.default"
15  }
16 }

```

Zdrojový kód 9: Nastavení Azure AD v souboru appsettings.json

strojích, v kontejnerech nebo jako serverless funkce. Pro zajištění bezpečnosti a správy je možné využít Azure Active Directory pro autentizaci a autorizaci uživatelů.

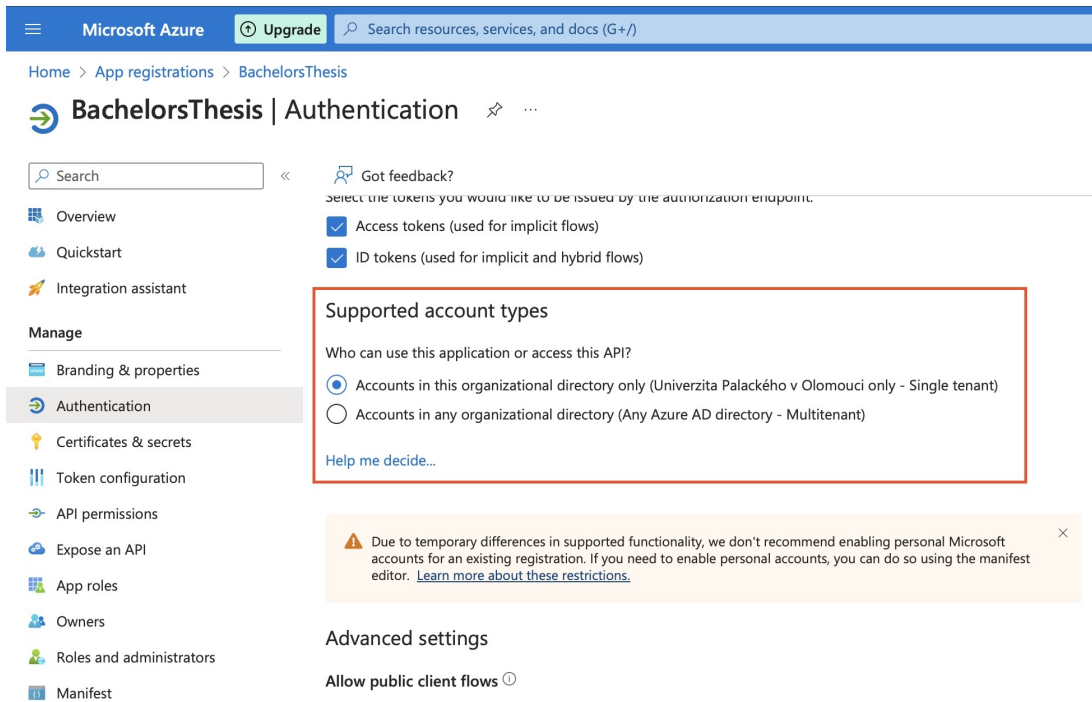
4.4.1 JetBrains Rider

JetBrains Rider^[50] je vysoce komplexní **Integrované vývojové prostředí (IDE)** pro .NET a .NET Core vývojáře. Poskytuje širokou škálu nástrojů pro efektivní vývoj, včetně inteligentního návrhu kódu, refaktorování, ladění, integrace s verzovacími systémy jako GIT a podporu pro .NET, Xamarin, Unity a další technologie. Díky intuitivnímu uživatelskému rozhraní a pokročilým funkcím usnadňuje JetBrains Rider vytváření, správu a testování aplikací. Ačkoli má placenou licenci, studenti univerzit mohou využít bezplatnou studentskou verzi.

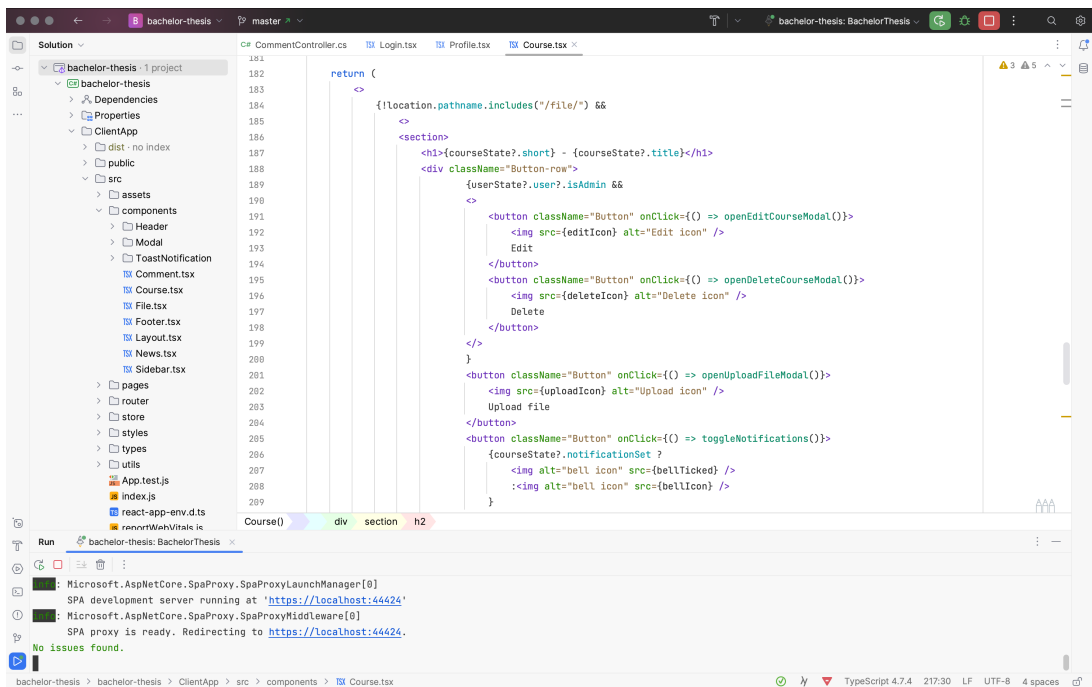
Rider není jediným dostupným prostředím pro vývoj .NET aplikací. Visual Studio od Microsoftu nabízí podobné funkce a volba mezi těmito **IDE** závisí na preferencích vývojáře.

4.5 Produkční prostředí

Produkční prostředí představuje klíčovou fázi v životním cyklu softwarové aplikace, kde se tato aplikace nasazuje do reálného provozu a stává se dostupnou pro uživatele. Je to prostředí, kde se odehrává skutečná interakce mezi aplikací a uživatelem. Kvalitní produkční prostředí by mělo být konfigurováno tak, aby splňovalo vysoké nároky na výkon, škálovatelnost a dostupnost. Zabezpečení je také důležité, a proto by měly být implementovány vhodné opatření pro ochranu dat a prevenci proti hrozbám.



Obrázek 9: Nastavení tenantů v Azure AD



Obrázek 10: Vývojové prostředí JetBrains Rider

V produkčním prostředí je třeba pečlivě zvažovat aktualizace a změny. Každá nová verze nebo aktualizace by měla být prověřena, otestována a následně nasazena s minimálním dopadem na provoz.

4.5.1 Možnosti produkčního prostředí

Nasazení .NET aplikace nabízí různé možnosti, které umožňují efektivně přenést aplikaci z vývojového prostředí do produkčního provozu. Tyto možnosti se liší v závislosti na požadavcích aplikace, typu prostředí a preferencích vývojového týmu.

Jednou z častých možností je nasazení na virtuální server nebo cloudovou platformu, jako je Microsoft Azure. S využitím služeb jako Azure App Service můžete snadno nasadit webové aplikace a API, což zahrnuje vysokou dostupnost, škálovatelnost a snadnou správu. Pro aplikace vyžadující kontejnery může být vhodné využít Kubernetes orchestraci⁷ na [Azure Kubernetes Service \(AKS\)](#).

Pro menší aplikace nebo rychlé nasazení může být zajímavou volbou serverless přístup, jako je Azure Functions. Tato metoda umožňuje spouštění kódu v reakci na události (typicky commit do master větve rezpozitáře na GitHubu) bez nutnosti spravovat infrastrukturu.

Pokud je kladen důraz na datovou vrstvu, Azure nabízí služby jako Azure SQL Database nebo Azure Cosmos DB⁸, které umožňují nasadit a spravovat databázi s vysokou dostupností a škálovatelností.

Pokud se vývojáři rozhodnou pro open-source řešení, mohou využít platformy jako Docker pro kontejnerizaci aplikace a následné hostování na vlastní infrastruktuře, nebo nasazení na jakýkoliv cloud nebo infrastrukturu podporující Docker kontejnery. V případě vlastních zdrojů, například virtuálních nebo fyzických serverů, je nutné, aby vývojář věnoval pozornost také celkové bezpečnosti celého řešení. Veřejné cloudové platformy automaticky řeší bezpečnostní aspekty, což uvolňuje vývojářský tým od těchto záležitostí a umožňuje se zaměřit výhradně na vývoj a zdokonalování aplikace a jejích funkcí.

4.5.2 Microsoft Azure App Service

Postup vytvoření nové aplikace v Microsoft Azure App Service začíná přihlášením do Azure Portálu. Poté je potřeba zvolit vytvoření nové služby *App Service*. Zde se vyplní základní informace o aplikaci, zvolí konfigurace a nastaví publikace pomocí vybraného způsobu (například Visual Studio, Git nebo FTP⁹). Pro tuto práci byla zvolena publikace GitHub repozitáře projektu pomocí GitHub Actions. Po potvrzení vytvoření je aplikace připravena a dostupná na zvolené [URL](#).

⁷Kubernetes orchestrace je proces automatického řízení a koordinace nasazení, škálování a správy kontejnerizovaných aplikací ve sdíleném prostředí.

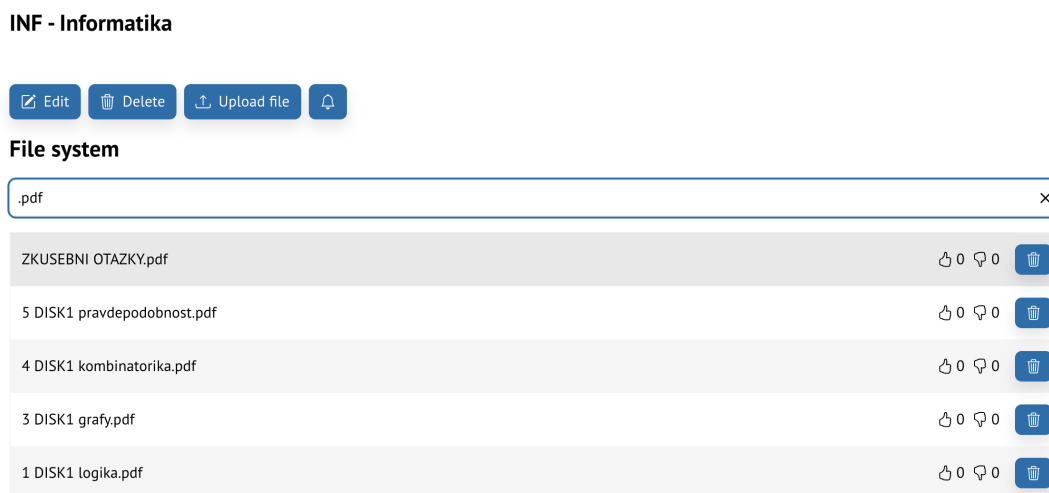
⁸Azure Cosmos DB je plně řízená databázová služba od Microsoftu, která poskytuje globálně škálovatelné a distribuované úložiště pro různorodé typy dat a aplikací.

⁹FTP (File Transfer Protocol) je zastaralý protokol používaný k přenosu souborů mezi počítači v síti, umožňující uživatelům nahrávat a stahovat soubory z vzdáleného serveru. Od FTP se postupně upouští kvůli nízké rychlosti, velké režii a nízké bezpečnosti.

4.6 Funkce aplikace

Tato část má za úkol detailněji popsat jednotlivé funkce, které jsou součástí této aplikace. Aplikace nabízí uživatelům mnoho funkcí, avšak zde budou podrobněji rozebrány a vysvětleny pouze ty nejvýznamnější.

4.6.1 Procházení souborů



Obrázek 11: Procházení souborů v kurzu

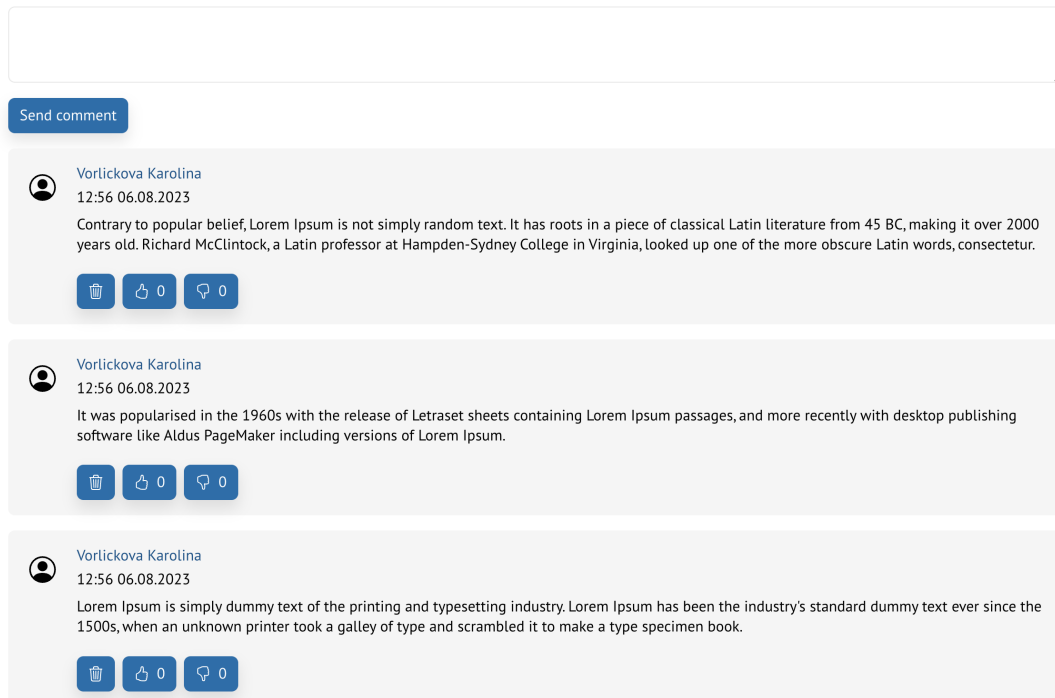
Aplikace nabízí pohodlné procházení a vyhledávání souborů. Jakmile uživatel klikne na konkrétní kurz, budou mu zobrazeny soubory patřící k danému kurzu. Mezi těmito soubory lze následně provádět rychlé vyhledávání. Tato vyhledávací funkce je integrována přímo na straně uživatelského rozhraní, což zajišťuje okamžitou odezvu na vstup uživatele. Záměrně je omezené vyhledávání pouze na rámec daného kurzu, namísto tzv. fulltextového vyhledávání¹⁰. Tento přístup umožňuje efektivní filtrování obsahu a zrychluje proces hledání požadovaných souborů. Pokud je přihlášený uživatel administrátor nebo autor souboru, aplikace mu umožní jej smazat. Ostatní uživatelé mohou soubor pouze stahovat, komentovat a nebo na soubor reagovat.

4.6.2 Komentování

Aplikace nabízí uživatelům možnost vytvářet interaktivní komentáře, které mohou být přiřazeny k jednotlivým souborům nebo k celým kurzům. V obou variantách je zajištěno, že na každý komentář lze reagovat formou dalších odpovědí a také udělit like nebo dislike jako projev souhlasu či nesouhlasu. Komentáře podléhají přísným pravidlům pro délku textu, omezenou na maximálně 400 znaků.

¹⁰Fulltextové vyhledávání je technika umožňující snadno a rychle nalézt veškerý obsah uložený v systému, ať už se jedná o textové dokumenty, zprávy nebo jiné informace.

Comment - max. 400 characters



Obrázek 12: Komentáře

Tato omezení jsou účelná, aby se zabezpečilo, že diskuse v rámci komentářů zůstává přehledná a konkrétní.

Standardně je na stránce zobrazeno 5 nejnovějších komentářů, což zajišťuje rychlý přehled o aktuálních reakcích uživatelů. V případě, že v kurzu nebo u souboru existuje více komentářů, je uživatelům prezentováno tlačítko umožňující načíst dalších 5 komentářů. Tato funkce je navržena tak, aby umožňovala postupné načítání komentářů bez zbytečného přetěžování stránky. Tento proces pokračuje automaticky, dokud jsou k dispozici další komentáře k zobrazení.

4.6.3 Reakce uživatelů

Jednou z dalších významných funkcí je možnost uživatelů vyjádřit svůj názor na konkrétní soubor nebo komentář prostřednictvím reakcí. Tyto reakce slouží jako prostředek pro projevení souhlasu nebo nesouhlasu. Mechanismus reakcí je koncipován tak, že uživatel má možnost přiřadit jednu reakci (like nebo dislike) k jedinému souboru či komentáři. Tuto reakci může libovolně měnit v závislosti na svém aktuálním názoru. Pokud se uživatel rozhodne zcela zrušit svou reakci a nechat soubor nebo komentář bez ní, stačí znovu kliknout na původní reakci, čímž dojde k jejímu zrušení.

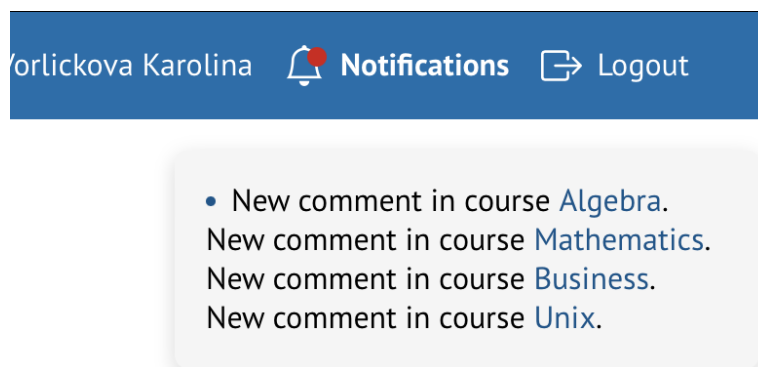
Pro každou reakci je přehledně zobrazeno číslo, které reprezentuje celkový počet reakcí na daný soubor nebo komentář. Toto číslo se dynamicky upravuje v závislosti na akcích ostatních uživatelů, čímž je zajištěn průběžný přehled o

images.jpg	👍 0 👎 0	🗑️
pr1.zip	👍 2 👎 1	🗑️
pr1.pdf	👍 1 👎 0	🗑️
pr1.m	👍 0 👎 0	🗑️
Obrázek1.png	👍 0 👎 1	🗑️

Obrázek 13: Reakce uživatelů

celkovém ohlasu na daný obsah.

4.6.4 Notifikace



Obrázek 14: Notifikace

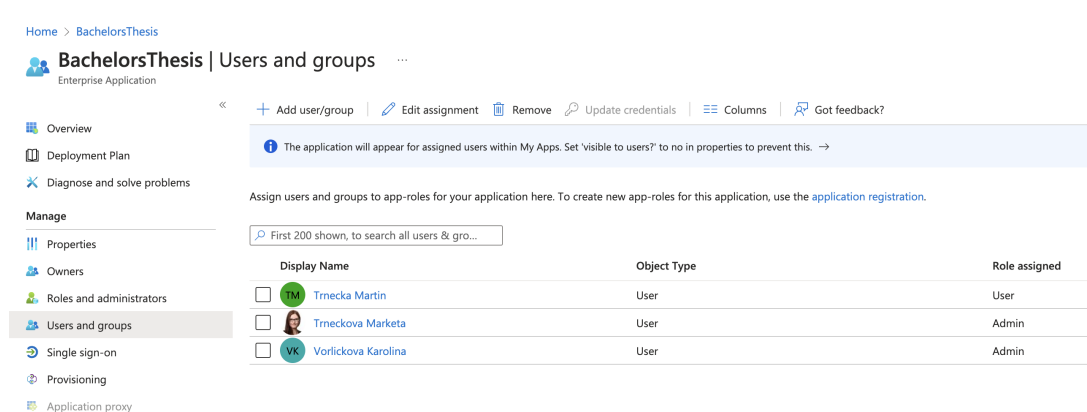
Další cennou funkcionalitou aplikace je schopnost aktivovat oznámení. Při navrhování tohoto prvku bylo nezbytné provést analýzu způsobu, jakým se bude aplikace využívat, a identifikovat důležité události, jež je vhodné uživatelům sdělovat. Oznámení byla pečlivě vybrána tak, aby informovala o zásadních změnách, aniž by vytvářela nepříjemný SPAM¹¹ s nepodstatnými informacemi.

Funkce oznámení lze aktivovat pro jednotlivé soubory nebo celé kurzy. Oznámení se zobrazuje v horním pravém rohu obrazovky a zároveň je uživateli, který tuto možnost zapnul, zasláno upozornění emailem. V případě souborů bude uživatel informován o novém komentáři přidaném k danému souboru. U kurzů jsou oznámení spojena s dvěma typy událostí. První zahrnuje přidání komentáře ke

¹¹SPAM je nevyžádaná a často nadměrná distribuce nebo zasílání nežádoucích a často irrelevantních zpráv.

kurzu a druhý typ události signalizuje nahrání nového souboru do kurzu. Záměrně nejsou posílána oznámení o přidání komentáře k souboru, pokud si uživatel zapnul notifikace pro celý kurz. Možnost nastavení oznámení je flexibilní a uživatelé si mohou své preference kdykoliv upravit.

4.6.5 Uživatelské role



Obrázek 15: Uživatelské role

Tato aplikace byla konstruována s ohledem na budoucí možnosti rozšíření uživatelských rolí. Pro správu rolí aplikace využívá Azure AD, což umožňuje po úspěšném přihlášení získat informace o uživateli včetně jeho přiřazené role. Koncept Azure AD staví na rovnosti všech uživatelů, dokud správce tenanta nepřihlásí konkrétní uživateli konkrétní role. Základní role, jako je *User*, jsou definovány automaticky, ačkoli lze přidat neomezený počet dalších rolí, jako je *Administrator*, *Manager*, *Moderator*, atd. V současné verzi aplikace je ovšem implementována pouze druhá role, *Administrator*. Možnost změny uživatelských rolí je dostupná pouze správci tenanta v sekci *App Roles* v rámci *App Registrations*.

Pokud se uživatel přihlásí s rolí *Administrator*, jsou mu nabídnuty rozšířené možnosti, včetně mazání souborů a komentářů, a také přidávání a mazání kurzů.

Uživatelské role aplikace získává až po přihlášení od Azure AD, a tudíž je nutné, aby se uživatel odhlásil a opět přihlásil, pokud dojde ke změně jeho role.

5 Budoucí rozšíření

Tato kapitola se zabývá možnými budoucími rozšířeními aplikace, které zahrnují několik klíčových prvků, které by mohly výrazně zlepšit uživatelskou zkušenost a rozšířit funkčnost. Tyto inovace představují směr, kterým by se mohla aplikace vydat, aby lépe sloužila potřebám uživatelů:

- **Rozšířený uživatelský chat** Implementace interaktivního chatu mezi uživateli by přinesla nový rozměr komunikace na platformě. Uživatelé by mohli snadno sdílet nápady, řešit otázky a vzájemně se podporovat při učení. Tato funkce by mohla být integrována s existujícími diskusními fóry a umožnila by rychlou a přímou výměnu informací.
- **Mobilní aplikace** Vytvoření mobilní aplikace by umožnilo uživatelům přístup ke kurzům a zdrojům kdekoli a kdykoli. Mobilní aplikace by mohla nabídnout přizpůsobitelné rozhraní a funkce optimalizované pro mobilní zařízení, což by zvýšilo pohodlí uživatelů při učení na cestách.
- **Rozšířená offline podpora prostřednictvím PWA** Integrace **PWA** by umožnila uživatelům pokračovat ve studiu i v situacích, kdy není dostupné internetové připojení. Nabídka offline přístupu k vybraným materiálům a funkcím by výrazně zvýšila flexibilitu a dostupnost naší platformy.
- **Nastavení oblíbených kurzů v sidebaru** Personalizace zobrazených a oblíbených kurzů v bočním panelu by umožnila uživatelům rychlejší a pohodlnější navigaci. Uživatelé by si mohli jednoduše vybrat kurzy, které je zajímají nebo na které se chtějí zaměřit, a měli by k nim snadný přístup z hlavního rozhraní.

Tato budoucí rozšíření by mohla výrazně přispět k rozvoji a zdokonalení celé aplikace. Uživatelé by měli k dispozici více možností, jak se zapojit, učit se a využívat služeb aplikace dle svých individuálních potřeb a preferencí.

6 Závěr

Tato bakalářská práce si kladla za cíl vytvořit inovativní webovou platformu, která by změnila způsob sdílení studijních materiálů. Hlavním úkolem bylo vytvořit moderní a efektivní nástroj pro studenty a učitele, usnadňující jim snadný přístup a interakci se vzdělávacími obsahy.

Během analýzy byly identifikovány nedostatky současných systémů a platform ve vzdělávacím prostředí, zejména v oblasti sdílení souborů a spolupráce mezi studenty. Tato situace byla pohnutkou k tomu, aby autorka této práce vytvořila webovou aplikaci, která by tyto problémy řešila a nabídla moderní řešení pro efektivní správu a sdílení studijních materiálů.

Systematický postup byl zaveden při návrhu a vývoji aplikace. To zahrnovalo analýzu požadavků, konceptuální návrh efektivního řešení a výběr optimálních technologií pro implementaci. Výsledkem byla uživatelsky přívětivá webová platforma, umožňující snadné nahrávání, organizaci a vyhledávání souborů v rámci kurzů. Důraz byl kladen i na zabezpečení, kde byla implementována moderní metoda **SSO** pro přihlašování pomocí Microsoft Azure Active Directory.

Význam této práce nebyl omezen pouze na teoretický a praktický přínos. Tato práce byla velice přínosná i pro autorku samotnou. Během vývoje se naučila mnoho nových moderních webových technologií a vývojových procesů, což jí poskytlo pevný základ pro její budoucí profesní dráhu. Nejvýznamnějším důsledkem bylo, že tato bakalářská práce byla pro autorku důvodem k napsání její první publikace, na kterou je náležitě hrdá.

Závěrem, tato bakalářská práce nejen přinesla malý příspěvek k modernizaci vzdělávacích platform a digitalizaci univerzitního prostředí, ale také výrazně obohatila profesní dovednosti autorky.

Conclusions

This bachelor thesis aimed to create an innovative web platform that would change the way learning materials are shared. The main task was to create a modern and effective tool for students and teachers, facilitating them to easily access and interact with educational content.

During the analysis, the shortcomings of current systems and platforms in the educational environment were identified, especially in the area of file sharing and collaboration between students. This situation motivated the author of this thesis to develop a web application that would address these issues and offer a modern solution for effective management and sharing of learning materials.

A systematic process was implemented in the design and development of the application. This included requirements analysis, conceptual design of an effective solution and selection of optimal technologies for implementation. The result was a user-friendly web-based platform that allows easy uploading, organization and searching of files within courses. The focus was also on security, where a modern [SSO](#) method for logging in using Microsoft Azure Active Directory was implemented.

The importance of this work was not limited to theoretical and practical contributions. This work was also very beneficial for the author herself. During the development process, she learned many new modern web technologies and development processes, which provided her with a solid foundation for her future career. The most significant consequence was that this undergraduate thesis was the reason for the author to write her first publication, of which she is duly proud.

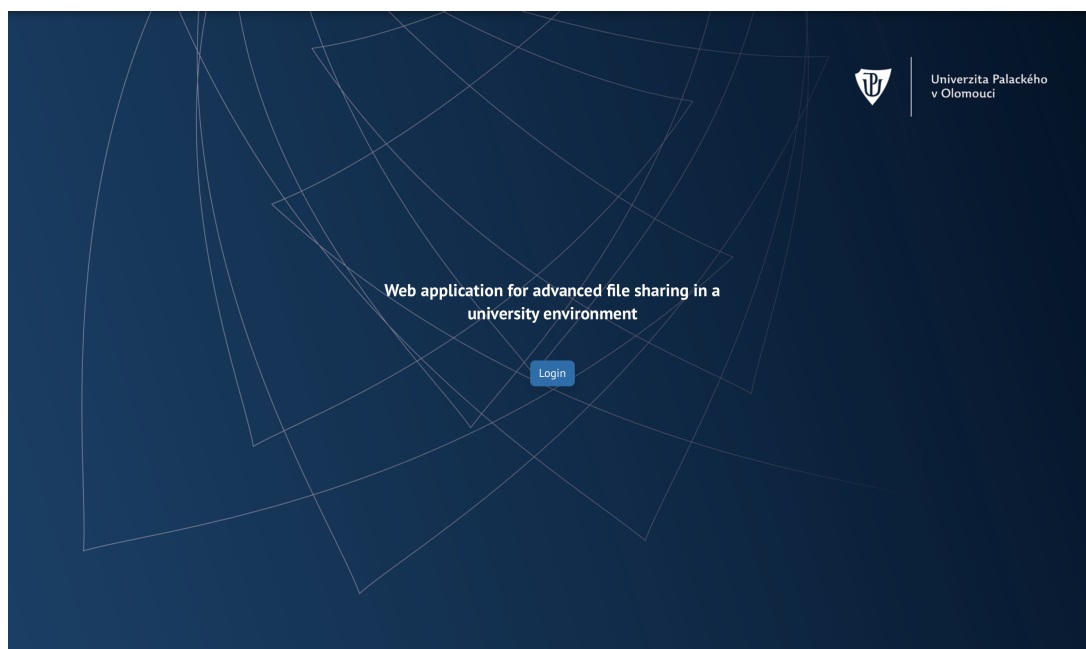
In conclusion, this bachelor thesis not only made a small contribution to the modernisation of educational platforms and the digitalisation of the university environment, but also greatly enriched the author's professional skills.

A Uživatelská příručka

Tato uživatelská příručka byla sestavena s cílem poskytnout užitečné informace a návody pro efektivní využívání této webové aplikace. Obsahuje detailní postupy pro procesy nahrávání, stahování a správy souborů, spolu s podrobnými vysvětleními klíčových funkcí, které tato aplikace nabízí.

A.1 Přihlašovací obrazovka

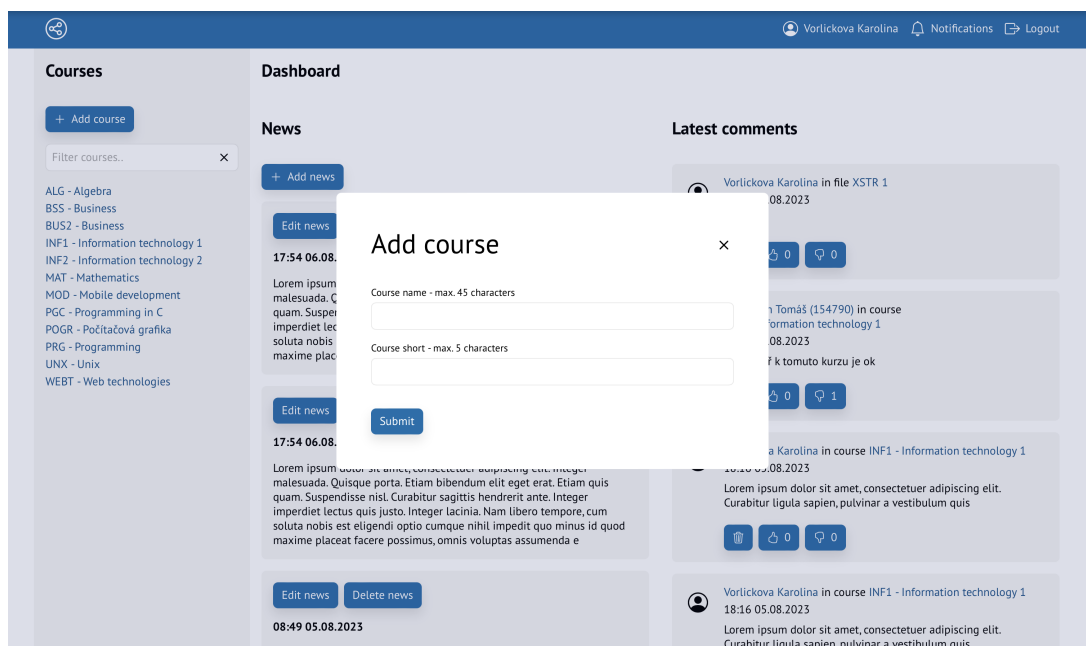
Přihlašovací obrazovka představuje vstupní bránu do aplikace, kde je k dispozici pouze přihlašovací tlačítko. Po stisknutí tohoto tlačítka je uživatel přesměrován na přihlašovací stránku Microsoftu, kde může dokončit svůj přihlašovací proces. Jako bezpečnostní opatření se tato obrazovka zobrazí také v případě, že nepřihlášený uživatel zadá neplatnou URL adresu aplikace, čímž je zajištěno, že pouze oprávnění uživatelé budou moci přistupovat k obsahu.



Obrázek 16: Přihlašovací obrazovka

A.2 Rozložení stránek

Každá stránka, která je viditelná pro přihlášené uživatele, byla uspořádána do několika základních částí. Hlavička obsahuje několik důležitých prvků: klikatelné logo, které vás přesměruje na nástěnku, odkaz na váš uživatelský profil, tlačítko notifikací, které po kliknutí otevře okno s aktuálními upozorněními, a tlačítko pro odhlášení. Boční menu nabízí seznam všech dostupných kurzů a umožňuje jednoduše provádět filtraci. Pro administrátory aplikace je zde také tlačítko,



Obrázek 17: Modální okno

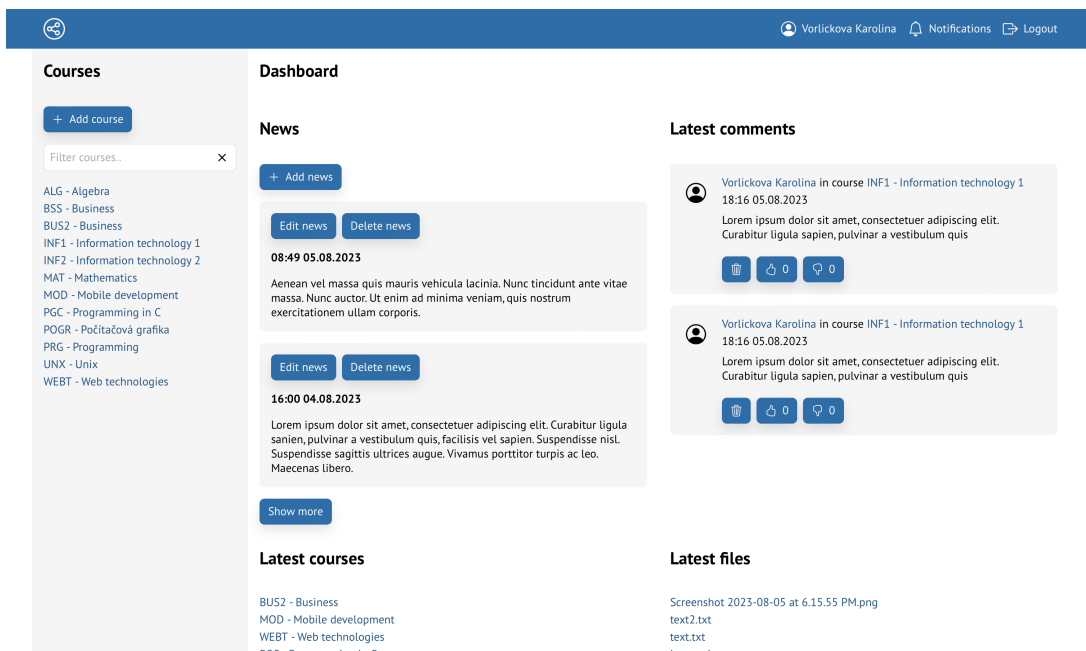
které spouští modální okno s formulářem pro přidání nového kurzu. V patičce stránky jsou k dispozici kontaktní informace na autora této aplikace, což může být užitečné pro další dotazy a zpětnou vazbu.

A.3 Nástěnka

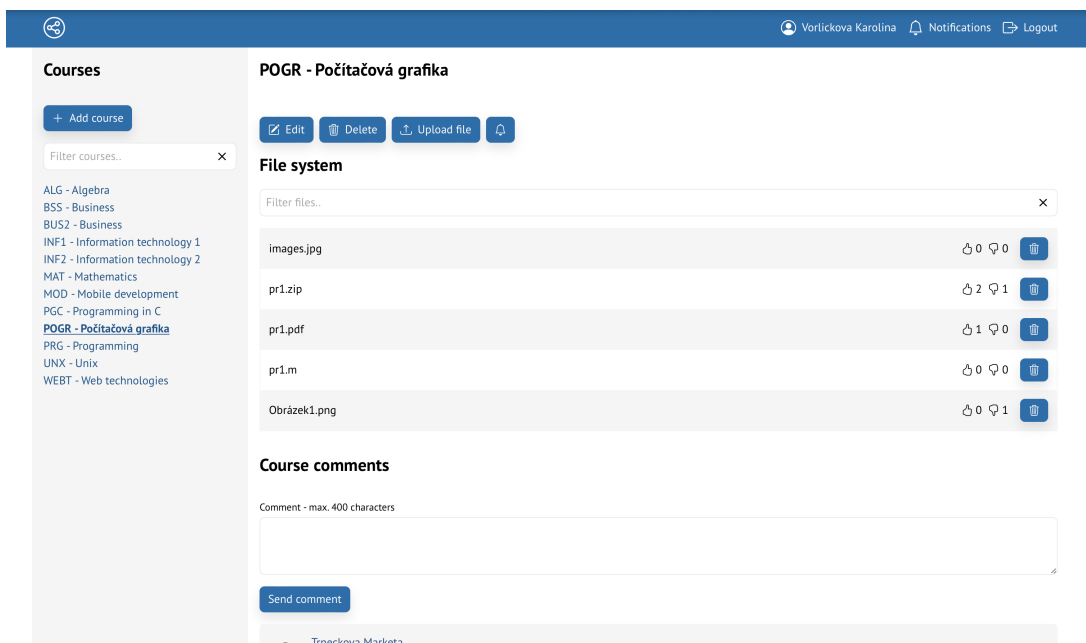
Nástěnka představuje centrální bod celé aplikace a tvoří první stránku, na níž uživatelé přistupují po přihlášení. Tato stránka je rozdělena do čtyř klíčových sekcí, které nejen zobrazují aktuální dění v aplikaci, ale také fungují jako praktické navigační prvky. Tyto čtyři sekce jsou novinky, nejnovější komentáře, dostupné kurzy a soubory. Tyto oddíly vám umožňují snadno a rychle získat přehled o nejnovějších událostech a obsahu v aplikaci a slouží jako výchozí bod, který usnadňuje orientaci a začátek práce.

A.4 Detail kurzu

Na detail kurzu se dostaneme kliknutím na požadovaný kurz v bočním menu. Pod názvem kurzu najdeme řádek s akčními tlačítky. Možnosti smazat a editovat kurz jsou dostupné pouze administrátorovi. Tlačítko upload file otevře modálové okno s polem pro nahrání souboru a textovým polem pro popis. Poslední tlačítko slouží k nastavení notifikace na nové soubory a komentáře. V tabulce souborů pod tlačítky vidíme kromě názvu souborů i počet reakcí a mazací tlačítko, pokud se jedná o soubor nahraný přihlášeným uživatelem. V tabulce lze vyhledávat pomocí textového pole nad ní. Poslední část stránky je komentářová sekce, kde se zobrazuje posledních 5 komentářů a tlačítko pro zobrazení dalších.

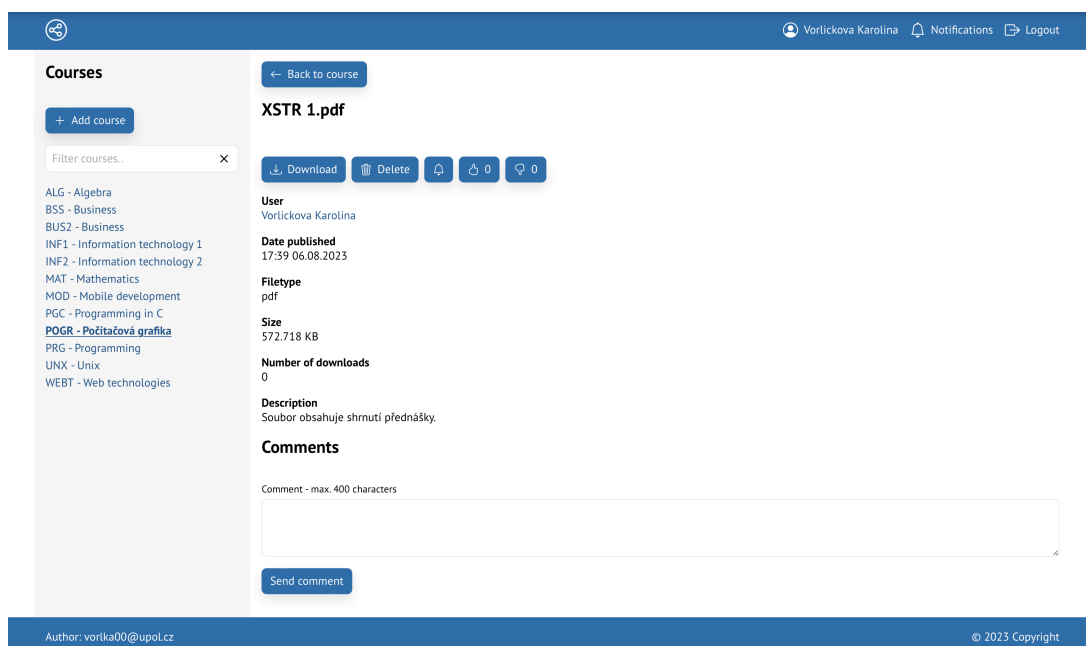


Obrázek 18: Nástěnka



Obrázek 19: Detail kurzu

A.5 Detail souboru



Obrázek 20: Detail souboru

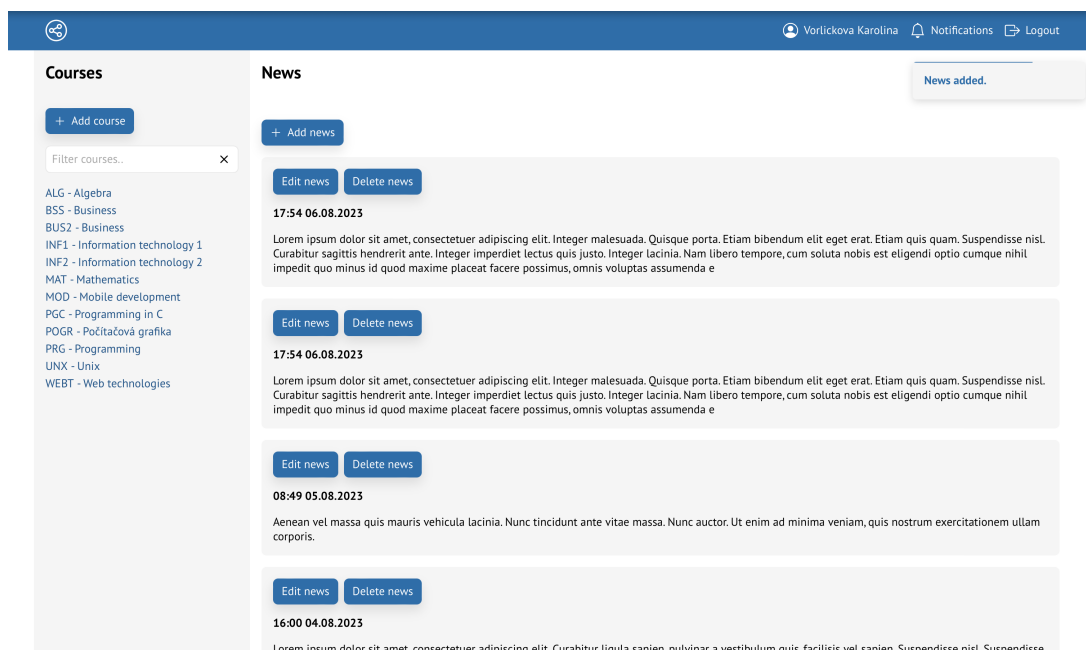
Na detail souboru se můžeme dostat kliknutím na soubor v tabulce na detailu kurzu. Stránka obsahuje všechny důležité informace o souboru jako je název, datum přidání, počet stažení, autor, velikost, typ a popis. Pod nadpisem opět najdeme řádek s akčními tlačítky jako na detailu kurzu. Běžnému uživateli je umožněno stahovat soubor, reagovat likem nebo dislikem a nastavit notifikaci, administrátor a autor souboru může soubor smazat. Pod informacemi se nachází sekce komentářů.

A.6 Sekce komentářů

Tato část se nachází na stránkách s detaily a slouží k usnadnění komunikace a sdílení názorů mezi uživateli ohledně daného kurzu nebo souboru. Každý komentář zahrnuje informace o autorovi, čas přidání a samotný text komentáře, který nelze po odeslání upravovat. Ke smazání mají oprávnění pouze autoři komentářů a administrátoři. Uživatelé mají možnost vyjádřit svůj postoj ke komentářům pomocí liků a disliků. Počet komentářů načtených na stránce je výchozím nastavením omezen na pět, další komentáře je možné načíst kliknutím na tlačítko *Více*.

A.7 Novinky

Tato stránka poskytuje přehled všech aktuálních novinek, které jsou řazené podle data vydání, s nejnovějšími zprávami nahoře. Přístup na tuto stránku je dostupný



Obrázek 21: Novinky

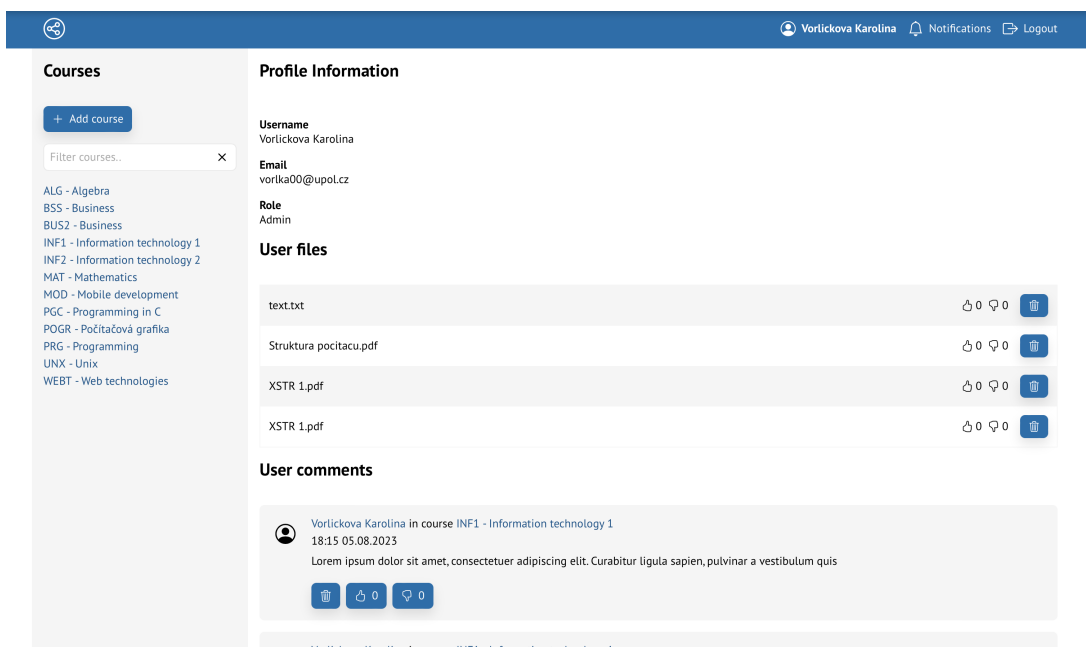
přímo z uživatelské nástěnky, stačí kliknout na tlačítko “Zobrazit více“ umístěné pod sekcí s novinkami. Pro správce nebo administrátora je navíc k dispozici možnost přidávat nová oznámení. Toto lze provést pomocí speciálního tlačítka, které otevře modální okno s textovým polem, kam lze zapsat obsah nového oznámení.

A.8 Profil

Profil každého uživatele poskytuje přehled o jeho osobních informacích, jako je e-mailová adresa, jméno a přiřazená role. Pod těmito informacemi se nachází seznam komentářů a souborů, které byly vytvořeny tímto přihlášeným uživatelem. V rámci tohoto seznamu jsou uvedeny odkazy, které umožňují přímý přístup k původnímu umístění daného komentáře nebo souboru. V profilu má uživatel také možnost tyto záznamy smazat, což usnadňuje správu vlastních aktivit a obsahu.

A.9 Detail uživatele

Detail uživatele je strukturálně podobný uživatelskému profilu, avšak liší se v tom, že nezahrnuje sekce s komentáři a soubory. Tuto stránku je možné zobrazit kliknutím na jméno autora komentáře nebo na jméno autora souboru, což uživatelům umožňuje jednoduše se dostat k podrobnostem o konkrétním uživateli.

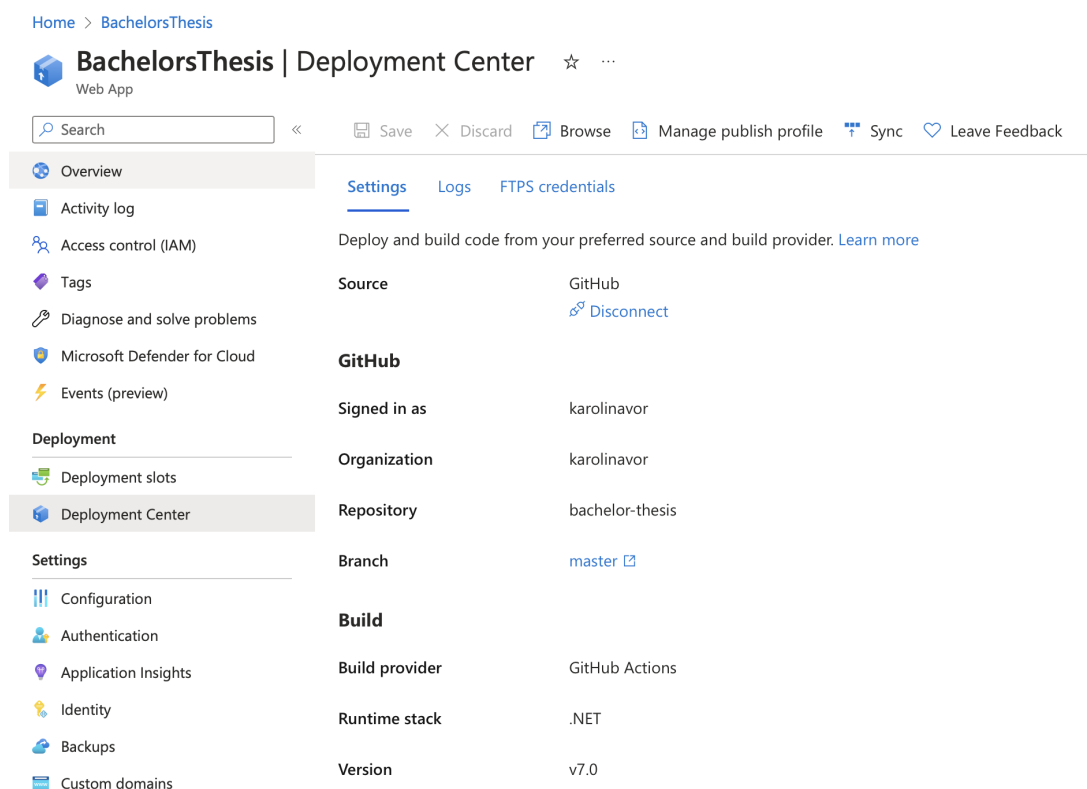


Obrázek 22: Profil

B Implementační manuál

Tento návod je vytvořen jako přehledný a srozumitelný průvodce procesem instalace a spuštění aplikace v lokálním prostředí, stejně jako v cloudovém prostředí. Pro případ lokálního spuštění jsou instrukce připraveny pro operační systémy Microsoft Windows a MacOS, aby vám poskytly přesné a praktické kroky k úspěšnému nastavení a implementaci.

B.1 Nasazení do Azure Web Cloud



Obrázek 23: Nastavení deploymentu v Azure AD

Pro nasazení na cloud je nutný Microsoft účet. Nastavení probíhá v Azure portálu, kde je nutné projít procesem vytvoření a nastavení služby. Nejprve je potřeba přejít do portálu Azure.

`https://aure.microsoft.com`

Následně vybereme App Services, klikneme na tlačítko Create. Následně vyplníme tyto informace:

- **Name** - Dle uvážení
- **Publish** - Způsob nahrání, vybereme Code

- **Runtime stack** - .NET 7 (STS)
- **Region** - West Europe
- **Operation System** - Windows

V záložce **Deployment** vybereme následující možnosti:

- **Github Actions Settings - Continuous Deployment** - Enable
- **Add Account** - po kliknutí
- **Organization** - GitHub účet
- **Repository** - vybereme příslušný repozitář
- **Branch** - vybereme příslušný branch

Následně klikneme na tlačítko *Create*, čímž bude celý proces dokončen a aplikace bude úspěšně spuštěna v Azure Cloudu. Po vytvoření se [URL](#) adresa nově vytvořeného webu zobrazí v administraci a bude možné se prokliknout na nasazenou webovou aplikaci.

B.2 Implementace na OS Windows

Pro zajištění úspěšného spuštění, je důležité nejprve nainstalovat potřebné technologie. K tomu je třeba provést instalaci softwaru Rider nebo Visual Studio, který vám umožní efektivně pracovat s aplikací. Tyto nástroje poskytují prostředí pro vývoj a testování a jsou nezbytné pro správnou funkčnost aplikace.

<https://www.jetbrains.com/rider/>
<https://visualstudio.microsoft.com/>

Aby bylo možné aplikaci úspěšně sestavit, je nezbytné nainstalovat Node.js a npm (Node Package Manager). Tyto klíčové nástroje umožní efektivní správu závislostí a provádění skriptů.

<https://nodejs.org/en/download>

Aplikace k samotnému běhu vyžaduje .NET Core 7 SDK. Tato vývojová sada poskytuje nezbytné nástroje a knihovny pro vytváření, kompilaci a spuštění aplikace.

<https://dotnet.microsoft.com/en-us/download/dotnet/7.0>

Pro práci s daty a databázemi je nutné nainstalovat .NET Entity Framework. Tato knihovna umožní snadnou a efektivní manipulaci s daty ve vaší aplikaci.

```
dotnet tool install --global dotnet-ef
```

Pokud se Vám nedaří nainstalovat EF, zkuste následující příkaz:

```
dotnet nuget add source  
https://api.nuget.org/v3/index.json -n nuget.or
```

Pro přípravu aplikace ke spuštění je nezbytné sestavit celé řešení. To zahrnuje kompilaci zdrojového kódu, správu závislostí a přípravu aplikace k spuštění. Otevřete .sln soubor v Rideru/Visual Studiu a aplikaci sestavte kliknutím na tlačítko Build. Aplikace automaticky stáhne front-endové i back-endové balíčky.

Pro vytvoření databáze použijte následující příkazy v terminálovém okně. Ujistěte se, že se nacházíte v kořenovém adresáři projektu, a teprve poté spusťte následující příkazy.

```
dotnet ef migrations add InitialCreate  
dotnet ef database update
```

Pro spuštění aplikace v vývojovém prostředí stačí jednoduše kliknout na zelené tlačítko s trojúhelníkem. Tímto způsobem se aplikace spustí a automaticky otevře v internetovém prohlížeči na portu 44424.

B.3 Implementace na MacOS

Pro zajištění úspěšného spuštění, je důležité nejprve nainstalovat potřebné technologie. K tomu je třeba provést instalaci softwaru Rider nebo Visual Studio, který vám umožní efektivně pracovat s aplikací. Tyto nástroje poskytují prostředí pro vývoj a testování a jsou nezbytné pro správnou funkčnost aplikace.

```
https://www.jetbrains.com/rider/  
https://visualstudio.microsoft.com/
```

Aby bylo možné aplikaci úspěšně sestavit, je nezbytné nainstalovat Node.js a npm (Node Package Manager). Tyto klíčové nástroje umožní efektivní správu závislostí a provádění skriptů.

```
https://nodejs.org/en/download
```

Aplikace k samotnému běhu vyžaduje .NET Core 7 SDK. Tato vývojová sada poskytuje nezbytné nástroje a knihovny pro vytváření, kompilaci a spuštění aplikace.

```
https://dotnet.microsoft.com/en-us/download/dotnet/7.0
```

Pokud používáte počítač Mac s architekturou procesoru ARM64 (Apple Silicon), zahajte postup následujícím příkazem:

```
sudo ln -s /usr/local/share/dotnet/x64/dotnet /usr/local/bin/
```

Pro práci s daty a databázemi je nutné nainstalovat .NET Entity Framework. Tato knihovna umožní snadnou a efektivní manipulaci s daty ve vaší aplikaci.

```
dotnet tool install dotnet-ef --global -a arm64g
```

Pro připravení aplikace ke spuštění je nezbytné sestavit celé řešení. To zahrnuje kompilaci zdrojového kódu, správu závislostí a přípravu aplikace k spuštění. Otevřete .sln soubor v Rideru/Visual Studiu a aplikaci sestavte kliknutím na tlačítko *Build*. Aplikace automaticky stáhne front-endové i back-endové balíčky.

Pro vytvoření databáze použijte následující příkazy v terminálovém okně. Ujistěte se, že se nacházíte v kořenovém adresáři projektu, a teprve poté spusťte následující příkazy.

```
dotnet ef migrations add InitialCreate  
dotnet ef database update
```

Pro spuštění aplikace vývojovém prostředí stačí jednoduše kliknout na zelené tlačítko s trojúhelníkem. Tímto způsobem se aplikace spustí a automaticky otevře v internetovém prohlížeči na portu 44424.

C Obsah odevzdané přílohy

BachelorsThesisPdf/

Text práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech příloh.

ClientApp/

Adresář obsahující front-endovou část aplikace. Obsahem jsou soubory se styly, Reactové komponenty a stránky, nastavení React Routeru, média, Redux soubory a další pomocné funkce.

Controllers/

Adresář obsahující C# soubory, ve kterých jsou definované API endpointy a jejich logika.

Database/

Adresář obsahující nastavení vytvoření databáze.

Models/

Adresář obsahující modely tříd, které se využívají při práci s daty na back-endu a při vytváření databáze.

Properties/

Nastavení spuštění projektu ve vývojovém prostředí.

readme.md a readme.txt

Instrukce pro lokální spuštění webové aplikace v prohlížeči, včetně všech požadavků na instalaci technologií pro její bezproblémový provoz, a webová adresa, na které je aplikace nasazena pro účel testování při tvorbě posudků práce a pro účel obhajoby práce.

Seznam zkratek

AD Active Directory

AKS Azure Kubernetes Service

AWS Amazon Web Services

Azure AD Azure Active Directory

CD Continuous Deployment

CI Continuous Integration

CSS Cascading Style Sheets

DOM Document Object Model

EF Core Entity Framework Core

IDE integrované vývojové prostředí

IMAP Internet Message Access Protocol

ORM Objektově Relační Mapování

PHP PHP: Hypertext Preprocessor

POP3 Post Office Protocol 3

PWA Progresivní Webová Aplikace

SASS Syntactically Awesome Style Sheets

SEO Search Engine Optimalization

SMS Short Message Service

SMTP Simple Mail Transfer Protocol

SPA Single Page Application

SSO Single Site On

URL Uniform Resource Locator

Literatura

- [1] Gamage, Sithara HPW; Ayres, Jennifer R; Behrend, Monica B. A systematic review on trends in using Moodle for teaching and learning. *International Journal of STEM Education*. 2022, roč. 9, č. 1 s. 1–24.
- [2] *Moodle - Open-source learning platform / Moodle.org* — *moodle.org*. [cit. 9. 7. 2023]. Dostupný z: <https://moodle.org/>.
- [3] *ownCloud - share files and folders, easy and secure* — *owncloud.com*. [cit. 9. 7. 2023]. Dostupný z: <https://owncloud.com>.
- [4] *Nextcloud - Online collaboration platform* — *nextcloud.com*. [cit. 9. 7. 2023]. Dostupný z: <https://nextcloud.com>.
- [5] *Primát.cz. Usnadníme ti studium – Primát.cz* — *primat.cz*. [cit. 9. 7. 2023]. Dostupný z: <https://www.primat.cz>.
- [6] *Fituška - Fórum studentů FIT VUT v Brně* — *fituska.eu*. [cit. 9. 7. 2023]. Dostupný z: <https://fituska.eu/>.
- [7] *Fektuška.cz - Fórum studentů Fakulty elektrotechniky a komunikačních technologií VUT v Brně*. [cit. 9. 7. 2023]. Dostupný z: <https://www.fektuska.cz>.
- [8] *Discord / Your Place to Talk and Hang Out* — *discord.com*. [cit. 9. 7. 2023]. Dostupný z: <https://discord.com>.
- [9] *ownCloud - share files and folders*. [cit. 9. 7. 2023]. Dostupný z: <https://owncloud.com>.
- [10] *Nextcloud - Online collaboration platform*. [cit. 9. 7. 2023]. Dostupný z: <https://nextcloud.com>.
- [11] *Big changes: I am leaving ownCloud, Inc. today*. [cit. 9. 7. 2023]. Dostupný z: <https://karlitschek.de/2016/04/big-changes-i-am-leaving-owncloud-inc-today/>.
- [12] *CESNET / CESNET, zájmové sdružení právnických osob*. [cit. 9. 7. 2023]. Dostupný z: <https://www.cesnet.cz>.
- [13] *ownCloud@CESNET*. [cit. 9. 7. 2023]. Dostupný z: <https://owncloud.cesnet.cz>.
- [14] *Pydio User Interface* — *pydio.com*. [cit. 9. 7. 2023]. Dostupný z: https://pydio.com/sites/default/files/styles/tinypng_optimisation/public/section/01_1.png.
- [15] *Pydio* — *pydio.com*. [cit. 9. 7. 2023]. Dostupný z: <https://pydio.com>.
- [16] *Pydio Github* — *github.com*. [cit. 9. 7. 2023]. Dostupný z: <https://github.com/pydio>.
- [17] *Docker: Accelerated Container Application Development*. [cit. 9. 7. 2023]. Dostupný z: <https://www.docker.com>.

- [18] *Home | FEKT VUT — fekt.vut.cz*. [cit. 9. 7. 2023]. Dostupný z: [⟨https://www.fekt.vut.cz⟩](https://www.fekt.vut.cz).
- [19] *Faculty of Information Technology — fit.vut.cz*. [cit. 9. 7. 2023]. Dostupný z: [⟨https://www.fit.vut.cz⟩](https://www.fit.vut.cz).
- [20] *PhpBB - Free and Open Source Forum Software — phpbb.com*. [cit. 9. 7. 2023]. Dostupný z: [⟨https://www.growjob.com⟩](https://www.growjob.com).
- [21] *GrowJOB Institute - Růst bez náhody — growjob.com*. [cit. 9. 7. 2023]. Dostupný z: [⟨https://www.phpbb.com⟩](https://www.phpbb.com).
- [22] *Wayback Machine - web.archive.org*. [cit. 9. 7. 2023]. Dostupný z: [⟨https://web.archive.org⟩](https://web.archive.org).
- [23] *Moodle - Open-source learning platform*. [cit. 9. 7. 2023]. Dostupný z: [⟨https://moodle.org/⟩](https://moodle.org/).
- [24] *Moodle in numbers during COVID-19*. [cit. 9. 7. 2023]. Dostupný z: [⟨https://moodle.com/news/moodle-in-numbers-during-covid-19/⟩](https://moodle.com/news/moodle-in-numbers-during-covid-19/).
- [25] *Moodle Statistics - stats.moodle.org*. [cit. 9. 7. 2023]. Dostupný z: [⟨https://stats.moodle.org⟩](https://stats.moodle.org).
- [26] *About Moodle - MoodleDocs*. [cit. 9. 7. 2023]. Dostupný z: [⟨https://docs.moodle.org/402/en/About_Moodle⟩](https://docs.moodle.org/402/en/About_Moodle).
- [27] Greenhow, Christine; Lewin, Cathy. Social media and education: Reconceptualizing the boundaries of formal and informal learning. In. *Social Media and Education*. 2019 s. 6–30.
- [28] Meishar-Tal, Hagit; Kurtz, Gila; Pieterse, Efrat. Facebook groups as LMS: A case study. *International Review of Research in Open and Distributed Learning*. 2012, roč. 13, č. 4 s. 33–48.
- [29] *Discord.com*. [cit. 9. 7. 2023]. Dostupný z: [⟨https://discord.com⟩](https://discord.com).
- [30] *How Discord Went From Gaming to Everything - The Story of a \$15B App*. [cit. 9. 7. 2023]. Dostupný z: [⟨https://www.linkedin.com/pulse/how-discord-went-from-gaming-everything-story-15b-app-wishpond/⟩](https://www.linkedin.com/pulse/how-discord-went-from-gaming-everything-story-15b-app-wishpond/).
- [31] *Microsoft Teams*. [cit. 9. 7. 2023]. Dostupný z: [⟨https://teams.microsoft.com/⟩](https://teams.microsoft.com/).
- [32] *What We've Lost ... And What We've Gained - Microsoft.com*. [cit. 9. 7. 2023]. Dostupný z: [⟨https://www.microsoft.com/en-us/worklab/pandemic-lost-and-gained⟩](https://www.microsoft.com/en-us/worklab/pandemic-lost-and-gained).
- [33] *Serverless web application - Azure Architecture Center | Microsoft learn*. [cit. 9. 7. 2023]. Dostupný z: [⟨https://learn.microsoft.com/en-us/azure/architecture/web-apps/serverless/architectures/web-app⟩](https://learn.microsoft.com/en-us/azure/architecture/web-apps/serverless/architectures/web-app).
- [34] *Microsoft Azure Portal*. [cit. 9. 7. 2023]. Dostupný z: [⟨https://azure.microsoft.com/cs-cz⟩](https://azure.microsoft.com/cs-cz).

- [35] *Serverless Computing - AWS Lambda - Amazon Web Services*. [cit. 9. 7. 2023]. Dostupný z: <https://aws.amazon.com/lambda/>.
- [36] *Azure Functions overview - Microsoft Learn*. [cit. 9. 7. 2023]. Dostupný z: <https://learn.microsoft.com/en-us/azure/azure-functions/functions-overview?pivots=programming-language-csharp>.
- [37] Wen, Jinfeng; Chen, Zhenpeng; Jin, Xin; Liu, Xuanzhe. Rise of the planet of serverless computing: A systematic review. *ACM Transactions on Software Engineering and Methodology*. 2023.
- [38] *React Framework*. [cit. 9. 7. 2023]. Dostupný z: <https://react.dev>.
- [39] *Angular - The web development framework for building the future*. [cit. 9. 7. 2023]. Dostupný z: <https://angular.io>.
- [40] *Progressive Vue Framework*. [cit. 9. 7. 2023]. Dostupný z: <https://vuejs.org>.
- [41] *Webpack - JavaScript module bundler*. [cit. 9. 7. 2023]. Dostupný z: <https://webpack.js.org>.
- [42] *Babel - JavaScript Compiler*. [cit. 9. 7. 2023]. Dostupný z: <https://babeljs.io>.
- [43] *ESLint - Pluggable JavaScript Linter*. [cit. 9. 7. 2023]. Dostupný z: <https://eslint.org>.
- [44] Zenchenko, Yaroslav. Paralelismus, asynchronismus a souběh v moderních programovacích jazycích. 2022.
- [45] Taneski, Viktor; Heričko, Marjan; Brumen, Boštjan. Systematic overview of password security problems. *Acta Polytechnica Hungarica*. 2019, roč. 16, č. 3 s. 143–165.
- [46] Pearman, Sarah; Zhang, Shikun Aerin; Bauer, Lujo; Christin, Nicolas; Cranor, Lorrie Faith. Why people (don't) use password managers effectively. In. *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*. 2019 s. 319–338.
- [47] Reese, Ken; Smith, Trevor; Dutson, Jonathan aj. A usability study of five {two-factor} authentication methods. In. *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*. 2019 s. 357–370.
- [48] *React Router*. [cit. 9. 7. 2023]. Dostupný z: <https://reactrouter.com/en/main>.
- [49] *TypeScript: JavaScript with syntax for types*. [cit. 9. 7. 2023]. Dostupný z: <https://www.typescriptlang.org>.
- [50] *Rider: The cross-platform .NET IDE from JetBrains*. [cit. 9. 7. 2023]. Dostupný z: <https://www.jetbrains.com/rider/>.
- [51] Mendrofa, Vinsensius Elman; Liza, Risiko aj. Analisis Performa Owncloud Dan Pydio Pada Aplikasi Layanan Cloud Storage. *SNASTIKOM*. 2022, roč. 1, č. 01 s. 204–212.

- [52] *MasterDC - Specialisté na firemní IT infrastrukturu.* [cit. 9. 7. 2023]. Dostupný z: <https://www.master.cz/>.
- [53] *Nextcloud server - sdílené úložiště na vlastním serveru.* [cit. 9. 7. 2023]. Dostupný z: <https://www.master.cz/nextcloud-server/>.