

Diplomová práce



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Katedra počítačů

## Aplikace pro podporu výuky rekvalifikačního kurzu Základy webového vyvoje

**Václav Smítka**

Školitel: Ing. Karel Frajták, Ph.D.  
Srpen 2023



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Smitka** Jméno: **Václav** Osobní číslo: **478205**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra počítačů**  
Studijní program: **Otevřená informatika**  
Specializace: **Softwarové inženýrství**

## II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

**Aplikace pro podporu výuky rekvalifikačního kurzu Základy webového vývoje**

Název diplomové práce anglicky:

**Web application to support Web development retraining course**

Pokyny pro vypracování:

Navrhněte a naimplementujte webovou aplikaci pro rekvalifikační kurz Tvorba webových stránek. Cílem kurzu je studenty naučit základům HTML, CSS a JavaScriptu za účelem vytvoření jednoduché webové aplikace. Takto nabyté znalosti pak využijí v novém zaměstnání.

Aplikace pomůže studentům jednak během výkladu, kdy bude výklad - tj. demonstrace základů HTML, CSS a JavaScriptu lektorem - přenášén online k nim do prohlížeče společně s interaktivní částí, kde si budou moci psát poznámky, vznášet dotazy a později si zkoušet probíraná témata.

Při domácím samostudiu by pak studenti měli mít možnost si materiál znovu projít a vyzkoušet a zopakovat kroky provedené lektorem. Platforma by měla maximálně zjednodušit přípravu prostředí, ve kterém budou studenti nově nabyté znalosti zkoušet.

Aplikaci řádně otestujte uživatelskými testy.

Seznam doporučené literatury:

Elizabeth Goodman, Mike Kuniavsky and Andrea Moed (2012). Observing the User Experience, Publisher: Morgan Kaufmann, <https://doi.org/10.1016/C2010-0-64844-9>  
Colin Ware. Information Visualization, perception for design, Publisher: Morgan Kaufmann (2019). ISBN 0128128763, 9780128128763  
Bill Buxton . Sketching User Experiences, Publisher: Morgan Kaufmann (2010). ISBN 0080552900, 9780080552903  
Miroslav Disman (2002). Jak se vyrábí sociologická znalost: Příručka pro uživatele, Vydáno: 2002, Karolinum, ISBN: 978-80-246-0139-7

Jméno a pracoviště vedoucí(ho) diplomové práce:

**Ing. Karel Frajták, Ph.D. laboratoř inteligentního testování systémů FEL**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **16.01.2023**

Termín odevzdání diplomové práce: **15.08.2023**

Platnost zadání diplomové práce: **22.09.2024**

Ing. Karel Frajták, Ph.D.  
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.  
podpis děkana(ky)

### III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta

## Poděkování

Děkuji vedoucímu své práce – panu Ing. Karlovi Frajtákovi, Ph.D. za odbornou pomoc s vypracováním diplomové práce. Jeho rady byly zejména při implementaci platformy a tvorbě osnovy práce velmi přínosné. Rovněž děkuji panu Petrovi Floriánovi ze společnosti MAVO s.r.o. za možnost vést kurzy Tvorba www stránek a konzultace související s fungováním Úřadu práce a MŠMT. Na závěr chci poděkovat všem svým blízkým, kteří mě během vypracovávání práce podporovali a především Michaele Trčkové, která mi pomohla s pravopisnou částí práce.

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 15. 8. 2023

## Abstrakt

Tato diplomová práce se věnuje návrhu a vývoji platformy pro podporu výuky rekvalifikačních kurzů, specializovaných na webové technologie. Dále se také věnuje návrhu a vývoji rozšíření platformy o možnost podpory výuky i dalších programovacích jazyků, než jen v prohlížeči podporovaného JavaScriptu.

Text práce obsahuje seznámení se s problematikou rekvalifikačních kurzů v České republice a ukazuje i alternativní způsoby vzdělávání v IT. Poté přibližuje technologie potřebné pro webové kód editory.

Implementace využívá jazyku TypeScript a knihoven React, Redux, CodeMirror, Quill, ShareDB a Feathers.js pro serverovou část.

**Klíčová slova:** Rekvalifikační kurz, výuka webových technologií, webový kód editor, zjednodušení výuky programování, integrace s GitHubem, webová aplikace

**Školitel:** Ing. Karel Frajták, Ph.D.  
Laboratoř inteligentního testování softwaru,  
Katedra počítačů,  
Karlovo náměstí 13,  
121 35 Praha 2

## Abstract

This thesis focuses on the design and development of a platform that supports teaching of retraining courses specialized in web technologies. It also discusses the design and development of the platform extension which would support learning other programming languages than JavaScript, which is supported by the browser.

The text of the thesis includes an introduction to the issue of retraining courses in the Czech Republic and shows alternative ways of IT education. Then it introduces the technologies needed for web code editors.

The implementation uses TypeScript and React, Redux, CodeMirror, Quill, ShareDB and Feathers.js libraries for the server side.

**Keywords:** Retraining course, learning web technologies, online code editor, simplifying learning programming, integration with GitHub, web application

**Title translation:** Web application supporting Web development retraining course

# Obsah

<b>1 Úvod</b>	<b>1</b>
<b>2 Rekvalifikační kurzy Tvorba www stránek</b>	<b>3</b>
2.1 Rekvalifikace v ČR.....	3
2.2 Poskytovatelé .....	5
2.3 Cílové skupiny .....	8
2.4 Uplatnění .....	8
2.5 Alternativy .....	9
<b>3 Webový kód editor</b>	<b>13</b>
3.1 Webový vs. desktopový kód editor	13
3.2 Limitace webového kód editoru .	14
3.3 WebAssembly .....	15
<b>4 Návrh vzdělávací platformy</b>	<b>17</b>
4.1 Cíloví uživatelé .....	17
4.2 Požadavky na platformu .....	19
4.3 Použité technologie a knihovny .	22
4.4 Architektura platformy .....	29
4.5 Uživatelské rozhraní.....	30
<b>5 Implementace</b>	<b>35</b>
5.1 Vzdělávací platforma .....	35
5.2 Webový kód editor.....	43
5.3 Chat .....	47
5.4 Integrace s GitHubem .....	48
<b>6 Ověření kvality</b>	<b>49</b>
6.1 Příprava testování .....	49
6.2 Využití při výuce .....	50
6.3 Administrační prostředí .....	55
<b>7 Závěr</b>	<b>57</b>
<b>Literatura</b>	<b>61</b>
<b>A Screenshoty platformy</b>	<b>67</b>
<b>B Testovací scénáře</b>	<b>75</b>
B.1 Registrace lektora .....	75
B.2 Tvorba kurzu .....	76
B.3 Tvorba kurzu se šablonou .....	77
B.4 Vložení studenta do kurzu .....	78
B.5 Přípravení lekce .....	79
B.6 Editování a viditelnost .....	80
<b>C Zpětná vazba z testování</b>	<b>83</b>
C.1 Chyby .....	83
C.2 Podněty k vylepšení .....	84
<b>D Struktura příloh elektronické verze práce</b>	<b>85</b>

## Obrázky

4.1 Diagram užití webového kód editoru .....	20	A.2 Screenshot formuláře pro registraci lektora .....	68
4.2 Diagram užití platformy .....	21	A.3 Screenshot administrace lekcí ..	68
4.3 Struktura Reduxu - převzato[1] .	23	A.4 Screenshot administrace studentů	68
4.4 Screenshot ukázky variability Ant design komponenty pro načítání stránky využívané v platformě ....	24	A.5 Screenshot registračního formuláře studenta .....	69
4.5 Screenshot využití Styled Componets v platformě .....	25	A.6 Screenshot detailu lekce z pohledu lektora .....	69
4.6 Screenshot CodeMirror editoru z platformy .....	26	A.7 Screenshot editoru z pohledu lektora při přípravě cvičení .....	70
4.7 Screenshot Quill editoru z platformy .....	27	A.8 Screenshot editoru z pohledu studenta .....	71
4.8 Drátěný model webového kód editoru .....	31	A.9 Screenshot chatovacího vlákna z pohledu studenta studenta .....	72
4.9 Drátěný model administrace platformy .....	32	A.10 Screenshot mailu příchozí po registraci studenta .....	72
4.10 Screenshot bočního panelu z pohledu lektora .....	33	A.11 Screenshot mailu příchozí po odevzdání cvičení .....	72
5.1 Diagram nasazení platformy ....	35	A.12 Screenshot struktury migračního souboru pro založení databáze ....	73
5.2 Relační schéma databáze .....	37		
5.3 Diagram serverových částí platformy .....	38		
5.4 Diagram klientských částí platformy .....	40		
5.5 Aktivity diagram tvorby kurzu .	41		
5.6 Aktivity diagram přidávání studenta do kurzu .....	42		
5.7 Stavový diagram lekce .....	42		
5.8 Stavový diagram plnění lekce studentem .....	43		
5.9 Diagram komponent webového kód editoru .....	44		
5.10 Struktura repozitáře ukládající výstupy editoru .....	46		
5.11 Sekvenční diagram tvorby zadání lekce .....	47		
6.1 Administrační rozhraní cloudové platformy Render .....	50		
6.2 Příklad řešení lekce Formulář ...	52		
6.3 Příklad řešení lekce Funkce, pole a objekt .....	54		
A.1 Screenshot přihlašovacího formuláře .....	67		



## Tabulky

2.1 Porovnání poskytovatelů rekvalifikačních kurzů nebo obsahově podobných kurzů bez rekvalifikace .	7
4.1 Seznam využitých ikon a jejich významů.....	33



# Kapitola 1

## Úvod

Když se v dnešní době člověk rozhodne začít pracovat v IT a nemá žádné předchozí zkušenosti, znamená to spoustu vynaloženého úsilí. I když existuje celá škála možností a zkratk, těžko očekávat, že na ně zájemce narazí. Většinou to znamená spoustu proseděných hodin u počítače ve víře ve vysněné zaměstnání. Občas ani první placená práce nemusí znamenat úspěch. Mnoho firem není pro zaměstnání začátečníka uzpůsobena. Pro začátečníky to většinou znamená nadměrný stres a časté nechápavé výrazy nadřízených.

Pracuji jako webový vývojář přes 5 let. Svě začátky jsem zažil na univerzitě, v laboratoři inteligentního testování systémů. Díky této příležitosti jsem měl možnost se rozvíjet všestranně. Zároveň jsem v sobě postupně objevil chuť předávat své nabyté zkušenosti dál. Tři roky působím jako lektor webových kurzů. Na škole třetím rokem cvičím studenty prvního ročníku bakalářského studia testování softwaru.

Odučil jsem přes 1000 hodin rekvalifikačních kurzů. Učil jsem jak úplně základy práce s počítačem, tak webové technologie. Většina kurzů probíhala prezenčně, ale během covidové doby jsem vedl kurzy i on-line.

Primární problém většiny účastníků je pojmenovat vzniklý problém. Ať se jedná o komplikaci v rámci nastavení operačního systému, nebo nefunkčnosti webové komponenty. Tento problém byl výraznější, když probíhala on-line výuka. Častým řešením je, že se student o řešení problému ani nepokusí. V lepším případě si student přivolá pomoc. Je na lektorovi, aby se situace aktivně zhostil a podpořil tápající studenty. V případě on-line výuky je tato snaha téměř nemožná.

V rámci této diplomové práce jsem vytvořil platformu pro podporu výuky webových technologií během rekvalifikačních kurzů. Důraz je kladen na vazbu mezi studentem a lektorem. Platforma umí sdílet vytvářený kód v reálném čase. Zároveň umožňuje si mezi sebou dopisovat v privátním kanále, vázaném ke konkrétnímu cvičení. Nechybí ani možnost v chatu poslat odkaz na konkrétní část kódu.

Text diplomové práce je rozdělen do pěti základních celků, v nichž popisují celý proces vývoje platformy a její integrace. V první části **2** se zabývám povahou rekvalifikačních kurzů a mapuji možnosti zájemců o práci na poli webových technologií. V druhé části **3** se věnuji technologickému zázemí platformy. Porovnávám možnosti a výhody desktopového a webového editoru

kódu. Také se blíže zabývám otázkou rozšiřitelnosti platformy na výuku jiných, než webových technologií. Ve třetí části 4 popisuji požadavky a návrh platformy. Čtvrtá část 5 obsahuje způsob implementace jednotlivých částí platformy. V poslední části 6 se zaměřuji na ověření kvality platformy pomocí uživatelských testů.

Smyslem této práce je nejen aplikace vědomostí získaných během studia, ale především zmapování aktuální situace poskytování vzdělávacích kurzů na poli webových technologií, podpora a zefektivnění výuky. Podpora studentů při prvních krůčcích k vysněné práci. Zároveň poskytnutí nástroje lektorům pro efektivnější řízení výuky nejen on-line kurzů.

Není to tak dlouho, co jsem sám byl tápajícím a díky možnosti vidět problematiku z obou stran věřím, že mé zkušenosti a platforma podpoří i další při hledání efektivnějšího způsobu výuky informačních technologií.

## Kapitola 2

### Rekvalifikační kurzy Tvorba www stránek

Osobně jsem vyučoval rekvalifikační kurz Tvorba www stránek, akreditovaný MŠMT ČR: č.j. MSMT-8089/2020-1 společností MAVO s.r.o.. Během kurzu trvajícího 80 hodin se student seznámí s HTML, CSS a programovacími jazyky JavaScript a PHP. Cílem je předat studentovi základní znalosti všech částí webového vývoje.

V této kapitole představuji pravidla rekvalifikačních kurzů v České republice. Bližším rozbohem rekvalifikačního kurzu společnosti MAVO a představením různorodých zástupců skupiny poskytovatelů kurzů chci představit výběr pro zájemce o kurz. Dále se zamýšlím nad uplatněním a možnými alternativami rekvalifikačních kurzů.

#### 2.1 Rekvalifikace v ČR

Dle Úřadu práce ČR se rekvalifikací rozumí získání nové kvalifikace a zvýšení, rozšíření nebo prohloubení dosavadní kvalifikace, včetně jejího udržování nebo obnovování. Za rekvalifikaci se považuje i získání kvalifikace pro pracovní uplatnění fyzické osoby, která dosud žádnou kvalifikaci nezískala. Při určování obsahu a rozsahu rekvalifikace se vychází z dosavadní kvalifikace, zdravotního stavu, schopností a zkušeností fyzické osoby, která má být rekvalifikována formou získání nových teoretických a praktických dovedností v rámci dalšího profesního vzdělávání (§ 108 odst. 1 zákona č. 435/2004 Sb., o zaměstnanosti, ve znění pozdějších předpisů).[2]

Rekvalifikační kurzy patří mezi nejvýznamnější nástroje aktivní politiky nezaměstnanosti. Aby se kurz stal rekvalifikačním, musí být akreditovaný Ministerstvem školství, mládeže a tělovýchovy. Akreditace obsahuje vzdělávací plán, podobu závěrečné zkoušky i vstupní předpoklady účastníků a profil absolventa. Kurzy se akreditují na dobu 3 let.[3]

Dle ustanovení Akreditační komise MŠMT ze dne 1. 4. a 5. 5. 2008 není možné akreditovat kurzy, které se věnují pouze dílčí činnosti počítačové gramotnosti či konkrétním počítačovým programům. Je možné akreditovat pouze takové kurzy, které vedou k získání nové plné kvalifikace.[4]

Rekvalifikační kurz Tvorba www stránek se zaměřuje na získání rekvalifikace pro oblast CZ-ISCO 2513 vývoj webu a multimédií[5] nebo CZ-NACE J 620 – činnost v oblasti informačních technologií.[6]

Úřad práce může na rekvalifikační kurzy přispět maximálně 50 000 Kč v průběhu 3 let. Nejedná se o vymahatelnou částku. Je nutné o příspěvek požádat. Úřad práce následně hradí kurz po úspěšném absolvování přímo rekvalifikačnímu zařízení. V případě neúspěchu je nucen celé kurzovné uhradit účastník sám.[2]

### ■ 2.1.1 Osnova kurzu

Osnovy rekvalifikačních kurzů se liší. Každý poskytovatel musí akreditovat vlastní kurz. Na druhou stranu je pro poskytovatele výhodné osnovy ladit podle očekávání Úřadu práce. Posléze mohou soutěžit o dodavatele rekvalifikačních kurzů v krajích.

Bližší rozvedu kurz, který jsem sám učil. Obsah výše zmíněného kurzu Webdesigner/Tvorba www stránek akreditován MŠMT ČR: č.j. MSMT-10099/2023-2 je rozdělen do několika sekcí a postihuje všechny důležité části vývoje od samotného HTML, až po marketing na internetu. Kurz má dotaci 80 hodin vyučovaných ve 14 dnech. Běžný výukový den se vyučuje od 8:30 do 15:00. Kurz je zakončen závěrečnou zkouškou. Kurz má akreditovanou osnovu:[7]

- Poučení o bezpečnosti a ochraně zdraví při práci (1 h)
- Strategie vytváření marketingově úspěšných www stránek (6 h)
- HTML (12 h)
- CSS (kaskádové styly) (13 h)
- Grafika a web (4 h)
- Navigační prvky (4 h)
- Tabulky (8 h)
- Bloky a Rámy (4 h)
- Zvuk a obraz (2 h)
- JavaScript (13 h)
- PHP (13 h)

Bližší rozdělení do jednotlivých dnů a obsah jednotlivých témat je na lektorovi. Kurz probíhá na počítačích poskytnutých společností MAVO prezentně. Pro výuku se využívá prostředí Visual Studio Code, které umožňuje vytváření HTML, CSS, JavaScriptu i PHP dokumentů. Pro spouštění PHP scriptů se využívá XAMPP. Nově se začaly zařazovat základy Wordpressu, jako zástupce redakčních systémů. Zároveň lze na Wordpressu demonstrovat základní povahu jazyka PHP. Seznámení s Wordpressem vítají především studenti se záměrem vytvořit si vlastní webové stránky bez ambice vstupu na trh práce jako webových vývojářů.

## 2.2 Poskytovatelé

Podle MŠMT je v tuto chvíli v České republice akreditováno 71 kurzů, týkajících se webového vývoje, provozovaných 27 společnostmi. Představuji několik z nich. Kompletní seznam je ke stažení na stránkách MŠMT. Žádný rekvalifikační kurz webových technologií nevede k profesní rekvalifikaci. Díky tomu poskytovatelé mají volnější ruku při definici závěrečné zkoušky. Sedm kurzů vyžaduje středoškolské vzdělání, 23 kurzů vyžaduje základní vzdělání a zbytek prerekvizity neuvádí. Nejkratší kurzy trvají 80 hodin. Nejdelší kurz poskytuje Green Fox Academy CZ s.r.o. s celkovou časovou dotací 390 hodin.[8]

### 2.2.1 MAVO s.r.o.

Společnost MAVO s.r.o., poskytuje celou škálu rekvalifikačních kurzů. Společnost nabízí kurz Tvorba www stránek za cenu 25 000 Kč s kapacitou 15 osob. Vyučuje se prezenčně. Během covidové situace byl kurz vyučován na platformě MS Teams, nicméně společnost není na tuto formu výuky připravena. Zároveň nevznikají žádné materiály, které by společnost poskytovala studentům. Každý student je odkázán na tvorbu vlastních poznámek.[7]

### 2.2.2 IT Networks

Web IT Networks obsahuje množství veřejně dostupných materiálů k tvorbě webových stránek, kde důkladně vysvětlují a popisují základy každé webové technologie, rovněž poskytují rekvalifikační kurzy. Konkrétně 6 kurzů zabývajících se vývojem webových stránek, podle jednotlivých technologií. Vedou kurzy programování v jazycích C# .NET, Java, Kotlin, Python, PHP a JavaScript.

Nejblíže kurzu poskytovanému společností MAVO je kurz „Programátor www v JavaScriptu“. Vyučuje se ve 121 hodinách, kde je 41 hodin odučeno prezenčně a zbylých 80 hodin on-line, kdy si student procvičuje látku a čte články. Kurz stojí pro samoplátce 34 900 Kč. Odhadovaná doba trvání jsou 3 měsíce. Kurz nad rámec kurzu společnosti MAVO obsahuje výuku objektově orientovaného programování v JavaScriptu, testování, verzování, UML a Node.js. Zároveň neobsahuje základy PHP.[9]

### ■ 2.2.3 PrimaFuture.cz s.r.o.

Stejně jako společnost MAVO se specializuje na výuku a školení. Má víceletou praxi na trhu. V odvětví IT nabízí mimo jiné specializovaný kurz na práci ve redakčním systému Wordpress. Nabízí jak prezenční, tak on-line kurzy. Také nabízí kurz „Programátor www aplikací“ s dotací 150 hodin rozložený do 22 dní. Prezenční forma kurzu stojí 40 000 Kč a on-line verze 30 000 Kč. Během kurzu se využívá Visual Studio Code. Oproti kurzu společnosti MAVO obsahuje jQuery, AJAX, MySQL a 1 den školení v PHP frameworku Laravel.[10]

### ■ 2.2.4 Green Fox Academy s.r.o.

Firma nabízející kurzy v odvětví webového vývoje garantující zaměstnání v nasmlouvaných společnostech pro každého studenta. Kurzy trvají buď 4,5 měsíce v případě intenzivního kurzu, nebo 10 měsíců v případě varianty studia při zaměstnání. Nabízejí prezenční i on-line kurzy.

Materiály jsou v angličtině. Kurzy jsou rozděleny do tří modulů: přípravný, orientační a projektový. Kurz obsahuje širokou škálu technologií HTML, CSS, JavaScript, React nebo Node.js. Vyučují také návrhové vzory a databázové struktury. K výuce se využívá Visual Studio Code, GitHub a Discord. Kurz stojí 182 000 Kč.[11]

### ■ 2.2.5 Abeceda PC Consulting a.s.

Poskytují spektrum výukových forem. Od konzultací po firemní školení. Rekvalifikační kurz „Tvorba www stránek“ nabízí prezenčně i on-line. Kurz trvá 80 hodin rozložených do 10 dní. Prezenční forma kurzu stojí 33 200 Kč a on-line 30 000 Kč. Oproti kurzu společnosti MAVO osnova kurzu obsahuje výuku redakčních systémů Wordpress a PrestaShop. Společnost zároveň věnuje část kurzu SEO. Na druhou stranu věnuje minimální pozornost výuce samotnému programování v programovacích jazycích JavaScript nebo PHP.[12]

### ■ 2.2.6 Praha Coding School s.r.o.

Kurz s dotací 72 hodin je vyučován v 8 týdnech. Student za kurz zaplatí 27 500 Kč. Náplň je podobná kurzu společnosti MAVO. Navíc věnuje pozornost nástroji Adobe Photoshop. Společnost stejně jako Green Fox Academy nabízí následné zaměstnání v partnerských společnostech.[13]

### ■ 2.2.7 NICOM a.s.

Akreditovaný kurz má dotaci 104 hodin rozložených do 13 dní. Kurz lze absolvovat jak prezenční formou, tak on-line. Kurz stojí 21 990 Kč. Obsahově je kurz podobný kurzu od společnosti MAVO. Navíc se věnuje výuce nástroje Adobe Photoshop.[14]



### 2.2.8 Edu Partners s.r.o.

Společnost nabízí dva rekvalifikační kurzy na tvorbu www stránek. Jeden je zaměřený více na tvorbu webových stránek a druhý na webový marketing. Kurz „od základů po samotného tvůrce“ má dotaci 80 vyučujících hodin. Vyučuje se ve 35 dnech. Kurzy je možné absolvovat jak prezenčně, tak on-line. Během kurzů se využívá Visual Studio Code. Nad rámec kurzu společnosti MAVO se vyučuje legislativa, SEO, Google Analytics a SCRUM. Na druhou stranu se nevyučují základy PHP.[15]

### 2.2.9 CZECHITAS z.ú.

Jako poslední uvádím společnost, která neposkytuje akreditované rekvalifikační kurzy. Jejím posláním je přivést do světa IT ženy. Od roku 2017 poskytují kurzy s cílem zvýšit množství žen v IT v České republice. Vládkovou lodí kurzů je Digitální akademie: Web. Cílem je připravit studentky na juniorní pozice webových vývojářů. Během 15 týdnů jsou studentky provedeny všemi částmi vývoje. Od návrhu po realizaci vlastního projektu. Czechitas se snaží maximálně podpořit studentky v jejich kariéře a poskytuje zároveň exkurze do firem, hackathon a další možnosti setkání se s lidmi z praxe. Kurz stojí 21 990 Kč. Kurz nad rámec kurzu od společnosti MAVO obsahuje výuku Reactu, Gitu a prezentačních dovedností. Neobsahuje základy PHP. Před nástupem do kurzu se počítá s tím, že účastnice jsou seznámeny se základy HTML, CSS a úvodem do programování v libovolném jazyce.[16]

### 2.2.10 Shrnutí

Společností existuje celá řada. Celý seznam udělených akreditací Ministerstvem a tělovýchovou lze nalézt na stránkách.[8] Pomocí společností výše uvedených demonstrují rozdílnost přístupu poskytovatelů k rekvalifikačnímu kurzu Tvorba www stránek. Kurzy se liší svou náplní, délkou i cenou. Přehled uvádím v tabulce:

Poskytovatel	Rekvalifikace	Délka	Cena
MAVO s.r.o.	MSMT-10099/2023-2	14 dní	25 000 Kč
IT Networks	MSMT-1493/2022-7	12 týdnů	34 900 Kč
PrimaFuture.cz s.r.o.	MSMT-8269/2023-4	22 dní	40 000 Kč
Green Fox Academy s.r.o.	MSMT-13850/2021-3	18 týdnů	182 000 Kč
Abeceda PC Consulting a.s.	MSMT-20370/2021-2	10 dní	33 200 Kč
NICOM a.s.	MSMT-8650/2022-2	13 dní	21 990 Kč
Edu Partners s.r.o.	MSMT-6316/2022-2	35 dní	24 500 Kč
Praha Coding School s.r.o.	ne	8 týdnů	27 500 Kč
CZECHITAS z.ú.	ne	15 týdnů	21 990 Kč

**Tabulka 2.1:** Porovnání poskytovatelů rekvalifikačních kurzů nebo obsahově podobných kurzů bez rekvalifikace

## 2.3 Cílové skupiny

Poskytovatelé cílí primárně na zájemce a uchazeče o zaměstnání registrované na Úřadu práce. Snaží se mít osnovy upravené tak, aby si účastnice mohly nechat proplatit kurz Úřadem práce. Zároveň Úřad práce vypisuje soutěže o dodavatele kurzů v konkrétním kraji. Díky tomu poskytovatel získá zaručený příjem.

Druhá skupina lidí si kurz hradí sama s motivací doučit se nebo oživit si znalosti webových technologií.

Věk účastníků se pohybuje od čerstvých absolventů základních nebo středních škol po účastníky starší padesáti let. Z vlastní zkušenosti mohou říct, že převážně účastníci ve věku do 35 let chtějí změnit zaměstnání. Starší účastníci mají primárně motivaci doplnit si vědomosti z oboru webového vývoje a například si vytvořit vlastní webové stránky nebo lépe porozumět IT oddělení ve stávající práci.

## 2.4 Uplatnění

Uplatnění se liší podle zaměření kurzu i jeho rozsahu. Zároveň hraje roli pojetí kurzů poskytovatelem. Společnosti jako například MAVO s.r.o., NICOM a.s., nebo Edu Partners s.r.o. pouze učí a po kurzu se účastníkům nijak nevěnují. Na rozdíl od toho Czechitas z.ú. poskytuje zkušební pohovory v partnerských společnostech. Green Fox Academy s.r.o. nebo Praha Coding School absolventům nabízejí zaměstnání ať v partnerských společnostech, nebo u nich samotných.[11]

Účastník je během kurzů primárně připravován na pozici kodéra webových stránek, nebo vývojáře klientské části webové stránky. Získá základy programování v JavaScriptu a některé společnosti poskytují i úvod do pokročilých technologií, jako je React, Angular, nebo Vue.

Dle rozhovorů s účastníky svých kurzů mohou konstatovat, že osnovy vyžadované Úřadem práce se rozcházejí s požadavky na pozice webových vývojářů. Zaměstnavatelé očekávají hlubší znalost JavaScriptu. Zároveň je očekávané portfolio s projekty. Během 80 hodinových kurzů není k vytváření projektů prostor. Obzvláště když je student úplný začátečník.

Na základě osnov vystavených poskytovateli rekvalifikačních kurzů na svých stránkách a rozhovory s absolventy kurzů společností MAVO, IT Networks a Czechitas je zřejmé, že žáci zvládnou během kurzu získat pouze úplné základy a je na jejich iniciativě, aby se potřebné znalosti doučili sami pro úspěch ve výběrovém řízení na pozici juniorního webového vývojáře.

## ■ 2.5 Alternativy

Absolvování rekvalifikačního kurzu je pouze jednou z mnoha možností, jak se vzdělat v oblasti webových technologií. V IT lze získat vzdělání jak středoškolské, tak vysokoškolské. Větší společnosti mají vlastní vzdělávací programy pro nováčky. Existuje celá řada materiálů, které jsou přístupné na internetu zdarma. V poslední řadě bych rád zmínil tištěné publikace.

### ■ 2.5.1 Středoškolské vzdělání

Základní vzdělání na obecných školách obsahuje základy HTML a CSS. Podle portálu STŘEDNÍŠKOLY.CZ existují obory se zaměřením na informační technologie nebo multimediální internetové služby. Absolventi mají predispozice pro získání dobré práce v oboru díky dlouholeté praxi v oboru.[17]

### ■ 2.5.2 Vysokoškolské vzdělání

Podle portálu VysokeSkoly.cz je v České republice možné obory související s informatikou studovat v rámci bakalářského studia na 48 školách a magisterské na 45 školách. Všechny obory mají výrazný přesah nad rámec webového vývoje. Webový vývoj je součástí oborů softwarového inženýrství. Zároveň se oproti rekvalifikačním kurzům jednotlivé části vývoje rozdělují do samostatných předmětů.[18]

### ■ 2.5.3 Pozice pro nováčky

Velké společnosti, jako například Vodafone, Škoda Auto, Komerční banka nebo Česká spořitelna mají své tzv. trainee pozice. Pro přijetí na tyto pozice zpravidla není třeba vzdělání v oboru. Důležitá je motivace a zázemí uchazečů. Vysokoškolské vzdělání je výhodou. Stejně tak jakákoliv zkušenost v IT.[19]

### ■ 2.5.4 On-line výukové platformy

V dnešní době mnoho lidí rádo využívá internet jako zdroj informací. Lze tak najít i bezplatné kurzy a webináře například na YouTube. Kanály nabízející celou škálu vzdělávacích videí, pokrývajících všechny oblasti webového rekvalifikačního kurzu a mnohem více:

- David Šetek – Hackni svou budoucnost[20]
- ROB WEB s yablkom[21]
- jaknaweby[22]
- developedbyed[23]

Zajímavou alternativou jsou gamifikované<sup>1</sup> formy vzdělávání:

- SCRATCH[25]
- CodeCombat[26]
- CodinGame[27]
- codewars[28]

Předem nahrané kurzy mají proti prezenčním kurzům výhodu, že každý student může postupovat vlastním tempem. Současně materiály studentovi zůstávají a může se k nim vždy vrátit. Zároveň má student možnost látku kdykoliv pozastavit nebo přehrát znovu. On-line platformy pro výuku webových technologií jsou například:

- Udemy[29]
- Coursera[30]
- Codecademy[31]
- Free Code Camp[32]

## 2.5.5 Knihy a články

Pro hlubší pochopení konceptů, ale i konkrétních technologií se hodí sledovat rozsáhlé blogy, nebo si pořídit knihy. Knihy mají nevýhodu, že informace v nich obsažené se mohou v průběhu času stávat neaktuální. To ale neplatí o knihách o teorii, nebo obecných přístupech. Mezi knihami mohu doporučit:

- Tvorba www stránek v HTML a CSS (2019)[33]
- Web ostrý jako břitva (2014)[34]
- Efektivní testování softwaru (2016)[35]

IT Networks má na svém webu mnoho článků k webovým tématům a dalším problematikám v IT. Pokud člověk zvládne studium v anglickém jazyce<sup>2</sup> uvádím dva oblíbené portály:

- Smashing Magazine[36]
- Frontend Mag[37]

Dalším kvalitním zdrojem jsou vědecké publikace a závěrečné práce studentů vysokých škol. Obzvláště bakalářské práce jsou dobrým zdrojem informací, včetně dokumentace. Publikace lze vyhledávat pomocí Google Scholar.[38] Další variantou je využít standardního Google vyhledávače s definicí datového typu požadovaných výsledků „filetype:pdf“[39]

<sup>1</sup>Gamifikace (anglicky gamification) – Použití herních prvků a herních principů v kontextech, které nejsou ve hře. Gamifikace běžně používá konstrukční prvky hry, které se používají v non-herních situacích pro zlepšení interakce uživatelů, organizační produktivity, flow, učení, crowdsourcingu, nábory zaměstnanců a vyhodnocení, snadnosti použití, užitečnosti systémů, fyzického cvičení a dalších.[24]

<sup>2</sup>Drtivá většina studentů, které jsem potkal během výuky rekvalifikačních kurzů, měla problém s nedostatečnou znalostí anglického jazyka

### ■ 2.5.6 Dokumentace a příklady

Jako poslední bych rád přiblížil způsob, který využívají převážně pokročilí programátoři. Když programátor ví, co chce a hledá konkrétní řešení svého problému, často skrze fóra a doporučení nalezne výčet technologií, které řeší jeho problém. Následně si vyhledá dokumentace technologií a pomocí návodů, příkladů a popisů implementuje své řešení.

Tímto způsobem se učí nové technologie, pro které nebyly zatím vytvořené kurzy. Využívá platform, jako je například Stack Overflow<sup>3</sup>, nebo GitHub.

---

<sup>3</sup>Stack Overflow je největší a nejdůvěryhodnější on-line komunita, kde se vývojáři mohou učit, sdílet své znalosti programování a budovat svou kariéru.[40]



## Kapitola 3

### Webový kód editor

Kód sám o sobě je text. Díky tomu v minulosti stačilo mít k dispozici textový editor a člověk mohl programovat. Bylo pouze potřeba takového editoru, který umožňuje uložit text bez formátování.

I když lze programovat webové stránky v Poznámkovém bloku, nebo ve Vimu v Linuxu, dnes se dává přednost profesionálním vývojovým prostředím, jako je například IntelliJ IDEA[41], nebo Visual Studio Code[42]. Díky nim lze formátovat kód, barevně odlišovat jednotlivé části kódu nebo usnadňovat odhalování chyb. Cílem je zefektivnění vývoje.

Pro účely výuky webových technologií se často používají jednodušší prostředí. Jak bylo zmíněno v předešlé kapitole, jedná se o Visual Studio Code. Na YouTube jsem se také setkal s textovým kód editorem Atom. Pro účely sdílení příkladů nebo drobné ukázky se využívají webová prostředí JSFiddle[43], nebo CodePen[44]. Zároveň existují i on-line vývojová prostředí, jako například CodeSandBox[45], nebo PlayCode[46].

#### 3.1 Webový vs. desktopový kód editor

Kód editory lze rozdělit do dvou hlavních skupin. Jedny, které jsou programy v počítači a druhé, které jsou přístupné z internetu a otevírají se v prohlížeči. Podle použitých technologií mají své výhody a nevýhody.

Webové editory potřebují internet pro fungování. Zároveň se vyznačují velkou dostupností. Například PlayCode nebo CodePen ve své placené verzi poskytují i editování jednoho kódu více lidmi zároveň. Některé desktopové kód editory dnes také poskytují sdílení kódu. Visual Studio Code obsahuje rozšíření Live Share, pomocí kterého lze v reálném čase spolupracovat na kódu jednoho člověka.[47] Podobný plugin má i IntelliJ IDEA, který se jmenuje „Code with Me“[48].

Velkou výhodou webových editorů je zjednodušení a zrychlení začátku práce na projektu. Stačí se pouze přihlásit k prostředí a člověk může začít vyvíjet. V případě desktopové verze je nutné: naklonovat si repozitář, nainstalovat si všechny závislosti a mít správné verze nástrojů, v případě webového vývoje v Node.js a npm.

Pokud se jedná o úplné základy a je třeba studentům neplést hlavu, není třeba nic klonovat z repozitářů, ani instalovat balíčky. Pak je nutné kód

otevírat ručně v prohlížeči a při každé změně kód uložit a znovu načíst v prohlížeči. Webový kód editor zjednodušuje proces na kliknutí na jedno tlačítko. Některé webové editory výsledky ukazují v reálném čase.

Další výhodou webových editorů je obsluha prostředí. Samotné webové prostředí je specializované pro konkrétní účely a obsahuje pouze konkrétní nastavení. Také není potřeba manuálních aktualizací. Prostor se aktualizuje automaticky. Specializované prostředí lze nalézt například:

- [Tutorials Point\[49\]](#)
- [JDoodle Java editor\[50\]](#)
- [Programiz Python editor\[51\]](#)
- [OnlineGDB databázový editor\[52\]](#)
- [Online Java IDE\[53\]](#)
- [Online Python IDE\[54\]](#)
- [SQL Online\[55\]](#)

Desktopové editory jsou vytvářené pro obecné účely. IntelliJ IDEA i Visual Studio Code lze využít k programování ve více jazycích a obsahují velké množství uživatelského přizpůsobení. Visual Studio Code lze rozšířit pomocí doinstalování rozšíření a po případné konfiguraci počítače ho lze použít pro vývoj i v programovacím jazyce Python, Java, JavaScript, SQL, C++ a mnoha dalších. IntelliJ IDEA podporuje v základu pouze jazyky Java, Kotlin, Scala, Groovy a webové technologie (HTML, CSS a JavaScript). Pomocí rozšíření ho lze použít i k programování v jazycích Python, Ruby, PHP, SQL a GO. Pro ty však lze využít nástroje od stejné společnosti – JetBrains. Pro obsluhu prostředí je nutné stahovat a instalovat aktualizace.

Webové kód editory buď kód ukládají ve vlastním úložišti, nabízejí integraci s repositáři, nebo nabízejí možnost si kód stáhnout. Ve dvou prvních případech je kód ve větším bezpečí, kdy o něj prakticky nelze přijít jinak, než ukončením fungování služby. U desktopových editorů je ztráta kódu rizikovější. V případě poškození počítače je kód nenávratně ztracen.

## 3.2 Limitace webového kód editoru

Desktopové editory jsou omezené primárně hardwarem počítače, na kterém jsou nainstalované. V případě webových kód editorů je důležité zohlednit internet a prohlížeč, jakožto klíčové technologie pro funkčnost editoru. Bezpečnost je důvodem omezení pro přístup k adresářové struktuře počítače. Proto musí být všechny soubory ukládané jako textový řetězec v prohlížeči po dobu jeho otevření.[56]



## ■ 3.3 WebAssembly

Jedná se o technologii vyvíjenou pod zastřešením W3C s cílem zrychlit kritické části aplikace. Díky WebAssembly je možné v prohlížeči spustit kód v jazycích, jako je například C nebo Python. Jazyk je převáděn do pseudo symbolických adres.<sup>[57]</sup>

Pro výuku dalších programovacích jazyků, než je JavaScript, je nutné si vždy nejdříve připravit prostředí. Pro jazyky C a C++ je třeba kompilátor. PHP, Java nebo Python potřebují interpret. Díky WebAssembly je možné připravit prostředí pro studenty v prohlížeči. Nebudou si muset nic instalovat.



## Kapitola 4

### Návrh vzdělávací platformy

Myšlenka vzdělávací platformy pro výuku rekvalifikačních kurzů vznikla na začátku roku 2021, kdy jsem vedl svůj druhý rekvalifikační kurz – Tvorba www stránek. Postupně se začal rodit koncept platformy, ve které by bylo možné efektivněji předávat zkušenosti a zároveň, aby ulehčila účastníkům začátky.

Postupně jsem si všímal různých opakujících se jevů. Lidé, kteří se účastnili mnou vedených rekvalifikačních kurzů se v co nejkratším časovém úseku chtěli naučit ideálně vše o tvorbě webových stránek. Při kurzech webových technologií to znamenalo, že během prvních 3 vyučovacích dní byli aktivní. Kladli dotazy a snažili se plnit všechny zadané úkoly. Poté aktivita začala upadat a mě jako lektora stálo více úsilí udržet jejich pozornost a naučit je další technologie. Důvodem bylo přehlcení novými informacemi.

Pouze 20 % studentů přibližně tušilo, co je čeká. Měli základy ze střední školy, nebo si sami zkusili vytvořit webovou stránku. Ať už pomocí HTML, CSS a JavaScriptu, nebo ve Wordpressu. Zbytek byl během prvního dne, kdy dělám úvod do webových technologií, ohromený komplexitou. Představa, co je čeká v následujících 13 pracovních dnech, byla pro ně nepředstavitelná.

Mým cílem jako lektora bylo provést studenty technologiemi, obsaženými v osnově kurzu. Zároveň nemám ambice tvořit programátory jako na běžícím páse. Jsem přesvědčený, že bez kvalitního základu se ze člověka za 14 pracovních dní programátor nestane. Snažím se vytvořit studentům nezaujatý a obecný základ pro celý webový průmysl. Zároveň, kvůli akreditované osnově, je mým cílem vytvořit kvalitní základy v základních technologiích – HTML, CSS a JavaScriptu. PHP probírám pouze okrajově na základě preferencí studentů a s přesvědčením, že student nedokáže pojmout dva programovací jazyky během 14 pracovních dní. Obzvláště pokud student začíná a i samotné HTML je pro studenta novinkou.

#### 4.1 Cíloví uživatelé

Během 3 let vyučování rekvalifikačních kurzů Tvorba www stránek jsem si jako lektor rozdělil studenty do 3 skupin podle zájmu, motivace a zkušeností.

Pro jednotky studentů by bylo lepší vytvořit samostatné skupiny. Například lidé, kteří nemají zájem si najít zaměstnání a pouze se snaží vyplnit čas než

půjdou do důchodu, nebo se snaží získat díky absolvování rekvalifikačního kurzu příspěvky. Na kurzech Tvorba www stránek jsem toto zažil pouze jedenkrát. Na rekvalifikačních kurzech Obsluha osobního počítače jich bylo přibližně 15 %.

Všiml jsem si, že velkým problémem mezi studenty je neschopnost formulovat dotaz. Tento problém vnímají i další lektori ve společnosti MAVO, Czechitas, nebo cvičící předmětu Testování softwaru na ČVUT. Nezáleží na tom, zda se jedná o úplného začátečníka, nebo pokročilého. Až na pár výjimek si studenti často nejsou jistí, zda jejich dotazy nejsou „hloupé“, nebo se ve snaze přijít na řešení samostatně neptají vůbec.

Při hlubších rozhovorech se studenty jsem se dozvěděl několik důvodů. Někteří studenti se potýkají s nízkým sebevědomím. Bojí se ptát kvůli strachu, co si o nich pomyslí ostatní studenti a nebo lektor. Další obavy lze charakterizovat otázkami: „*To už bych se ptala moc často*“. I když například Czechitas má motto „*Žádná otázka není hloupá*“, studentky často nevyužívají kapacity lektorů a koučů pro zodpovězení dotazů nebo pomoci.<sup>1</sup>

#### ■ Motivovaní s cílem nového zaměstnání

Čtvrtina studentů je motivovaná najít si nové zaměstnání v oboru. Většinou mají v oboru základní zkušenosti. Ví, co je HTML a mají obecnou představu o struktuře webových stránek. Ví, že existuje server a že se webová stránka zobrazuje v klientově počítači díky prohlížeči. Mají čas a prostor se učit.

#### ■ Se zájmem vytvořit si vlastní webové stránky

Lidé s podnikatelským zájmem, matky po mateřské dovolené, nebo drobní podnikatelé, přichází na kurz s cílem vytvořit si vlastní webové stránky. Tato skupina čítá polovinu studentů. Nemají vzdělání v oboru. Mají vzdělání v jiném oboru, nebo víceletou zkušenost v podnikání. Chtějí dále pokračovat v původním oboru a zároveň si chtějí rozšířit své znalosti. Jejich zázemí je zpravidla ruší od vzdělávání se ve tvorbě webových stránek, ale jejich vůle dojít k cíli je podporuje a pomáhá jim překážky překonat.

#### ■ Bez zkušeností

Poslední čtverno skupinou jsou lidé v oboru neznalí. Bez konkrétní motivace. Jejich zázemí není přizpůsobené pro vzdělávání. Polovina studentů je z chudších poměrů, nebo nevlastní počítač. Je to i jediná skupina, ze které pár jednotek studentů za mou praxi kurz nedokončilo.

<sup>1</sup>Na kurzu Digitální Akademie: Web je na každé programovací lekci jeden lektor a 3 až 5 koučů, kteří tam jsou pouze za účelem zodpovídání dotazů a individuální výpomoci studentkám.

## 4.2 Požadavky na platformu

Cílem je maximálně podpořit studenty, kteří si chtějí najít novou práci a neodradit zbytek studentů od hlubšího studia webových technologií. V ideálním případě motivovat část studentů s chutí vytvořit si své vlastní webové stránky ke hledání nové práce v oboru. V poslední řadě je cílem usnadnit průchod kurzem nerozhodným a představit tvorbu www stránek v nejlepším duchu.

Každá skupina má své preference, zkušenosti a cíle. Když se mi v kurzu sešli studenti s velkými rozdíly zdatnosti, snažil jsem se zadávat nadprůměrně schopným samostatnou práci. Zároveň jsem se snažil přednášet tempem těch nejpomalejších. I zde jsem postrádal platformu, která by podpořila individuální přístup.

### 4.2.1 Webový kód editor

Hlavní přidanou hodnotou platformy a klíčovou částí je kód editor. Student dostane zadání v podobě naformátovaného textu a předem připraveného kódu.

Student bude podle zadání kódovat webovou komponentu, nebo celou webovou stránku v HTML, CSS a JavaScriptu. Zároveň bude mít možnost naprogramovat základní konstrukty v PHP. Editor bude zbarvovat části kódu pro lepší přehlednost a našeptávat možnosti pro danou technologii.

Diagram užití na obrázku 4.1 shrnuje a vizualizuje navržené schopnosti editoru.

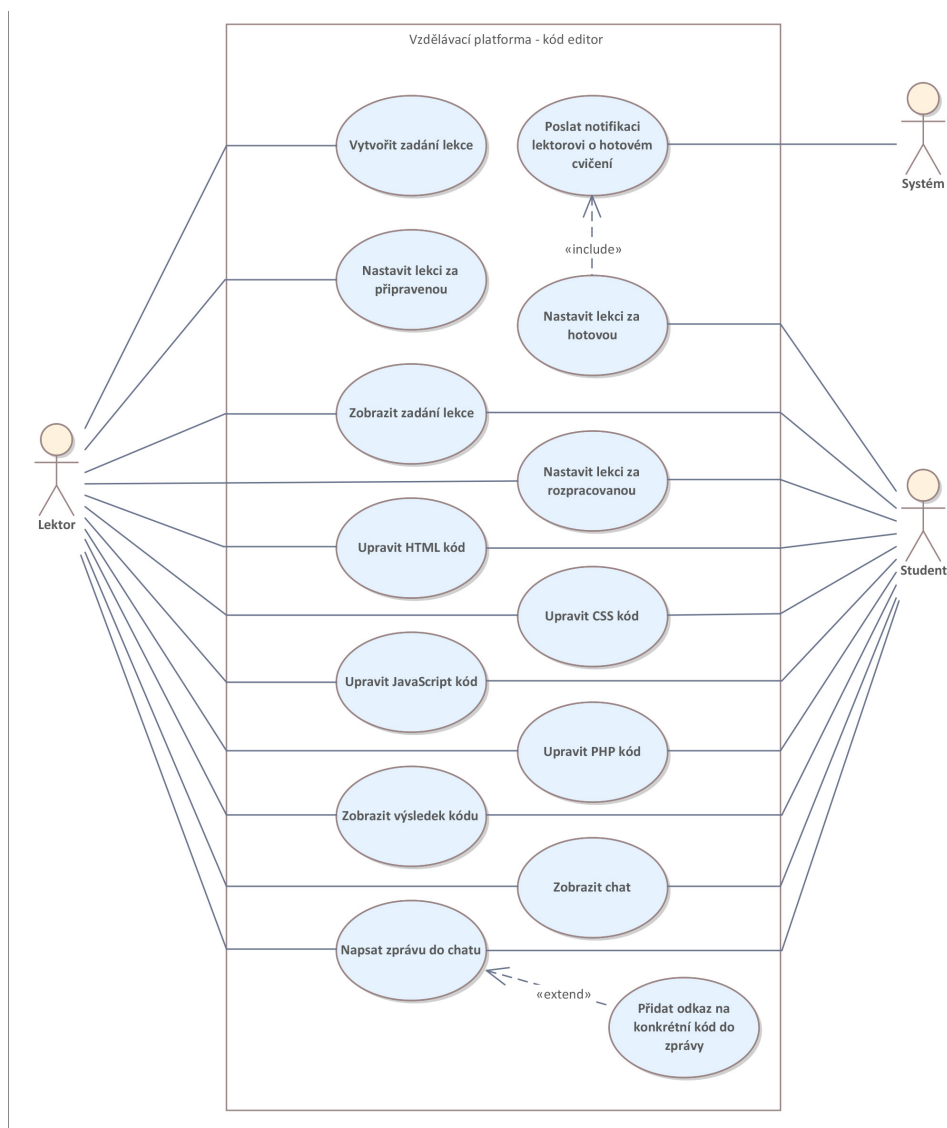
V případě nesnázi studenta se lektor může připojit ke studentovu řešení. Student uvidí kurzor lektora a lektor uvidí studentův kurzor. V reálném čase budou moci upravovat kód.<sup>2</sup> Pro možnosti soukromé komunikace a podporu méně mluvných studentů obsahuje stránka s editorem chat. Student a lektor si mohou dopisovat a řešit detaily ohledně konkrétního zadání. Každá lekce má vlastní chat. Pro možnost komunikace nad konkrétní částí kódu je možné označit kus kódu a referenci poslat skrze chat, aby si adresát mohl uvedenou část kódu zvýraznit.

V poslední řadě zabírá studentům mnoho času ověřit si zda to, co vytvořili, odpovídá jejich očekávání. Na úplných začátcích, když člověk nezná Node.js, se využívá buď Visual Studio Codu rozšíření, nebo se musí dokument ukládat do počítače. Následně otevřít dokument v Průzkumníku souborů a nakonec otevřít HTML dokument v prohlížeči. Z těchto důvodů je součástí platformy zároveň náhled aktuálního řešení.

### 4.2.2 Platforma

Nejde jen o podporu studentů, ale také o úsporu času lektorů, zefektivnění jejich výuky a usnadnění celkové režie kurzů. Platforma obsahuje možnost administrace jednotlivých kurzů. Standardně lektor vede jeden a ten samý kurz několikrát do roka. Je zbytečné, aby lektor vše musel vždy znovu zapisovat

<sup>2</sup>Tato vlastnost byla inspirována primárně Covidovou dobou.



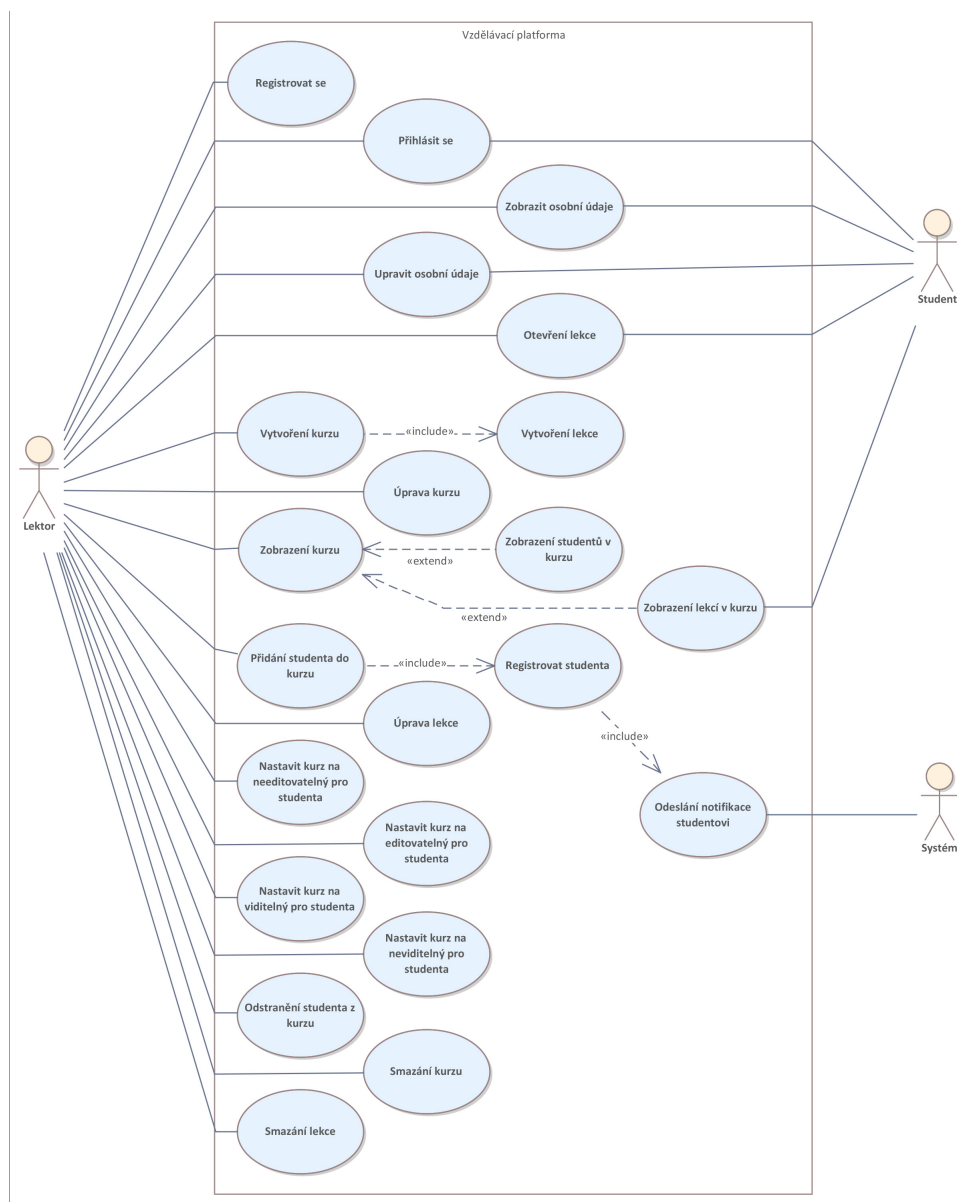
**Obrázek 4.1:** Diagram užití webového kód editoru

od nuly. Proto je při vytváření kurzu také možné vytvořit kurz dle šablony existujícího kurzu.

Po založení kurzu včetně lekcí může lektor přidat studenty. Platforma má sloučit studenty výlučně pro účely kurzů vedených lektorem, proto bude lektor přidávat studenty do platformy sám. Tím se eliminuje možnost přihlášení se od kohokoliv bez oprávnění. Studentům následně přijde e-mail s url adresou platformy.

Diagram užití na obrázku 4.2 vizualizuje všechny navržené funkcionality samotné platformy.

Lektor může ovlivňovat, jaké lekce kteří studenti uvidí a které lekce mohou vyplňovat. Platforma se snaží pouze doplňovat živou výuku, cílem platformy není implementovat vzdělávací materiály. To může být předmětem budoucích



Obrázek 4.2: Diagram užití platformy

rozšíření. Jsou zde pouze cvičení k vyplnění. Studenti mohou na cvičení bez omezení pracovat. Lektori mohou kdykoliv nahlížet do kódu studentů bez jakékoliv notifikace studenta. Student může lektora zpozorovat pouze pokud sám bude chtít něco editovat. V editoru se zobrazí kurzor lektora.

V poslední řadě studenti mohou označit cvičení za dokončené. V tu chvíli se odešle notifikace na e-mail lektora, aby se mohl na cvičení podívat a případně označit cvičení za nedokončené, nebo si cvičení s žákem osobně projít.





## ■ React

Knihovna napsaná v JavaScriptu vydaná pod volnou licencí 29. března 2013 společností Meta urychluje a zjednodušuje vývoj uživatelského rozhraní webových aplikací. Obsahuje základní a často opakující se návrhové vzory. Díky virtuálnímu DOMu<sup>3</sup> dokáže React rychle vykreslovat stránky. Platforma je postavena na verzi 18.2.0.[60]

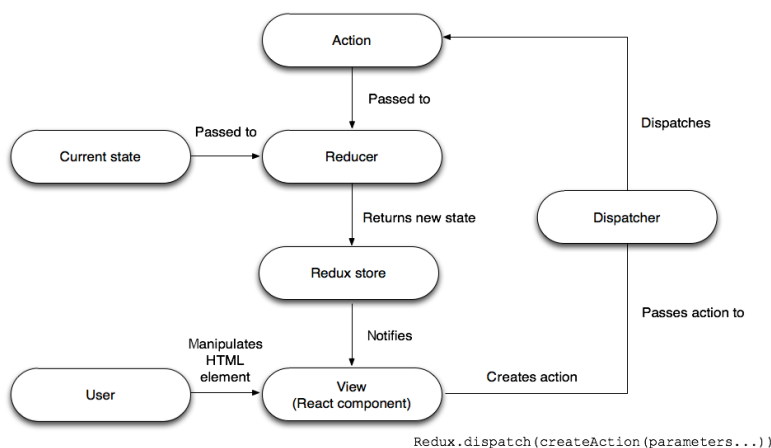
## ■ Redux

Pro účely centrální správy stavu dat aplikace je využita knihovna Redux. Dan Abramov a Andrew Clark ji vydali pod volnou licencí 2. června 2015. Jedná se o specializovanou knihovnu pro management stavu aplikace. Jedná se o jeden z nejkomplicovanějších aspektů webového vývoje. Knihovna zjednodušeně implementuje architekturu Facebookového Fluxu.[61]

Knihovna funguje na základě třech principů:

- Aplikace drží stav v jediném objektu
- Stav aplikace je needitovatelný a lze ho změnit pouze pomocí specializovaných actions funkcí, která změnu iniciuje
- Reducers funkce, které mají za úkol vytvořit nový stav na základě podnětu actions funkcí.

Aplikace principů a tok dat je znázorněn na obrázku 4.3. Uživatel manipuluje s HTML elementy. Vzniká akce, která posílá popud ke změně stavu. V Reducer funkci z aktuální stavu a nových dat vzniká nový stav uložený v Redux úložišti.



**Obrázek 4.3:** Struktura Reduxu - převzato[1]

<sup>3</sup>Domein Object Model je pojem popisující strukturu webové stránky jak ji vnímá JavaScript.

## ■ Ant design

Pro efektivní vývoj uživatelského rozhraní je využito knihovny specializované na Reactové komponenty – Ant designu. Vznikla 21. června 2015 od společnosti Alibaba pod volnou licencí. Velkou výhodou knihovny jsou rozsáhlé a volně přístupné komponenty s širokou škálou přizpůsobení. Platforma využívá verze 5, která opustila zpracování kaskádových stylů pomocí dynamického jazyka Less a začala využívat ke generování kaskádových stylů přímo JavaScript.[62]

Na obrázku 4.4 je ukázka komponenty využívané pro vizualizaci načítání dat z knihovny využívané v platformě. Knihovna umožňuje úpravu komponenty pomocí vlastních atributů.



**Obrázek 4.4:** Screenshot ukázky variability Ant design komponenty pro načítání stránky využívané v platformě

## ■ Styled Components

Knihovna umožňuje psát CSS v JavaScriptu. Vznikla 15. listopadu 2016 od vývojářů Glenna Maddernema a Maximiliana Stoibera. Zvolena je pro účely tvorby vlastních komponent. Je možné vytvářet úplně nové a nebo dědit z vzniklých komponent Ant designu a přidávat vlastní styly. Knihovna má následující výhody[63]:

- Vkládá do stránky pouze styly, které jsou pro stránku potřeba a žádné jiné

- Generuje unikátní názvy tříd, díky kterým nemůže dojít ke konfliktům mezi jmény
- Jednoduché dynamické stylování
- Jednoduchá údržba kaskádových stylů

Na obrázku 4.5 je znázorněno využití knihovny v platformě. Komponenty mohou mít definované proměnné podle kterých lze vykreslovat nebo vypočítávat chování na základě různých vstupů.

```
export const Wrapper = styled.div<{ $type: PositionType }>`
display: flex;
flex-direction: column;
/* flex: 1 1 21.25rem; */
position: relative;
row-gap: 1rem;
padding: 1rem;
padding-bottom: 1.5rem;
width: 80%;
box-shadow: var(--box-shadow);
border-radius: ${BORDER_RADIUS};
transition: 0.3s;

${(props) =>
  props.$type === PositionType.left ?
  css`
    margin-right: 20%;
    border: 1px solid var(--primary-color);
    : css`
    margin-left: 20%;
  `
}

[data-theme='dark'] & {
  background: var(--secondary-background-color);
}

&:hover {
  box-shadow: var(--box-shadow-hover);
}
`;
```

Obrázek 4.5: Screenshot využití Styled Componets v platformě

## CodeMirror

Klíčová komponenta celé platformy je knihovna, obsluhující samotný kód editor. Oproti knihovně CodeMirror stála knihovna Monaco Editor, která je integrována ve Visual Studio Code.

Monaco Editor je integrovatelný s Reactem a sám o sobě má mnoho rozšíření, která dokáží obsáhnout potřebnou funkcionalitu. Má dobrou dokumentaci a má na GitHubu 35 tisíc hvězdiček.[\[64\]](#)

CodeMirror je obecnou knihovnou napsanou v JavaScriptu, obsahující mnoho rozšíření a cílí na jednoduchou rozšiřitelnost. Historie knihovny sahá do roku 2007, kdy vznikla první verze. Obsáhlou dokumentaci pouze komplikuje velký rozdíl mezi verzí 5 a 6. K poslední verzi na internetu není mnoho návodů. I když má na GitHubu méně hvězdiček je její stahovanost dle NPM trends za poslední rok trojnásobně větší. Je to i z toho důvodu, že knihovna je rozdělena do více modulů a každý má vlastní repozitář.[\[65\]](#)

Na základě hlubšího průzkumu a vyzkoušení obou knihoven jsem se rozhodl pro CodeMirror. Lze jednodušeji implementovat sdílení kurzorů mezi lektorem a studentem. Zároveň je obecně oproti Monaco editoru jednodušeji rozšiřitelná.

Na obrázku 4.6 je zachyceno využití knihovny v platformě. Aktivní je editor pro HTML a je vidět barevné odlišení značek od atributů a hodnot atributů.

```

HTML CSS JavaScript PHP
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Nová houba</title>
7 <link rel="stylesheet" href="styles.css">
8 </head>
9 <body>
10 <h1>Obohaťte náš atlas</h1>
11 <!-- Zde začnete vytvářet kód -->
12
13 <!-- Zde končí vlastní kód -->
14 <script src="index.js"></script>
15 </body>
16 </html>
17

```

Obrázek 4.6: Screenshot CodeMirror editoru z platformy

## ■ Quill

Pro účely komplexnějšího zadávání úkolu studentovi jsem se rozhodl vybrat knihovnu podporující všechny základní funkce textového editoru. Vybíral jsem z editorů uvedených níže.

TinyMCE je jedním z dlouho využívaných a oblíbených editorů s 13 tisíci hvězdičkami na GitHubu a s desítkami rozšíření. Podporuje velké množství funkcionalit. Nevýhodou je robustnost řešení. Podle platformy bundlephobia je Quill o 69 % menší balíček.<sup>[66]</sup>

Draft knihovna poskytující stavební bloky pro tvorbu editoru je dobrým řešením v případě potřeby vlastního editoru. Na GitHubu má 22 a půl tisíce hvězdiček. Má dobrou dokumentaci a je integrovatelná s Reactem.<sup>[67]</sup>

ProseMirror vytvořil stejný autor, který vytvořil knihovnu CodeMirror. Je také dělena do více modulů a obsahuje rozsáhlé pole možností vlastního rozšíření. Pro účely platformy vyžaduje mnoho práce a designu, aby vše fungovalo.<sup>[68]</sup>

Quill textový editor, používaný platformou LinkedIn nebo v komunikačním nástroji Slack, s 36 a půl tisíci hvězdičkami na GitHubu, odpovídá požadavkům platformy. Je integrovatelný s Reactem. Nabízí kompletní uživatelské rozhraní a snadné uložení výstupu do databáze. Zároveň má kvalitní dokumentaci a mnoho příkladů. První verze vyšla 7. března 2014.<sup>[69]</sup>

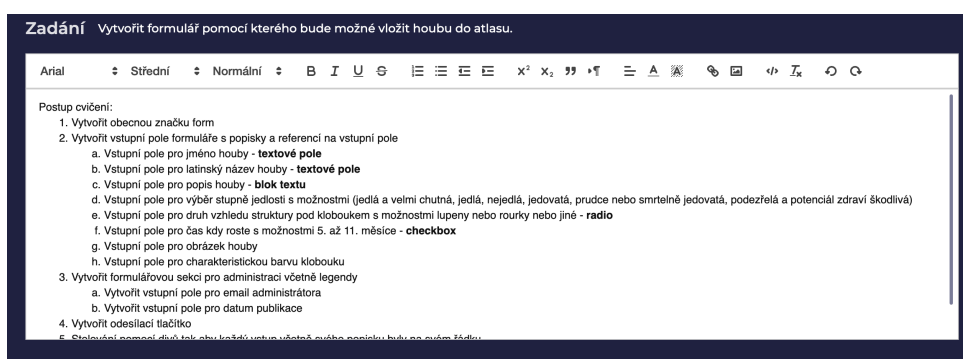
Na obrázku 4.7 je zachyceno využití knihovny v platformě včetně ukázkového vyplnění zadání.

### ■ 4.3.2 Serverová část

Rozhodoval jsem se primárně mezi dvěma technologiemi. Buď použít framework Symfony napsaný v PHP, a nebo použít knihovnu Feathers.js napsanou v JavaScriptu a na základě toho pro backend použít Node.js. Jde i o cenu a místo hostingu. Pro Symfony je možné využít webhosting v České republice. PHP servery jsou cenově dostupné a lze sehnat webhosting například u společností<sup>4</sup>:

- Můj cloud za 40 Kč/měsíc<sup>[70]</sup>

<sup>4</sup>Všechny ceny jsou uváděné včetně DPH



Obrázek 4.7: Screenshot Quill editoru z platformy

- Wedos za 50 Kč/měsíc[71]
- eBola.cz za 55 Kč/měsíc[72]
- Active24.cz za 110 Kč/měsíc[73]
- Český hosting za 140 Kč/měsíc[74]

Na druhé straně stojí cloudové řešení. Služby vyžadují obsáhlejší konfigurace oproti nasazení PHP. Buď se konfigurují Docker konteinery, nebo služby mají vlastní způsoby, jak nastavit webhosting. Poskytovateli jsou například<sup>5</sup>:

- DigitalOcean za 450 Kč/měsíc[75]
- Render za 600 Kč/měsíc[76]
- Heroku za 700 Kč/měsíc[77]
- Azure za 6 000 Kč/měsíc[78]<sup>6</sup>
- AWS za 8 000 Kč/měsíc[79]<sup>7</sup>

Rozhodl jsem se pro cloudové řešení. Poskytuje větší variabilitu ve výběru technologií. Díky integraci s repositáři lze automaticky nasazovat novější verzi po vytvoření kódu. Zároveň cloudový poskytovatelé mají lépe vedené dokumentace.

<sup>5</sup>Ceny jsou uvedené přibližně s ohledem na jeden kurz s 80 vyučujícími hodinami pro 15 studentů. Cena zahrnuje hosting i databázi

<sup>6</sup>při volbě App Platform

<sup>7</sup>Volba Amazon EC2

## ■ Node.js

Vznikl 27. března 2009 pod volnou licencí. Node.js je multiplatformní běhové prostředí postavené na enginu V8 pro JavaScript tvořící jádro prohlížeče Chrome. Je podobný technologiím Event Machine v jazyce Ruby a Twisted v jazyce Python. Představuje smyčku událostí<sup>8</sup> jako běžnou funkci a ne jako knihovnu. Díky tomu je rychlejší.

Node.js využívá pouze jednoho vlákna. Asynchronní volání je realizováno pomocí výše zmíněných událostí. Neukládá žádná data. Pouze data posílá.[80]

## ■ Feathers.js

Stejně jako pro PHP Symfony, pro Javu Springboot jsem chtěl pro JavaScript použít framework implementující MVC architekturu serverové části. Velká výhoda Feathers.js je rychlý a jednoduchý vývoj jednotlivých endpointů. Po vygenerování služby se automaticky vytvoří implementace pro GET, POST, PATCH a DELETE HTTP metody. Má kvalitní dokumentaci včetně velkého množství příkladů na internetu. Na GitHubu má 14 a půl tisíce hvězdiček a týdně se stáhne více než 55 tisícrát z npm.[81]

## ■ WebSocket knihovna pro Node.js

Jednoduchá, rychlá a důkladně otestovaná knihovna pro klientskou i serverovou implementaci WebSocketu. Knihovna vydaná pod volnou licencí má 20,2 tisíce hvězdiček na GitHubu. Knihovna umožňuje komunikaci v reálném čase mezi klienty.[82]

## ■ ShareDB

Kompletní knihovna vydána pod otevřenou licencí pro spolupráci více klientů nad dokumenty ve formátu JSON<sup>9</sup> v reálném čase. Poskytuje pro server Node.js koordinaci a odesílání úprav. Zároveň obsahuje klientskou část napsanou v JavaScriptu pro manipulaci s dokumenty, kterou lze spustit buď v Node.js nebo v prohlížeči. Základní řízení konfliktů je řešeno prostřednictvím provozní transformace (OT)<sup>10</sup>. Je použita v DerbyJS frameworku.[85]

### ■ 4.3.3 Databáze

Platforma potřebuje primárně ukládat data o kurzech, lekcích a studentech. Všechna tato data lze uložit do relačního schématu.

---

<sup>8</sup>Event loop

<sup>9</sup>JSON(JavaScript Object Notation) Formát výměny dat definován v ECMA-404[83]

<sup>10</sup>Technologie řešení konkurenčního přístupu k jedinému dokumentu. Základní myšlenkou OT je transformovat parametry editační operace podle účinků dříve provedených souběžných operací.[84]

## ■ PostgreSQL

Primárně jsem se rozhodl mezi MySQL a PostgreSQL. Obě jsou relační databáze splňující základní požadavky na ukládání dat platformy. PHP hostingy zmíněné v kapitole výše nabízejí v ceně MySQL. Oproti tomu cloudové společnosti nabízejí buď pouze PostgreSQL a nebo obojí. Azure a AWS nabízejí vlastní SQL databáze.

PostgreSQL je oproti MySQL plně ACID.<sup>11</sup> Další výhodou je podpora MVCC<sup>12</sup>, díky čemuž může více uživatelů přistupovat k datům a měnit je bez ohrožení integrity.[86]

## ■ GitHub

Kód je pouze text. Bylo by možné ho ukládat do relační databáze. Toto řešení konzervuje kód uvnitř platformy a nebylo by možné si ho pohodlně uložit, nebo využít k jiným účelům. To je v přímém rozporu s vizí platformy.

Pro účely ukládání kódu platforma využívá externí verzovací systém Git. Hlavními kandidáty jsou Bitbucket od společnosti Atlassian a GitHub vlastněný Microsoftem. Bitbucket je primárně placenou službou, ale umožňuje zdarma vést soukromé repozitáře. Atlassian vlastní také Jiru a další nástroje, které jsou mezi sebou integrované.[87] Oproti tomu GitHub umožňuje zdarma vést privátní i veřejné repozitáře. V tuto chvíli obsahuje přes 330 milionů repozitářů a čítá přes 100 milionů uživatelů.[88] GitHub má otevřené API<sup>13</sup> jak RESTové tak GraphQL, které šetří množství přenesených dat a je variabilnější, jak je uvedeno v dokumentaci.

Díky popularitě zdarma vedených veřejných repozitářů a dobré dokumentaci API jsem se rozhodl pro GitHub.

## ■ 4.4 Architektura platformy

Na základě požadavků je platforma webovou aplikací. Serverová část je hostovaná v cloudu a databáze je využita dle nejlepší cenové nabídky. Feathers.js lze spojit s množstvím databází, ale je preferovaná PostgreSQL.

Celá aplikace je napsána v JavaScriptu. Mezi serverovou a klientskou částí je několik komunikačních kanálů. Primárně se data posílají přes REST rozhraní v JSON formátu. Tímto způsobem se posílají veškerá data související s uživateli a administrativní úkony lektora. Pro účely sdílení kurzoru a aktuálního stavu kódu je využit duplexní komunikační kanál (socket). Druhý socket je vystaven pro chat.

<sup>11</sup>ACID = Atomicita, konzistence, izolace a trvanlivost

<sup>12</sup>Multiversion concurrency control je pokročilá funkce databáze vytvářet kopie záznamů pro bezpečné čtení a zápis.

<sup>13</sup>Application programming interface = rozhraní pro volání služeb z venčí aplikace

Platforma je určena pro počítače s monitorem. Hlavním důvodem je nutnost velké obrazovky, která je pro pohodlné kódování klíčová. Platforma je uzpůsobena i pro dotykové displeje.

Vývoj je cílen na poslední verze prohlížečů Chrome a Firefox. Chromium vykreslovací jádro stejně jako Chrome využívají i MS Edge a Opera, které jsou také podporovány.

## 4.5 Uživatelské rozhraní

Kvalitní uživatelské prostředí a jeho přívětivost je klíčem pro úspěch produktu. V dnešní rychlé době se stává tvorba kvalitního uživatelského rozhraní ještě větší výzvou. Důležité je, aby design vznikal na základě průzkumu a pečlivého testování. Jedině tak můžeme dosáhnout lepších výsledků.<sup>[89]</sup>

Platforma bude mít v první fázi minimalistický design s účelem ověření, jestli vize podpory výuky jde dobrým směrem. Barvy, fonty a mezery budou využity ze základního nastavení tématu Ant design knihovny. Po ověření vhodnosti platformy pro podporu výuky se upře pozornost k vytvoření vizuální identity a úpravě uživatelského prostředí. Zároveň se na základě nasbírané zpětné vazby navrhnou úpravy v souladu s vylepšením uživatelské přívětivosti.

Vzdělávací platforma má dvě hlavní části. První je rozhraní pro správu kurzu nebo kurzů a druhá je samotný webový kód editor, který je klíčovou přidanou hodnotou celé platformy.

Platforma musí umět registraci, přihlášení, náhled profilu a změnu hesla pro lektora i studenta. Lektor musí mít možnost vytvořit kurz, kde vytvoří lekce a zároveň přidá studenty.

Editor musí obsahovat možnost vytvořit zadání a následně ho zobrazovat studentovi. Zobrazit přehledně všechny soubory, které bude moci student upravovat. Jedná se o soubory index.html, style.css, index.js a index.php. Součástí obrazovky musí být i panel pro chat a náhled výsledku.

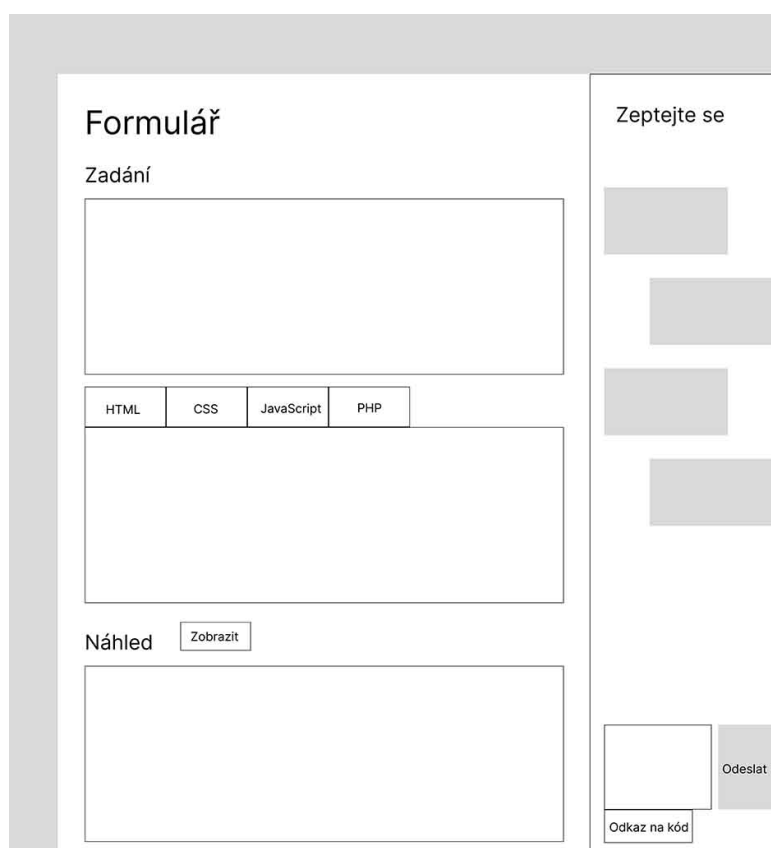
Možný návrh rozhraní editoru je načrtnutý pomocí drátěného modelu na obrázku 4.8. Pro přehlednost jsou zadání, kód editory a náhled pod sebou. Jednotlivé editory je pak možné mezi sebou překlikávat. Není možné mít vedle sebe dva editory najednou. Panel pro chat je na pravé straně, protože levý panel už je zabraný rozcestníkem mezi kurzy. Přijaté zprávy se zarovnávají doleva a odeslané doprava, stejně, jak je tomu v chatovacích platformách, jako je Messenger, Instagram nebo WhatsApp.

### 4.5.1 Tvorba kurzu

Tvorba kurzu bude rozdělena do tří kroků:

1. Zadání obecných informací ke kurzu
2. Základní zadání lekcí
3. Shrnutí všech informací





**Obrázek 4.8:** Drátěný model webového kód editoru

V prvním kroku se budou zadávat obecné informace ke kurzu. Zároveň bude možné ze všech existujících kurzů vybrat šablonu, díky které se překopírují jednotlivé lekce, aby je lektor nemusel vždy zadávat od začátku.<sup>14</sup>

Další část se věnuje základnímu zadání lekcí. Jde pouze o zadání jmen a obecných podnadpisů lekcí. Podrobné zadání lekcí probíhá po vytvoření kurzu. Tvorba zadání je často časově náročná. Zároveň se mohou stejná témata vykládat a cvičit na různých příkladech. Osobně se snažím vést kurzy tematicky. Jednou studenti mají za úkol vytvářet webové komponenty pro kuchařku, jindy pro autosalon.

V poslední části se věnuji shrnutí informací. Je možné se vrátit zpět a data editovat.

<sup>14</sup>Kurzy jsou akreditované na 3 roky a je nutné držet se osnovy po celou dobu akreditace, která mimo jiné obsahuje osnovu.

### 4.5.2 Administrace kurzu

Každému, kdo se do platformy sám za sebe zaregistruje, je přiřazena lektorská role. Uživatel může vytvářet libovolné množství kurzů. Na úvodní stránce uživatel vidí notifikace z chatů studentů ze všech kurzů. Zároveň se kurzy přidávají do levého bočního panelu pod sebe viz ukázka 4.10. Zároveň uživatel může editovat svůj profil.



Obrázek 4.9: Drátěný model administrace platformy

Po vytvoření kurzu je možné si otevřít přes boční panel administrační stránku kurzu, kde je možné spravovat lekce a studenty kurzu. Návrh obecné struktury administračního rozhraní je znázorněn drátěným modelem na obrázku 4.9. Při zobrazení jednotlivých stránek lekcí může lektor spravovat viditelnost a editovatelnost pro jednotlivé studenty zvlášť. Tato funkcionality je pro individuální přístup ke studentům. Tímto způsobem lektor může odkrývat bonusové lekce jen pro vybrané studenty.

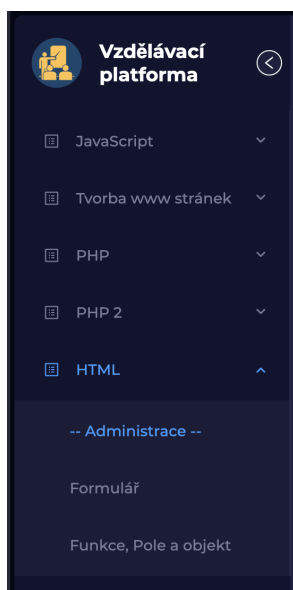
Účet studenta je vázán ke konkrétnímu kurzu. Při přihlášení má student možnost se z úvodní stránky prokliknout přímo do konkrétního cvičení. Zároveň může editovat svůj profil.

### Řízení lekcí a přístupů










Data studentů a lekcí jsou pro přehlednost vizualizovaná v tabulkách. Tabulka se studenty existuje ve dvou typech. Jeden typ se zobrazuje lektorovi v administraci ve správě studentů a druhý v detailu lekce. Stejně existují tabulky lekcí dvou typů. Jeden typ se zobrazuje lektorovi v administraci, kde řídí přípravu lekcí a druhý se zobrazuje studentovi.

Je důležité zůstat v zobrazování dat konzistentní nejen v rámci jedné stránky, ale i napříč celou aplikací. Zároveň platforma využívá pro vizualizaci stavu ikony. Cílem je zvolit ikony tak, aby významem odpovídaly i využití mimo platformu. To výrazně podporuje intuitivnost platformy.[90]

Použité ikony a jejich význam je shrnut v následující tabulce:



**Obrázek 4.10:** Screenshot bočního panelu z pohledu lektora

Význam	Ikona
Viditelnost	
Neviditelnost	
Editovatelnost	
Needitovatelnost	
Nepřipravené zadání	
Připravené cvičení	
Nezačaté cvičení	
Rozpracované cvičení	
Hotové cvičení	

**Tabulka 4.1:** Seznam využitých ikon a jejich významů

### ■ 4.5.3 Lekce

Každá lekce obsahuje stejnou obrazovku. Rozdíl je v tom, jestli si lekci otevírá lektor jako editor, a nebo jestli je lekce otevírána k vyřešení. V druhém případě ji lektor uvidí stejně jako student jen s rozdílem, že bude psát do chatu sám za sebe.

Platforma mimo PHP neobsahuje vlastní konzolový náhled. Je to primárně kvůli cíli naučit studenta pracovat s konzolí prohlížeče. Je důležité studenta připravit na reálné podmínky, kdy pro ladění svého kódu bude primárně používat konzoli prohlížeče.

### ■ Zadaná cvičení

Při vytváření cvičení stránka obsahuje textový editor, ve kterém lektor sepíše zadání. Je možné vytvářet nadpisy, odstavce, seznamy číselné i bodové a vkládat obrázky. Text je možné ztučnit, zkosit nebo podtrhnout.

Obsahuje okno pro webový editor s možností předvyplnění kteréhokoliv z podporovaných souborů. V poslední řadě obsahuje okno, kde si lektor může zobrazit, jak vypadá výsledek připraveného kódu. Okno se aktualizuje na popud tlačítka.

Když má lektor vše připravené, může kliknout na tlačítko „Lekce je připravena“. Tím se cvičení nastaví na připravené a je možné ho zadat studentům k vyplnění.

### ■ Vyplněná cvičení

Oproti stránce pro zadání cvičení je stránka určená k vyplnění cvičení obohacena o chatovací pravý panel. Zde si lektor a student mezi sebou mohou dopisovat v reálném čase.

Pokud má student cvičení hotové, může kliknout na tlačítko „Lekci mám hotovou“. Tím se lekce nastaví na hotovou a lektorovi je odeslána notifikace na e-mail.

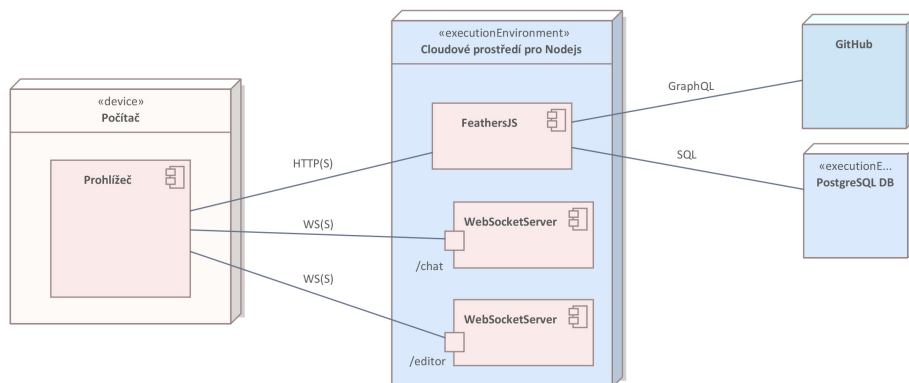
# Kapitola 5

## Implementace

V této kapitole podrobně popisují způsob implementace všech funkcionalit vzdělávací platformy a zároveň se věnují i popisu jejich funkčnosti.

### 5.1 Vzdělávací platforma

Pro vývoj webového kód editoru bylo třeba jako první vytvořit prostředí, ve kterém se bude editor nacházet. Na základě návrhu z předešlé kapitoly je platforma navržena jako klient-server aplikace. Všechny její části jsou napsané v TypeScriptu. Pro účely hostování aplikace je využito poskytovatele cloudového prostředí Render, který má možnost bezplatného plánu obsahující i PostgreSQL databázi.



Obrázek 5.1: Diagram nasazení platformy

Základní rozdělení a technologie jsou znázorněny na UML diagramu nasazení 5.1. V platformě je využíván TypeScript jak v serverové, tak klientské části. Server využívá frameworku Feathers.js pro tvorbu REST API. Přes toto rozhraní probíhá většina datových toků. Výjimkou je kód editor, který je posílán přes WebSocket. Stejně tak jsou posílány zprávy chatu pro dosažení komunikace v reálném čase.

Pro účely přenosu kódu včetně kurzoru je využito obalení zpráv pomocí WebSocketJSONStream, který využívá knihovna ShareDB. Technologie vyžaduje celý socket, proto pro chat je nutné vytvořit druhý socket.

Dle návrhu pro integraci s GitHub repozitářem je vhodnější komunikace pomocí dotazovacího jazyka GraphQL. Hlavní výhodou je možnost dotazování pouze na data, která platforma potřebuje.

### ■ 5.1.1 Databáze

Data nemají grafovou strukturu a není třeba využívat volnější dokumentové struktury, proto lze uložit v relačním schématu. 5.2 Platforma obsahuje entity:

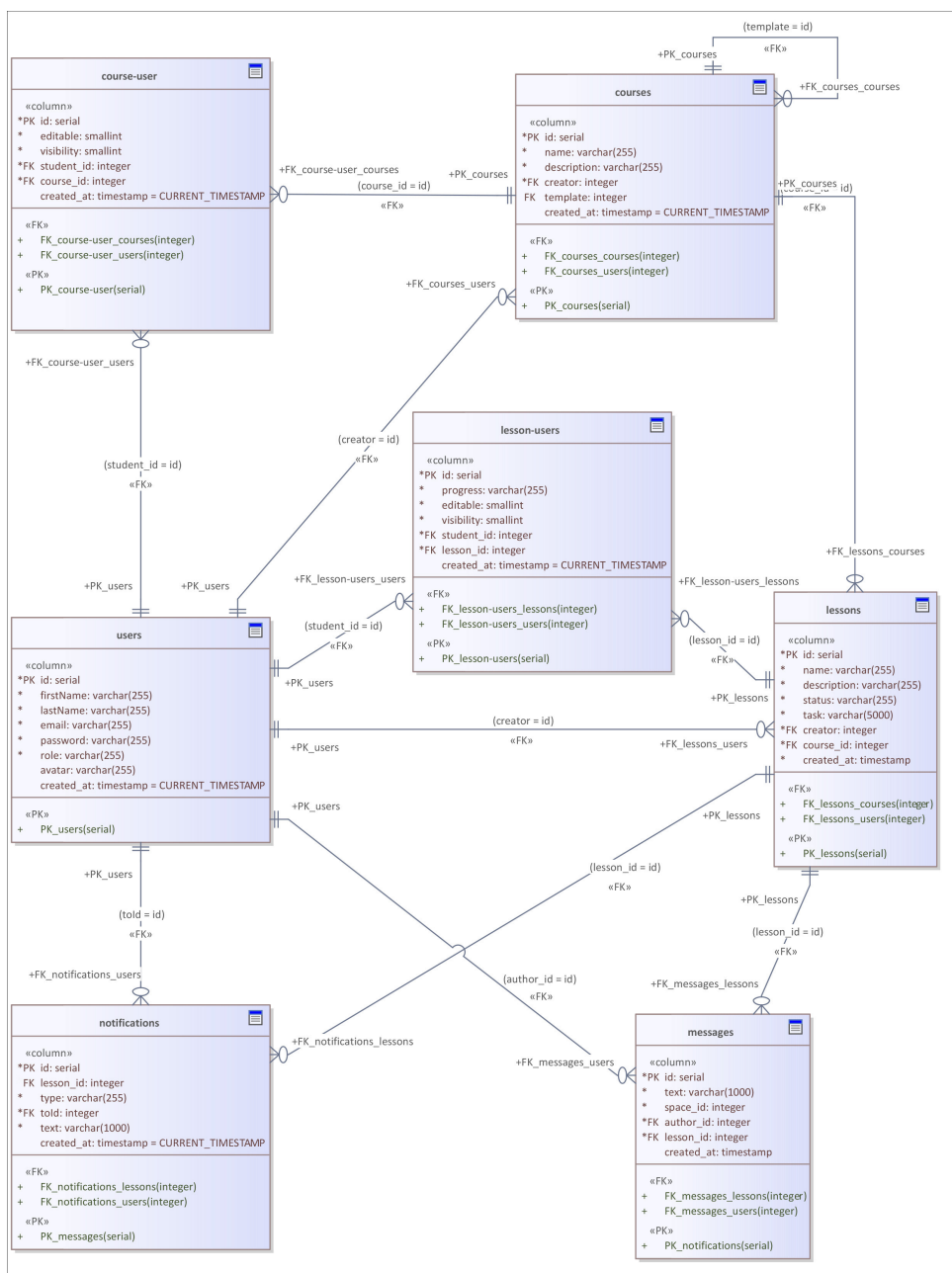
- Uživatel
- Kurz
- Lekce
- Zpráva
- Notifikace

Uživatel je tvůrcem libovolného množství kurzů i lekcí. Kurz i lekci může vytvořit vždy pouze jeden uživatel. Každý kurz má své lekce. Kurz může obsahovat libovolné množství lekcí. Zároveň každá lekce náleží právě jednomu kurzu. Notifikace je vytvořena vždy pro konkrétního uživatele s možností udání právě jedné lekce. Ke každému uživateli může být navázáno libovolné množství notifikací. K libovolné lekci může být navázáno libovolné množství notifikací. Zprávy jsou vždy vázány ke konkrétnímu uživateli a konkrétní lekci. Uživatel může poslat libovolné množství zpráv a v lekci může být neomezeně dlouhé chatovací vlákno.

Uživatel může být účastníkem více kurzů a zároveň jeden kurz může mít více účastníků. Uživatelé kurzy nemusí pouze vytvářet, ale mohou být přiřazeni jako řešitelé libovolného množství lekcí. Současně libovolné množství uživatelů může být řešiteli jedné lekce.

Na základě vazeb mezi entitami vzniklo relační schéma včetně tabulek reprezentujících M:N vazby. Datové typy jsou určeny tak, aby obsahovaly dostatečné rezervy. Atributy, které by mohly být realizovány výčtem hodnot jsou reprezentovány textovým řetězcem pro účely jednodušší rozšiřitelnosti. Každá tabulka automaticky obsahuje, pro účely ladění platformy, atribut „*created\_ad*“, plněný datem vzniku záznamu.

Atributy reprezentující editovatelnost a viditelnost, které jsou logickou podstatou, reprezentují na základě zkušenosti číslem. JavaScript automaticky převádí číslo 0 na nepravdu a libovolné číslo na pravdu. Zároveň při potřebě rozšíření funkcionality o další stav není nutné měnit datové schéma.



Obrázek 5.2: Relační schéma databáze

## 5.1.2 Serverová část

Adresářová struktura platformy 5.2 je inspirována Node.js doporučeními a zároveň má Feathers.js framework vlastní doporučenou adresářovou strukturu. Při využití Feathers.js CLI pro generování služeb se struktura včetně souborů pro testy vytváří sama.

Ve složce migration jsou soubory určené pro tvorbu schématu databáze pomocí knihovny Knex.js. Při každé změně vytváří Feathers.js nový soubor

stejné struktury. Pro zjednodušení jsem skripty ke všem tabulkám dal do jediného souboru viz obrázek v příloze A.12.

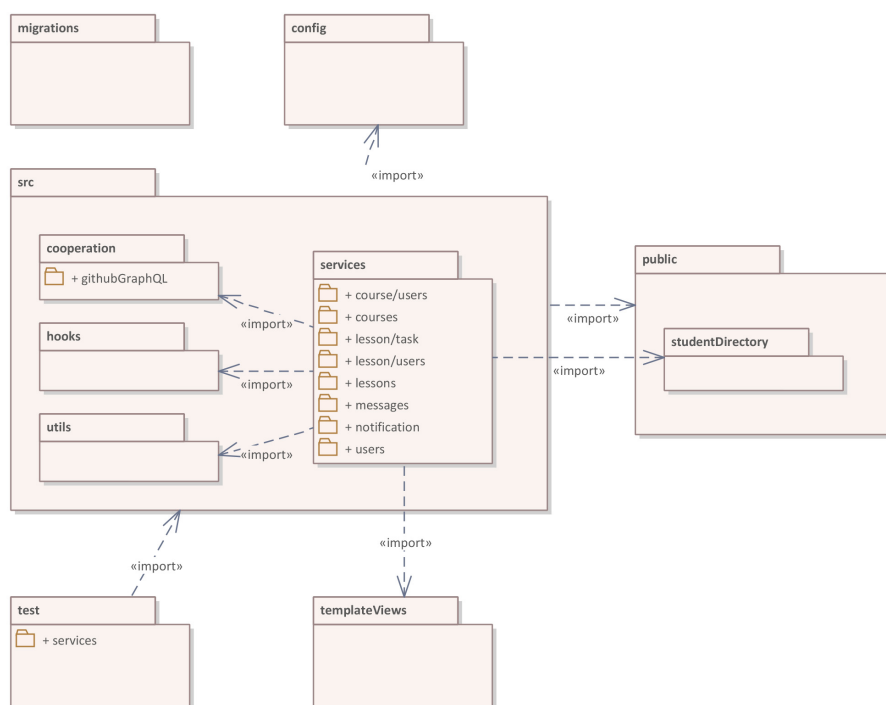
Config složka obsahuje JSON soubory definující základní nastavení Feathers.js. Src složka obsahuje všechny zdrojové soubory definující logiku serverového chování.

Jedinou přístupnou částí, kterou definuje Feathers.js, je složka public. Obsahuje složku studentDirectory, ve které jsou všechny projekty.

Notifikace jsou posílány jako HTML e-maily pro přívětivější vzhled s možností posílání odkazů, na které je možné kliknout. Každý e-mail využívá šablony, které jsou ukládány zvlášť ve složce templateViews ve formátu Handlebars.

Pro kód komunikující s GitHubem je vytvořen speciální adresář cooperation, který obsahuje kód obsluhující integraci s GitHubem. Hooks a utils obsahují podpůrný kód frameworku.

Celá serverová část je napsaná v TypeScriptu pro přehlednější a efektivnější vývoj. Definice API a logika jednotlivých rozhraní je definována uvnitř složky services. Ta se dále dělí podle jednotlivých tříd. Každé třídě Feathers.js automaticky vytváří endpointy typů: GET, POST, PATCH a DELETE pro obsluhu všech základních operací: čtení, tvorba, úprava a odstranění.



**Obrázek 5.3:** Diagram serverových částí platformy



### ■ 5.1.3 Klientská část

Stejně jako serverová část i klientská část je napsána v TypeScriptu. Dynamika webové aplikace je realizována pomocí Reactu. Všechny Reactové komponenty jsou funkcemi. Pro účely obsluhy stavu je využit Redux. Jak React, tak Redux mají svou doporučenou adresářovou strukturu, kterou se platforma inspiroje. Struktura adresářů platformy je znázorněna na obrázku 5.4.

Hlavním stavebním kamenem webu je stránka. Každá stránka má ve složce pages vlastní soubor. Stránky se skládají z opakovaně použitých komponent, které jsou uloženy ve složce components. Upravené komponenty Ant designu jsou uloženy v podsložce common. Komplexní komponenty, jako jsou kód editor, chat, nebo tabulky, ve svých složkách obsahují složitější strukturu. Obsahuje další specializované komponenty.

Protože aplikace obsahuje velké množství komponent a zároveň jednotlivé stránky mají odlišné komponenty, platforma využívá postupné načítání stránek<sup>1</sup> pomocí Reactové funkce lazy. Pro načtení dat z databáze je potřeba volat příslušné funkce definované Reduxem ve složce slices, uvnitř složky store. Pak jsou samotná volání implementována ve složce api.

### ■ 5.1.4 Tvorba kurzu

Jeden ze dvou klíčových procesů platformy je znázorněn na obrázku 5.5. Založit kurz může pouze uživatel s rolí lektora. V platformě je na roli navázána podmínka viditelnosti tlačítka startující proces tvorby kurzu.

Lektor vyplňuje nejdříve formulář, týkající se obecných informací o kurzu a následně formulář o jednotlivých lekcích. Formulář pro zadávání lekcí je realizován pomocí Ant design tabulky, kde každý řádek odpovídá jedné lekci. Lektor zadává název a popis lekce. Popisek je primárně určen pro lektora, aby později, když bude vytvářet detailní zadání k lekci věděl, co je cílem lekce.

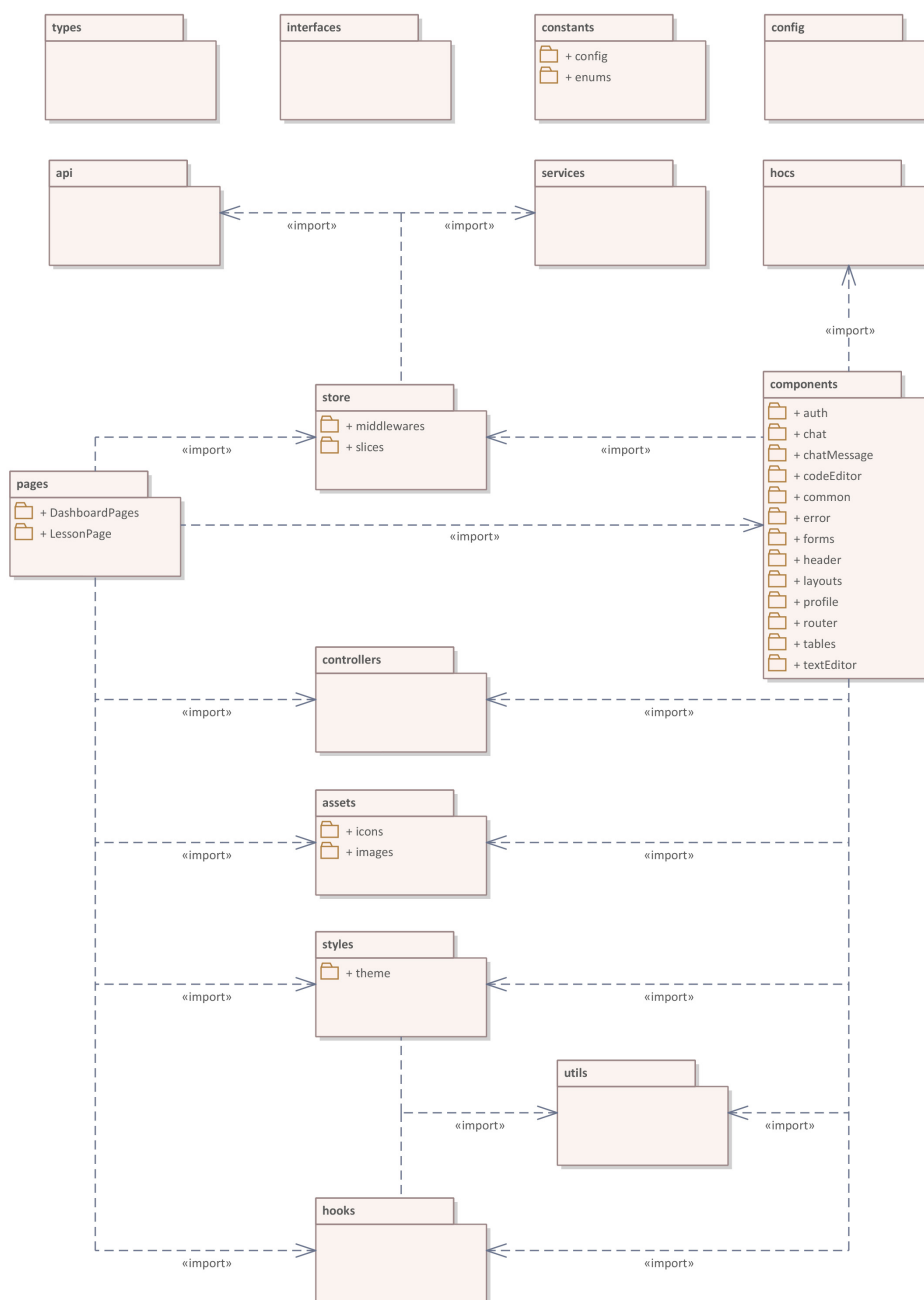
Akreditace na rekvalifikační kurzy trvají 3 roky, během kterých se osnova nesmí měnit. Jednotlivé kurzy jsou často stejné, proto je implementována možnost vytvořit lekce podle šablony. Šablonou může být každý vytvořený kurz.

Ve chvíli, kdy lektor vyplní všechny lekce a projde shrnující obrazovku, se data odešlou. V jedné transakci nejprve proběhne uložení dat do tabulky kurzů. Díky tomu je následně možné uložit jednotlivé lekce s referencí na kurz. Kurzům i lekcím je nastaven jako autor uživatel, který požadavek na server posílá.

Pokud proces proběhl v pořádku, klient je přesměrován na úvodní stránku platformy. V pravém horním rohu se zobrazí hláška, že byl kurz vytvořen. Zároveň se kurz zobrazí v levém menu, kde je seznam kurzů lektora. Jinak uživatel zůstává stále na stránce shrnutí tvořeného kurzu a v pravém horním rohu obrazovky se objeví chybová hláška s textem chyby.

---

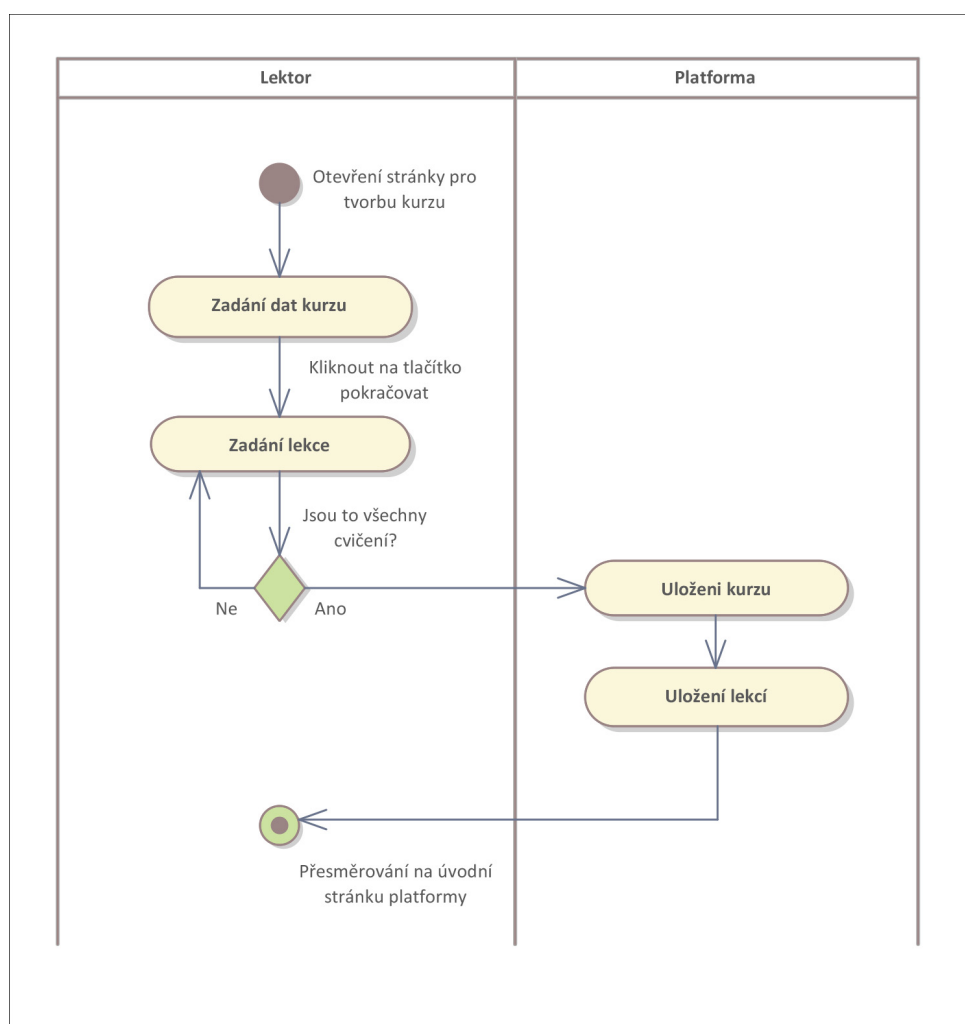
<sup>1</sup>tzv. Lazy Loading – technika pro načítání dat ve chvíli potřeby.[56]



Obrázek 5.4: Diagram klientských částí platformy

### 5.1.5 Přidání studenta do kurzu

Druhým klíčovým procesem je přidání studenta do kurzu – znázorněn na obrázku 5.6. To může udělat pouze autor kurzu. V administrační sekci kurzu lektor může přidávat studenty. Otevře se modální okno, které obsahuje stejný formulář, který vyplňuje lektor, když se chce do platformy zaregistrovat.



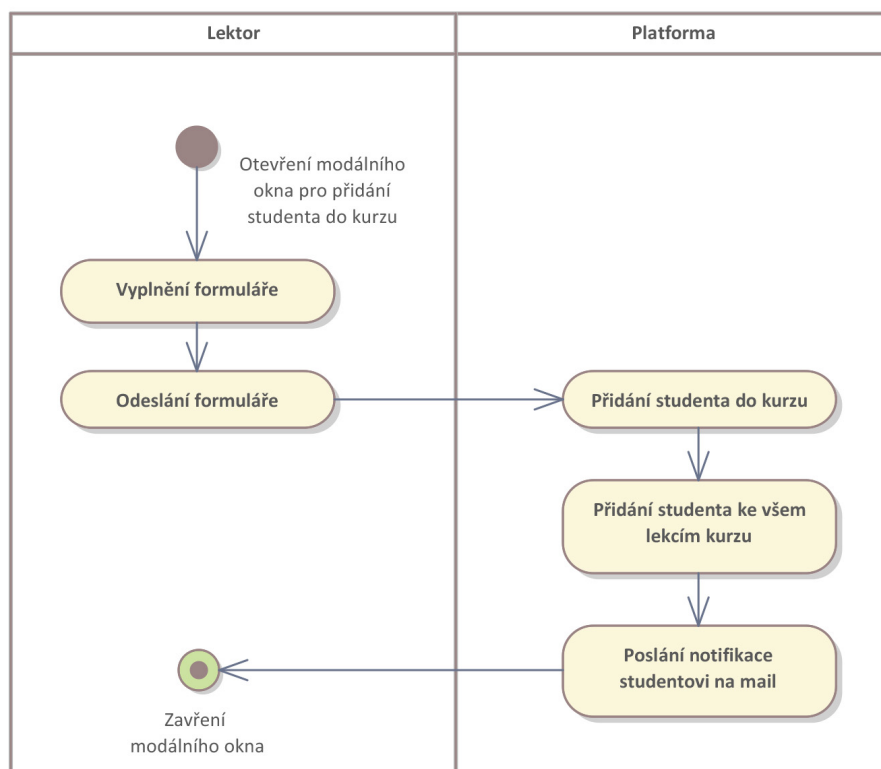
**Obrázek 5.5:** Aktivita diagram tvorby kurzu

Po odeslání formuláře na server proběhne v rámci jedné transakce nejdříve vytvoření studenta a jeho následné přidání do tabulky pro vazbu mezi uživatelem a kurzem. Následně lze podobným způsobem uživatele přidat ke všem lekcím kurzu. V základu kurzy i lekce nelze editovat a vidět.

Když celá transakce proběhne v pořádku, je studentovi na e-mail poslána notifikace, že byl přidán do kurzu. Zároveň se lektorovi zavře modální okno a vpravo nahoře obrazovky se objeví zelené upozornění, že student byl úspěšně přidán. V opačném případě se modální okno nezavře a zobrazí se v pravém horním rohu obrazovky červené upozornění s textem chyby.

### ■ 5.1.6 Životní cyklus lekce

Lekce nemůže stát sama o sobě. Musí být součástí kurzu. Jedná se o objekt, který má v relačním schématu 5.2 největší množství vazeb. Je to klíčová část celé platformy.



**Obrázek 5.6:** Aktivita diagram přidávání studenta do kurzu

Vznik je spojen s výše zmíněným procesem tvorby kurzu. Během procesu vzniká záznam v databázi o nově vzniklých lekcích. Lekce mají nastavený status na „*SEED*“. Název statusu byl zvolen podle anglického výrazu pro semeno. V tuto chvíli má lekce pouze název a podnázev. Lekce je vytvořena jako viditelná pro každého studenta bez možnosti editace.

V další fázi lektor vytváří detail cvičení. Píše zadání a připravuje kód pro doplnění. Ve chvíli, kdy je spokojený se zadáním kliká na tlačítko, že je lekce připravena. Status lekce se mění na „*READY*“.

Přechody mezi stavy jsou znázorněné ve stavovém diagramu na obrázku 5.7.



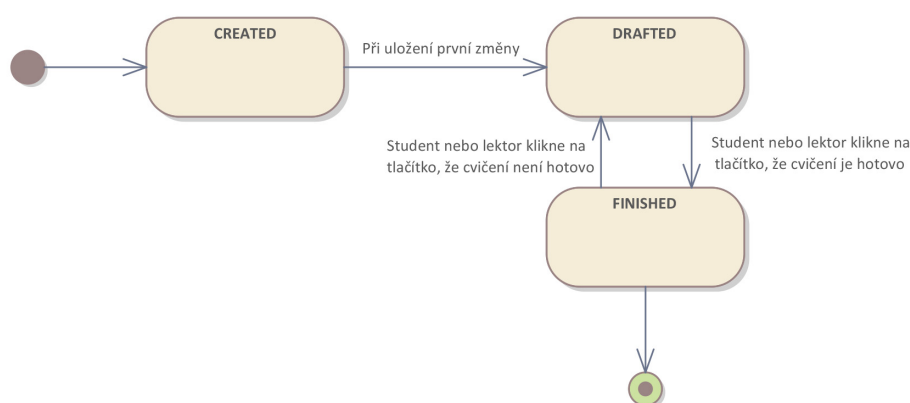
**Obrázek 5.7:** Stavový diagram lekce

Ve chvíli, kdy je student přidán lektorem do kurzu, jsou atributy „progress“ všech lekcí studentovi nastaveny na „*CREATED*“. Student si může zadání lekce zobrazit teprve ve chvíli, kdy má lektor lekci připravenou.

Ve chvíli, kdy se studentovi povede uložit první změnu kódu, je atribut progress nastaven na „*DRAFTED*“. Opomenutí uložení lekce nehrozí díky tomu, že je možné na stránce editoru kód uložit manuálně pomocí tlačítka, nebo automaticky při opuštění obrazovky lekce.

Pokud je student se svým řešením spokojený, může kliknout na tlačítko, že má řešení hotové. Díky tomu se nastaví atribut progress na „*FINISHED*“ a zároveň se pošle notifikace na e-mail lektora, že cvičení bylo dokončeno. Student i lektor mají možnost vrátit lekci do rozpracovaného stavu pomocí tlačítka na stránce webového kód editoru.

Přechody mezi stavy rozpracovaného cvičení jsou znázorněné ve stavovém diagramu na obrázku ??.



Obrázek 5.8: Stavový diagram plnění lekce studentem

## 5.2 Webový kód editor

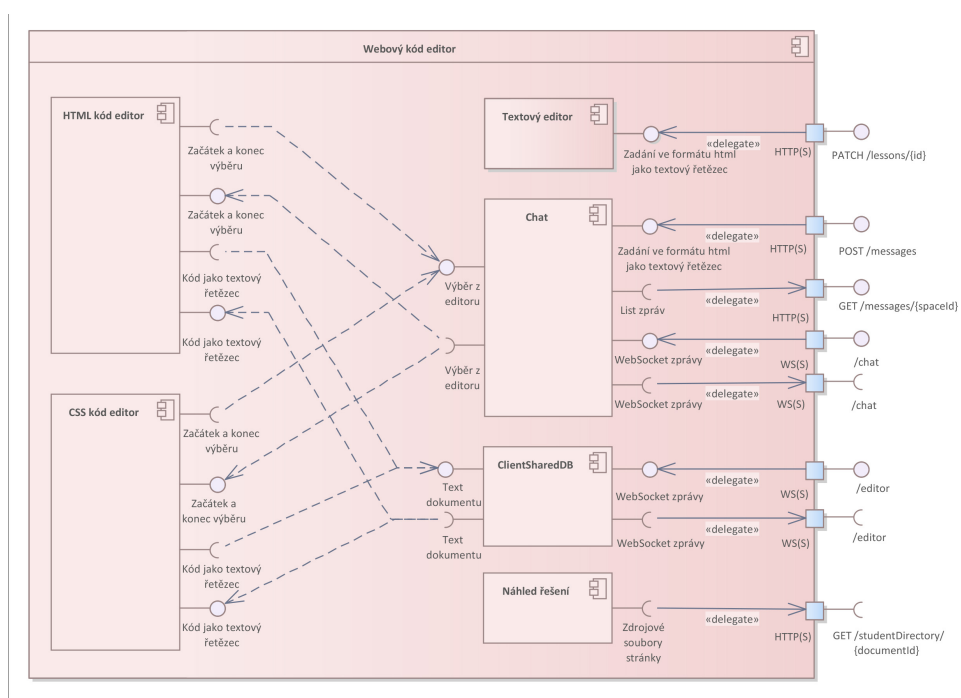
Nejkomplexnější částí celé platformy je kód editor. Zdrojový kód se nachází na několika místech. Obecná část se nachází ve složce LessonPage a konkrétní komponenty dále v codeEditor, textEditor a zároveň se využívá několika standardních komponent Ant designu, jako například Tabs. Tímto způsobem je uživatelsky řešeno překlíkávaní mezi jednotlivými webovými technologiemi.

Jedním z cílů rekvalifikačního kurzu je naučit studenty psát čistý kód. Pro kurz je nejvhodnější dělit kód dle technologií do vlastních souborů, jako to dělají i ostatní editory, jako například JSFiddle nebo PlayCode. Pro HTML je vždy soubor s názvem index.html, pro CSS style.css, pro JavaScript index.js a pro PHP je to index.php. Kvůli jednodušší administraci kolaborace mezi studentem a lektorem a pohodlnější sdílení kurzoru je každý soubor realizovaný speciální instancí CodeMirror editoru. Instancí je fixní číslo 4 pro každou lekci. Pro účely zjednodušení jsou na obrázku 5.9 zobrazeny pouze komponenty pro HTML a CSS kód editor, ale pro JavaScript i PHP jsou komponenty i jejich rozhraní stejné.

Součástí stránky editoru je také textový editor, který umožňuje lektorovi zadat komplexní stylizovaný popis, co se od studenta očekává. Je realizovaný knihovnou Quill.

Další součástí editoru je náhled řešení. Protože kód není žádným způsobem kontrolovaný proti zadání škodlivých příkazů a zároveň je potřeba důvěryhodně simulovat prostředí obrazovky, využívá se HTML značka iframe jako sandbox pro testování svého díla bez zásahu do zbytku platformy.

Diagram komponent na obrázku 5.9. mimo jiné ukazuje, jakým způsobem mezi sebou komponenty komunikují. Komponenty kód editorů jsou prakticky stejné. Liší se pouze svým nastavením. Každý editor má definovaný jazyk, kterému rozumí. Díky tomu pak zvládá našeptávat i barevně odlišovat jednotlivé části kódu.



Obrázek 5.9: Diagram komponent webového kód editoru

### 5.2.1 PHP kód editor

Webový kód editor pro výuku PHP je stejně jako HTML, CSS i JavaScript realizován pomocí knihovny CodeMirror. Způsob ukládání a kooperace je proveden úplně stejně.

Hlavním rozdílem je implementace náhledu řešení. I když je soubor hostovaný ve stejném adresáři společně se soubory HTML, CSS a JavaScriptu, tak Node.js nedokáže interpretovat PHP. Na druhou stranu interpretace programovacího jazyka pouze v prohlížeči je bezpečnější, než na serveru.

Pro interpretaci PHP platforma využívá knihovny PIB: PHP in Browser (and Node.js) aka php-wasm, která PHP kód převádí WebAssembly formátu.

Pro vyhodnocení kódu je nutné, aby knihovna byla připravena. Když je vše iniciované je možné PHP kód převést a nechat vyhodnotit. Služba vrací pole textových řetězců odpovídající výsledku.

V případě chyby knihovna vrací po sobě tři výsledky. Nejdříve pole s jedním textovým řetězcem obsahující znak nového řádku. Druhé pole obsahuje chybovou hlášku a třetí pole obsahuje stejný výstup jako první.

Výsledky po kliknutí na tlačítko „zobrazit“ na stránce editoru sloučí do jednoho pole. Každý textový řetězec obalí HTML značkami `p` a zobrazí na stránce editoru.

## 5.2.2 Tok dat skrz editor

Protože webový kód editor využívá všechny části aplikace, rád bych v této podkapitole přiblížil, jakým způsobem jsem jednotlivé části navrhl a jak fungují.

Jak bylo nastíněno v kapitole Životní cyklus lekce 5.1.6, je k lekci v různých chvílích přistupováno různě. Statusy „*SEED*“ a „*READY*“ a stavy řešení „*CREATED*“ a „*DRAFTED*“ odrážejí skutečnost, zda kód zadání lekce nebo kód studenta existuje v GitHub repozitáři. Pokud neexistuje, je součástí procesu jeho vytvoření. V opačném případě se pouze edituje stávající stav.

Pro ilustraci nyní představím situaci, kdy lektor cvičení vytváří, aby bylo zřejmé, co se děje. Tento proces je zachycen na sekvenčním diagramu na obrázku 5.11 Případ pro začátek plnění úkolu, dílčí úprava zadání lektorem, nebo úprava řešení studentem probíhají podobně s jednou změnou, týkající se komunikace s GitHubem, která jsou buď stejná, nebo chudší o tvorbu větve.

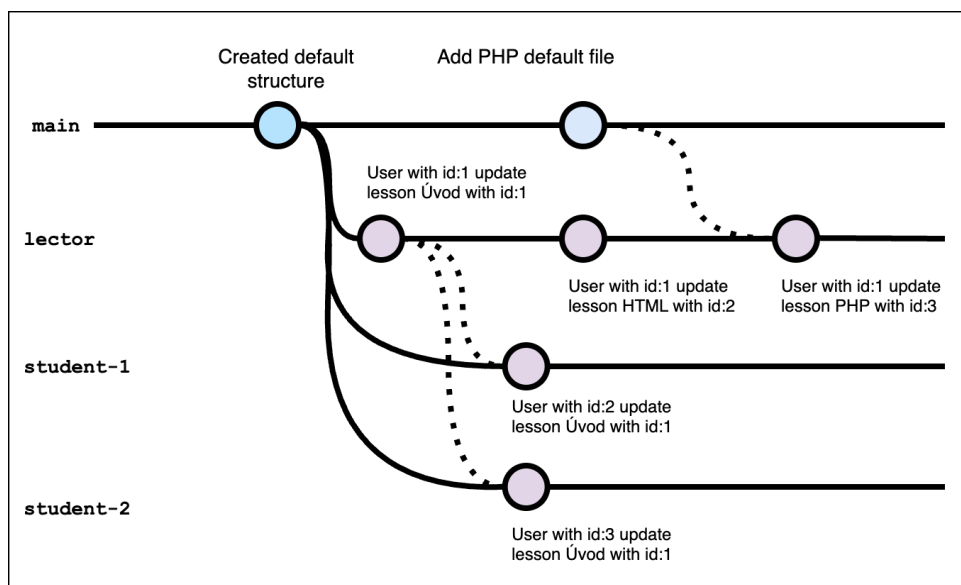
Hostované projekty není možné nijak smazat. Není to účel ani z pohledu bezpečnosti. Autoři jsou skrz platformu anonymizováni a podle atributu `id` není možné ani v repozitáři určit, kdo je kdo, dokud někdo nezíská přístup k datům databáze platformy.

## Úvodní volání

V první fázi, před vykreslením stránky, se klientská sekce snaží navázat spojení se socketem editoru pro účely následných úprav dokumentu.

V druhé fázi se realizují volání pro načtení stránky. Probíhá dotaz na server po datech k lekci. Klientovi se vrací kompletní objekt cvičení obsahující název a obecný popis. Následuje dotaz na informace po vazbě mezi lekcí a uživatelem. Na serveru probíhá komunikace s GitHubem. Pro účely práce s větvemi potřebujeme `OID`, které musíme získat. Pokud neexistuje větev, kterou hledáme, tak nová větev vzniká z hlavní(main) větve. Proto chceme `OID` main větve. Ta obsahuje pouze šablonu prázdného projektu. Když zakládáme novou lekci, její obsah se kopíruje na server. Tím se obsah hostuje pod dokumentem `id`. V závěrečné fázi druhého volání na server se server připojuje k poslouchání socketu, aby mohl aktualizovat stav hostovaného řešení podle úprav studenta.

Jak je vidět na obrázku 5.10 každý student má vlastní větev. Lektoři mají jednu dohromady. Ve chvíli, kdy je cvičení připravené ve větvi lektora, tak



**Obrázek 5.10:** Struktura repozitáře ukládající výstupy editoru

se kód kopíruje na serveru ke studentovi do adresářové struktury. Upravený kód studentem se následně ukládá do jeho vlastní větve. Nejedná se o přímou vazbu mezi větvemi. Čárkovanou čarou jsou naznačena pouze propagace dat skrz server.

Ve třetí fázi úvodního volání po načtení stránky se prohlížeč ptá serveru na obsah hostovaného obsahu, aby mohl zobrazit náhled v iframu v editoru.

## ■ Úpravy

Uživatel během každého kliknutí na klávesnici vyvolává událost změny obsahu, která se automaticky bere a posílá socketem. Díky tomu se o změně stavu server dozví. Z výkonnostních důvodů není dobrý nápad na serveru měnit obsah při každé změně, ale jednou za 800 milisekund. Číslo je zvoleno experimentálně. V případě žádného nebo malého časového intervalu měla aplikace výpadky.

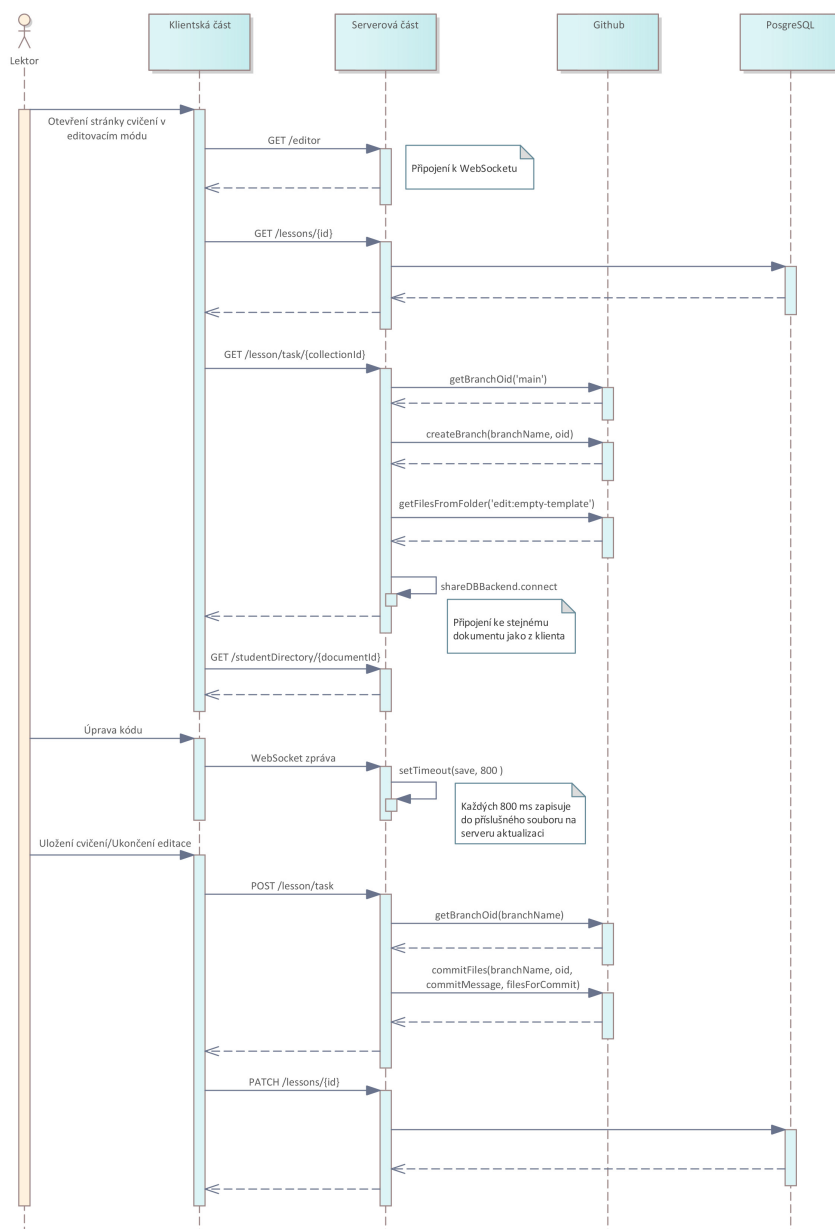
## ■ Ukončení úprav

Když se jedná o editaci zadání lekce a klientská část se snaží uložit kód, probíhá ve dvou fázích. První fáze se týká uložení samotného kódu a druhá uložení zadání.

Protože kód je současně hostovaný na serveru, není třeba ho na server posílat. Stačí poslat požadavek na uložení konkrétního kódu. Na popud klienta server vezme obsah souborů lekce a převede je na Base64. Následně je odešle a uloží do konkrétní větve. Tím vzniká commit v repozitáři.

Zadání cvičení je bráno z Quillu a uloženo jako textový řetězec přímo do databáze.





Obrázek 5.11: Sekvenční diagram tvorby zadání lekce

## 5.3 Chat

Jak je naznačeno na diagramu komponent na obrázku 5.9, součástí stránky s webového kód editoru je panel umocňující posílání zpráv mezi studentem a lektorem. Chat se zobrazuje pouze ve chvíli, kdy je cvičení v procesu vypracovávání a lektor chat vidí pouze, když si zobrazí studentovu lekci. Každá lekce má svůj chat a je možné zprávy posílat v reálném čase. Cílem chatu je umožnit méně komunikativním studentům zeptat se na dotaz, nebo

se zeptat mimo hodiny kurzu. Stále je bráný za primární komunikační kanál fyzický nebo on-line dialog mezi studentem a lektorem.<sup>2</sup>

Chat funguje v reálném čase. Pro účely získání historie se po načtení stránky nejdříve pošle dotaz na server na všechny napsané zprávy.

Přidanou hodnotou chatu je schopnost posílat ve zprávě odkaz na konkrétní část kódu. Student nebo lektor mohou označit libovolnou část kódu a kliknout na tlačítko „Přilož odkaz na kód“. Tím se do zprávy přidá odkaz na danou část kódu.

Z důvodu implementace přidání odkazu na kód se pro zadávání zprávy využívá také Quill, ale ve své úplně nejzákladnější verzi bez panelu s možnostmi formátování. Je povolený pouze odkaz, který je vytvářen kódem. Do atributu href se plní ve formátu id editoru a výběr od a výběr po. Díky tomu má každý editor vlastní unikátní 11místné id, které definuje náhodný generátor na straně serveru. Je unikátní napříč celou platformou.

Ve chvíli, kdy čtenář na odkaz klikne, v konkrétním editoru se označí zaslaná část. Jedná se pouze o výběr. Díky tomu nedochází k rušení jiné syntaxe.

## 5.4 Integrace s GitHubem

GitHub je úložištěm kódu. Serverová část platformy je přes GraphQL integrována s GitHubem. Pro podporu dotazů platforma využívá minimalistické knihovny graphql-request napsané v TypeScriptu.

Lektor má vlastní větev, ve které má rozpracované nebo vypracované zadání k jednotlivým lekcím. Každý student má vlastní větev, ve které si ukládá vypracovaný kód jak je znázorněno na obrázku 5.10. Ve větvích lektora i studentů složky odpovídají jednotlivým lekcím.

Když lektor začíná připravovat lekci, kopíruje se struktura ze složky empty-template, která je v každé větvi. Je to jediná složka, kterou obsahuje hlavní větev main, ze které všechny větve vznikají a žádné se nesmí mergovat zpátky do ní. Když začíná student lekci vypracovávat, z větve lektora se kopírují soubory na server a pak vzniká nová složka s upravenými zdrojovými soubory.

---

<sup>2</sup>Platforma má sloužit jako doplněk k frontální výuce.

## Kapitola 6

### Ověření kvality

V této kapitole pomocí testů použitelnosti inspirované knihou[91] ověřím funkčnost platformy. Kniha apeluje na průběžné testování. Testy by měli probíhat od obecných, kterých se může účastnit kdokoliv až po cílené, kterých se bude účastnit pouze výběr vhodných zákazníků. Cílem je otestovat webový kód editor a použitelnost pro lektora během výuky rekvalifikačního kurzu. Hlavními testovanými funkcionalitami jsou:

- Zadávání lekce lektorem
- Vypracování úkolu studentem
- Kooperace studenta s lektorem

#### 6.1 Příprava testování

Po provedení Nano<sup>1</sup> a Mikro<sup>2</sup> testech užití jsem se rozhodl připravit testování uživatelské testy simulují výuku rekvalifikačního kurzu u společnosti MAVO. Výuka probíhá v jedné učebně s maximální kapacitou 15 studentů pod vedením jednoho lektora po dobu 14 dní. Lektor přednáší před tabulí. Programuje se studenty ukázkové příklady, které promítá na plátno. K vysvětlení komplexnější látky nebo odpovědí na otázky využívá tabuli. Výuka jednoho dne se skládá z dopoledního a odpoledního bloku, rozdělených pauzou na oběd.

Testování probíhá ve třech fázích. První fází je testování základní orientace na platformě a vypracování lekce, obsahující pouze HTML zadání. Druhá fáze je zaměřena na komplexnější testování webového kód editoru. Poslední fáze obsahuje testování administrátorského rozhraní.

První dvě fáze obsahují přednášku lektora, který zároveň provází procesem testování a během celého procesu sbírá zpětnou vazbu od studentů. Testování administračního rozhraní probíhá pomocí průchodů platformou podle detailních testovacích scénářů podle metodiky TMAP Next. V testovacích scénářích rozděluji prioritu na nízkou, střední a vysokou. Hodnotu priority

<sup>1</sup>Základní forma ověření kvality, kde je klíčové získání libovolného člověka, kterému záleží na produktu a nechat ho chvíli pracovat s platformou a pouze pozorovat.[91]

<sup>2</sup>Detailnější ověření kvality, kde je je zapotřebí již definovat úkol a získání správného člověka pro jeho splnění.[91]

určuje důležitost dané funkce pro fungování celého procesu jako celku.[92] Testy pokrývají požadavky z diagramů užití 4.2 a 4.1.

Platforma pro účely testování byla vystavena u společnosti Render. Byl využit bezplatný plán. Platforma běží na adrese: „<https://platform-client-zj90.onrender.com>“. Obsah hostingu je vidět na obrázku 6.1. Pro účely testování byl vytvořen nový repozitář.

NAME	STATUS	TYPE	RUNTIME
platform_client	Deploy succeeded	Static Site	Static
platform_backend	Deploy succeeded	Web Service	Node
platform_db	Available	PostgreSQL	PostgreSQL 15

**Obrázek 6.1:** Administrační rozhraní cloudové platformy Render

### 6.1.1 Volba účastníků

Vybral jsem pět dobrovolníků, kteří neznají webové technologie. Pomocí nich budu testovat uživatelskou přívětivost platformy a schopnost práce s platformou.

Další čtyři lidi jsem vybral z absolventů mnou vedeného rekvalifikačního kurzu Tvorba www stránek ve společnosti MAVO. Poslední účastníci je absolventka Digitální Akademie: Web společnosti Czechitas. Všichni účastníci ovládají git. Zde sleduji adaptaci ze známého prostředí Visual Studio Code a očekávám komplexnější využití platformy a experimentování s tím, co účastníkovi dovolí.

## 6.2 Využití při výuce

Vytvořil jsem pro testování dvě cvičení a zároveň jsem vytvořil osnovu celého rekvalifikačního kurzu:

### 1. HTML

- a. HTML struktura, základní značky (p, span, a, img, ol, ul)
- b. sémantické značky
- c. div, atributy (id)
- d. formulář
- e. tabulka, video, audio
- f. iframe, Google maps, Youtube video

## 2. CSS

- a. Napojení stylů, stylování značek, základní styly (color, background-color, font-family, font-style, font-width, font-size, width, height)
- b. Stylování tříd a id, flex box (direction, justify, align, grow) , styly (min-\*, max-\*)
- c. Jednotky, boxmodel - margin, outline, border, padding
- d. Kaskáda neboli hierarchie stylů
- e. Přechody a animace

## 3. JavaScript

- a. Napojení na HTML, proměnná, datový typ, console.log, document.write, šablonový text
- b. Podmínky a porovnání, for cyklus
- c. Funkce, pole a objekt
- d. Selector a eventlistener
- e. Manipulace s DOMem (appendChild, setAttribute, innerHTML)

## 4. PHP

- a. Proměnná, datový typ, var\_dump
- b. Pole a for cyklus
- c. Objekt a dědičnost
- d. Napojení na databázi SQLite

Jednotlivé lekce jsou poskládané tak, aby přímo odpovídaly požadavkům akreditované osnovy 2.1.1. Lekce na sebe navzájem navazují. Komplexita zadání postupně roste. Pro účely demonstrace řešení má lektor zároveň vytvořený jeden studentský účet, kde může demonstrovat vlastní řešení.

### ■ Jednoduchá lekce

Pro tento účel jsem zvolil lekci „Formulář“. Cílem lekce je studenty seznámit a naučit HTML značky používané pro formulář, včetně atributů for, type a name. Lekce se řadí do druhé fáze HTML bloku. Od studentů se očekává znalost základních značek, sémantické značky, div a atribut id.<sup>3</sup>

Popis lekce je sepsán v krocích. Kroky na sebe navazují:

1. Vytvořit obecnou značku form
2. Vytvořit vstupní pole formuláře s popisky a referencí na vstupní pole
  - a. Vstupní pole pro jméno houby – textové pole

<sup>3</sup>Účastníci bez znalosti webových technologií budou znevýhodněni, mým cílem je ale i otestovat schopnost pomocí platformy dovést krok za krokem studenta k očekávanému výsledku.

- b. Vstupní pole pro latinský název houby – textové pole
  - c. Vstupní pole pro popis houby – blok textu
  - d. Vstupní pole pro výběr stupně jedlosti s možnostmi (jedlá a velmi chutná, jedlá, nejedlá, jedovatá, prudce nebo smrtelně jedovatá, podezřelá a potenciálně zdraví škodlivá)
  - e. Vstupní pole pro vzhled struktury pod kloboukem s možnostmi lupeny, rourky, nebo jiné – radio
  - f. Vstupní pole pro čas, kdy houba roste s možnostmi 5. až 11. měsíce v roce – checkbox
  - g. Vstupní pole pro obrázek houby
  - h. Vstupní pole pro charakteristickou barvu klobouku
3. Vytvořit formulářovou sekci pro administraci včetně legendy
    - a. Vytvořit vstupní pole pro e-mail administrátora
    - b. Vytvořit vstupní pole pro datum publikace
  4. Vytvořit odesílací tlačítko
  5. Stylování pomocí divů tak, aby každý vstup včetně svého popisku byl na svém řádku

Lekce je sestavena tak, aby se student s nejdůležitějšími prvky setkal hned na začátku a postupně si teprve rozšiřoval své znalosti. Z praxe vím, že je pro studenty lepší, když si probíranou látku mohou zasadit do reality, proto jsem jako téma testování zvolil atlas hub. Během cvičení student pomocí HTML vytvoří kompletní formulář pro vkládání houby do pomyslné databáze. Možné řešení je na obrázku 6.2.

Informace o houbě

Zadejte název houby:

Zadejte latinský název houby:

Zadejte popis houby:

Vyberte stupeň jedlosti:

Vyberte vzhled struktury pod kloboukem:  lupeny  rourky  jiné

Vyberte období růstu:  květen  červen  červenec  srpen  září

Nahrajte obrázek houby:  No file chosen

Vyberte charakteristickou barvu klobouku:

---

Informace o uživateli

e-mail:

datum publikace:

**Obrázek 6.2:** Příklad řešení lekce Formulář

## ■ Komplexní lekce

Vybral jsem lekci „Funkce, pole a objekt“ z bloku JavaScript. Cílem lekce je zopakovat naučené dovednosti, přidat dělení kódu do funkcí a seznámit studenty s komplexními datovými typy polem a objektem. Pro účastníky bez znalosti webových technologií bude řešení této lekce pokus.

Popis lekce je sepsán v krocích. Kroky na sebe navazují:

### 1. HTML část

- a. Stránka bude mít nadpis úrovně jedna s textem: „Generování hub“
- b. Vytvořte formulář s jedním vstupním polem typu číslo a odesílacím tlačítkem

### 2. CSS část

- a. Nastylujte formulář tak, aby byl na středu a tlačítko bylo pod vstupním polem.
- b. Tlačítko bude měnit svůj svou barvu pozadí podle toho, jestli je na něj najeto myší, či není

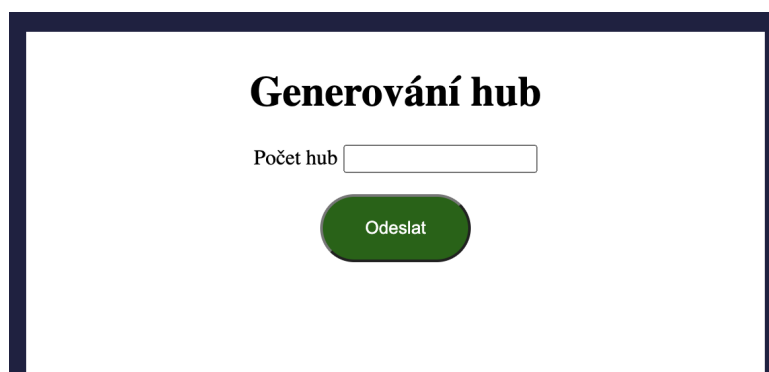
### 3. JavaScript část

- a. Vytvořte proměnou druhHouby, která bude mít v sobě pole textových řetězců: „Hřib“, „Holubinka“, „Muchomůrka“
- b. Vytvořte proměnou jménoHouby, která bude mít v sobě pole textových řetězců: „Borový“, „Zelená“, „Kovář“, „Tygřovaná“
- c. Vytvořte proměnou popisky, která bude mít v sobě pole textových řetězců: „Výborná houba rostoucí spíše na kraji lesa. Má ráda teplejší prostředí.“, „Chutná houba rostoucí v symbióze se smrky. Svým vzhledem snadno zaměnitelná s muchomůrkou zelenou.“, „Houba s výrazně bílým podhoubím. Lze od praváka odlišit svou hořkou chutí“, „Chutný houba rostoucí v blízkosti modřínů. Charakteristický svým žlutým kluzkým kloboukem.“
- d. Vytvořte proměnou jedlosti, která bude mít v sobě pole textových řetězců: „jedlá a velmi chutná“, „jedlá“, „nejedlá“, „jedovatá“, „prudce nebo smrtelně jedovatá“, „podezřelá“ a „potenciálně zdraví škodlivá“
- e. Vytvořte proměnou houby a nastavte jí hodnotu prázdné pole
- a. Vytvoří se funkce pro generování jmen hub se dvěma vstupy. Funkce na základě náhodného generátoru vezme prvek s vygenerovaným id zvlášť pro jméno a zvlášť pro druh. Výsledný řetězec spojí a vrátí jako jednu hodnotu
- b. Vytvoří se funkce pro generování popisu. Funkce bude brát na vstupu jednu hodnotu. Bude se jednat o pole popisků. Na základě náhodně vygenerovaného klíče vrátí hodnotu z pole.

- c. Vytvoří se funkce pro generování jedlosti. Funkce bude brát na vstupu jednu hodnotu. Bude se jednat o pole jedlostí. Na základě náhodně vygenerovaného klíče vrátí hodnotu z pole.
  - d. Vytvoří se funkce pro generování data. Funkce nebude mít žádný vstup. Na základě náhodného generátoru vygeneruje je datum, který i vrátí.
  - e. Vytvoří se funkce pro generování rozsahu období růstu. Funkce bude brát na vstupu dvě hodnoty. Jedna se bude týkat růstu od a druhá růstu do. Funkce bude vracet textový řetězec ve formátu DD.MM.YYY – DD.MM.YYY
  - f. Vytvoří se funkce pro generování url obrázku. Funkce bude mít na vstupu jméno houby a na výstupu bude validní url: „https://atlashubv-sources.eu/{jménoHouby}\_{druhHouby}” a vše malými písmeny.
- a. Obecná a základní funkce bude volána při odeslání formuláře. Bude generovat houby. Pomocí volání předem připravených funkcí bude vytvářet objekt houby a následně s ním naplní pole v proměnné houby. Na konci vypíše objekt houby do konzole prohlížeče
- (i) Vytvořit objekt, který bude reprezentovat houby. Klíče objektu budou: jméno, popis, jedlost, dobaRůstu, urlHouby

Cvičení začíná v úplně prázdném editoru. Student musí vytvořit celou HTML strukturu včetně napojení na další soubory. Zadání je navrženo volně, aby měl student možnost zadání implementovat dle svých schopností a uvážení. Protože výuka odchyťávání událostí pomocí funkce „addEventListener” bude vyučována v následující lekci, tak se využije HTML atributu „onclick” nebo „onsubmit”. Cvičení simuluje prostředí rekvalifikačního kurzu, u kterého se počítá, že by mělo zabrat včetně vysvětlení půl dne, proto není cílem, aby se během testování stihlo celé.

Možné řešení vizuální části je na obrázku 6.3.



The image shows a web form with a dark blue border. At the top center, the title "Generování hub" is displayed in a bold, black font. Below the title, there is a text input field with the label "Počet hub" to its left. Underneath the input field is a green, rounded rectangular button with the text "Odeslat" in white.

Obrázek 6.3: Příklad řešení lekce Funkce, pole a objekt



### ■ 6.2.1 Výsledky testování

I když dotazníkové šetření je jednodušší na vypracování, vysoce efektivní kvantitativní technikou, nedochází u ní k tzv. „interview bias“<sup>4</sup> a náklady jsou prakticky nízké jsem se rozhodl pro formu sběru dat pomocí rozhovorů. Jedná se o pracnou složitě vyhodnocující techniku. Na druhou stranu v aktuální situaci, kdy platforma není kompletní a testování se účastní i lidé, kteří nemají zkušenosti s webovými technologiemi vidím formu rozhovoru jako přínosnější. Cílem bylo pokládat obecné otázky a zkoumat názory účastníků na použitelnost platformy.[93]

Testování proběhlo během dvou setkání. Během prvního setkání jsem začal představením platformy a její vize. Následovala přednáška seznamující účastníky s webovými technologiemi pro lekci. Během ukázky jsem využil speciální studentský účet pro demonstraci základních principů.

Druhé setkání drželo strukturu prvního setkání. Obě setkání byla naplánována na 3 hodiny, rozdělené do dvou bloků. První blok byl cílen na představení problematiky. Druhý blok sloužil pro samostatnou práci účastníků s prostorem pro dotazy a dodatečné vysvětlení látky.

Studenti se bez problémů zvládli přihlásit do platformy a pracovat na zadaném úkolu. Všichni studenti alespoň jednou využili chat. Kolaborace mezi studentem a lektorem fungovala bez problému. Vizualizace řešení splnila očekávání studentů.

Nejzávažnější nalezenou chybou je občasné špatné uložení kódu. Do kódu se uloží mezery navíc. Náhled následně nekoresponduje s vytvořeným kódem. Pro pokračování dál je nutné lekci zavřít a posléze ji znovu otevřít a kód opravit. Dalším výstupem je poptávka po notifikacích nové přijaté zprávy. V tuto chvíli tato funkcionality není implementována. Pokud si účastník skryl panel se zprávami například kvůli malému displeji svého notebooku, tak neměl možnost se dozvědět, že mu přišla od lektora zpráva. Kompletní seznam nasbíraných podkladů pro opravy a vylepšení je v příloze C.

## ■ 6.3 Administrační prostředí

Lektor může vytvářet kurzy, spravovat lekce a studenty. Cílem testování je otestovat funkčnost implementovaných procesů 5.5 a 5.6.

Platforma v tuto chvíli neobsahuje implementaci všech případů užití z diagramu 4.2. Služby jako úprava a mazání osobních údajů, kurzů, lekcí a studentů v tuto chvíli buď chybí úplně, a nebo jsou v platformě připravena hluchá tlačítka. Z toho důvodu byly testovány pouze funkcionality, které platforma obsahuje.

Testovací scénáře jsou uvedeny v příloze B.

<sup>4</sup>Zkreslení na základě různorodosti položení otázek a chování tazatele.

### ■ 6.3.1 Výsledky testování

Testovací scénáře úspěšně ověřily funkčnost procesu tvorby kurzu i procesu přidávání studentů do kurzu. Testy odhalují pozitivní průchod platformou.

Během testování bylo odhaleno, že data vytvořených kurzů i studentů se špatně delegují uvnitř platformy. V posledních krocích testů se nezobrazují kurzy v bočním panelu a studenti v tabulce studentů. Pro jejich zobrazení je nutné znovu načíst stránku, aby se data načetla z databáze. Tento problém vzniká chybnou propagací dat z komponenty do centrálního úložiště dat a následné načítání dat z úložiště.

# Kapitola 7

## Závěr

Rekvalifikační kurzy jsou jedním ze základních nástrojů aktivní politiky nezaměstnanosti. Stát poskytuje až 50 000 Kč v průběhu tří let jednotlivci, hledajícímu nové zaměstnání. Na území České republiky v tuto chvíli rekvalifikační kurzy aktivně provozuje 27 společností, zaměřených na webové technologie. Další společnosti, jako například Czechitas, provozují kurzy za účelem pomoci lidem získat zaměstnání v IT bez akreditace.

Všechny tyto společnosti jsou potenciálními zájemci o platformu vytvořenou v rámci této diplomové práce. Sami se potýkají s otázkou jak nové zájemce efektivně naučit základy IT.

Já osobně tuto otázku řeším přes dva roky. Během té doby jsem ji diskutoval s mnohými zkušenějšími lektory, hledal možnosti na internetu a sám odučil přes 1 000 hodin rekvalifikačních kurzů.

Pro mě samotného byla tvorba této diplomové práce velmi přínosná. Prošel jsem si celý vývojový proces platformy: zpracování požadavků, návrh řešení, implementace, otestování a sepsání dokumentace. Zároveň bylo povzbudivé poslouchat názory, jakou má platforma budoucnost a k čemu všemu by bylo možné ji využít. Prohloubil jsem si znalosti vývoje v Reactu, Feathers.js a Socketech. Poprvé jsem implementoval webový editor, což jsem shledal jako komplexní a náročné.

### ■ Dosažené výsledky

Při tvorbě platformy byly dosaženy cíle stanovené v úvodu práce a byly splněny všechny požadavky zadání až na on-line přenos videa skrz platformu. Z důvodu preference společnosti MAVO vedení kurzů prezenčně, a zároveň možnosti pro přenos videa paralelně s platformou využít služby Google Meets nebo Zoom, jsem nakonec požadavek přesunul do plánu na další verzi platformy. Společnost MAVO, u které vedu rekvalifikační kurzy, přistoupila na on-line výuku pouze z nouze a jak to bylo možné, hned se vrátila k prezenční výuce. Vznikaly problémy s poskytováním počítačů studentům a i studenti chtěli prezenční výuku. Nad rámec zadání jsem věnoval pozornost platformě pro správu kurzů jako takové.

Text práce obsahuje na úvod seznámení se s problematikou rekvalifikačních kurzů v České republice. Představuji blíže rekvalifikační kurz společnosti MAVO a uvádím další poskytovatele rekvalifikačních kurzů. Porovnal jsem

jejich služby a poukázal na rozdíly. Dále v textu rozšiřuji povědomí o možnostech alternativních způsobů vzdělávání es ve webových technologiích. V další kapitole se věnuji porovnání desktopových a webových kód editorů. Představuji různé přístupy k implementaci webových editorů.

Pokračuji detailním představení postupu vývoje první verze platformy. Definuji tři hlavní skupiny uživatelů, pro které je platforma určena. Pomocí diagramů užití a drátěných modelů definuji požadavky na platformu. Následně popisuji samotnou implementaci podpořenou o UML diagramy.

Pro ověření kvality jsem vytvořil návrh osnovy jednotlivých lekcí celého kurzu Tvorby www stránek. Kompletně jsem dodělal dvě cvičení. Platformu hostuji u společnosti Render. Provedl jsem uživatelské testy se skupinou dobrovolníků. Zároveň jsem představil aplikaci několika dalším lektorům.

### ■ Rozšiřitelnost platformy

Platforma je vyvinuta s cílem ji do budoucna rozšiřovat. Dalším krokem bude začlenění platformy do výuky rekvalifikačních kurzů ve společnosti MAVO nebo Czechitas. Na základě nasbírané zpětné vazby ji dále budu upravovat a rozvíjet. Platforma v tuto chvíli obsahuje pouze základní funkcionality pro využití při výuce, pro sdílení vývojového prostředí a pro pohodlné řešení náhledu.

Během testování se objevily chyby v implementaci platformy, které je v první řadě třeba vyřešit. Zároveň vznikly zajímavé podněty pro rozšíření platformy.

Platformu lze rozšířit o podporu dalších programovacích jazyků. Nejprve bych uvažoval o dodání programovacích jazyků Python a Java. Realizace by proběhala pomocí WebAssembly, podobně jako v tuto chvíli funguje implementace podpory jazyka PHP.

Dále lze platformu rozšířit tak, aby umožnila plně samostatnou on-line výuku. Jednalo by se primárně o domyšlení a implementaci výkladových materiálů v platformě.

Jedna věc je vývoj a rozšiřování samotné platformy. Na druhou stranu by bylo zajímavé pokračovat ve vizi platformy pro podporu výuky čistě pedagogickým směrem. Zkoumat a navrhnout efektivní osnovu, časovou dotaci a postup výuky tak, aby to pro studenty bylo co nejstravitelnější a za co nejkratší čas byli schopni se ucházet o zaměstnání v oboru.

### ■ Využitelnost platformy

Platforma je využitelná především pro lektory rekvalifikačních kurzů a jejich studenty. Platformu si může každý hostovat v preferovaném prostředí. Stačí pouze doplnit vlastní proměnné.

Hlavním přínosem platformy je podpora začínajících studentů, aby se během výuky mohli soustředit na seznámení se s technologiemi a nemuseli řešit zprovozňování prostředí. Dalším klíčovým přínosem je propojení studenta s lektorem. Oprava i drobné chyby může zabrat mnoho času. Jak již ukázalo testování a rozhovory s lektory, platforma zrychluje proces pomoci studentovi.

Lektor může z pohodlí svého počítače s kódem experimentovat a prohlédnout si mnohem rychleji potřebné pasáže.

Platforma se snaží zároveň podpořit nesmělé studenty k dotazům na lektora ohledně nejasností skrze chat, zabudovaný uvnitř platformy. Zároveň umožňuje studentům vytvořený kód pohodlně přenášet do jiných zařízení díky ukládání kódu do veřejného repozitáře na GitHubu.





## Literatura

- [1] (2016, Jul) Introducing redux. [Online]. Available: <<https://developer.ibm.com/tutorials/wa-manage-state-with-redux-p1-david-geary/>>
- [2] (2023, Aug) Rekvalifikace. [Online]. Available: <<https://www.uradprace.cz/rekvalifikace-1>>
- [3] J. Kudrna. (2023, Aug) Jak fungují rekvalifikace. [Online]. Available: <<https://www.youtube.com/watch?v=gjc7Irubgt0>>
- [4] (2023, Aug) Usnesení akreditační komise k akreditaci počítačových kurzů, nespécifických rekvalifikací a kurzů svařování. [Online]. Available: <<https://www.msmt.cz/vzdelavani/dalsi-vzdelavani/usneseni-akreditacni-komise-k-akreditace-pocitacovych-kurzu-a-nespecifickych-rekvalifikaci>>
- [5] (2023, Aug) 2513 vývojáři webu a multimédií. [Online]. Available: <<http://www.cz-isco.cz/isco/2513-vyvojari-webu-a-multimedii/>>
- [6] (2023, Aug) 62 - Činnosti v oblasti informačních technologií. [Online]. Available: <<http://www.nace.cz/62-cinnosti-oblasti-informacnich-technologii>>
- [7] (2023, Aug) Webdesigner/tvorba www stránek. [Online]. Available: <<https://mavo.eu/produkt/webdesigner-tvorba-www-stranek-17-4-2023-4-5-2023/>>
- [8] (2023, Aug) Seznam udělených akreditací rekvalifikačních programů 13.07.2023. [Online]. Available: <[https://www.msmt.cz/uploads/231/Rekvalifikace/Seznam\\_udelenych\\_akreditaci\\_rekvalifikacnich\\_programu\\_13.07.2023.xlsx](https://www.msmt.cz/uploads/231/Rekvalifikace/Seznam_udelenych_akreditaci_rekvalifikacnich_programu_13.07.2023.xlsx)>
- [9] (2023, Aug) Javascript - akreditovaný rekvalifikační kurz. [Online]. Available: <<https://www.itnetwork.cz/prace-a-podnikani-v-it/rekvalifikacni-kurzy/rekvalifikacni-kurzy-javascript>>

- [10] (2023, Aug) Programátor www aplikací. [Online]. Available: <<https://www.primakurzy.cz/kurz-programovani/#osnova>>
- [11] (2023, Aug) Kurz junior programátora. [Online]. Available: <<https://www.greenfoxacademy.cz/rekvalifikacni-kurzy-na-junior-programatora#details>>
- [12] (2023, Aug) Školení tvorba www stránek. [Online]. Available: <<http://skoleni.abecedapc.cz/rekvalifikace-webdesigner>>
- [13] (2023, Aug) Web development (full-stack). [Online]. Available: <<https://prahacoding.cz/courses/web-development-full-stack-course-praha-coding-school-2/>>
- [14] (2023, Aug) Tvorba webových stránek – rekvalifikace. [Online]. Available: <[https://www.nicom.cz/tvorba\\_webovych\\_stranek\\_nicom\\_tvorba\\_webovych\\_stranek\\_rekvalifikace-918542832-121534689-skoleni-rekvalifikacni-kurzy/](https://www.nicom.cz/tvorba_webovych_stranek_nicom_tvorba_webovych_stranek_rekvalifikace-918542832-121534689-skoleni-rekvalifikacni-kurzy/)>
- [15] (2023, Aug) Tvorba www stránek. [Online]. Available: <<https://www.edu-partners.cz/tvorba-www-stranek-rekvalifikacni-kurz-80-hodin-online-prezencne-praha-ostrava/>>
- [16] (2023, Aug) Digitální akademie: Web. [Online]. Available: <<https://www.czechitas.cz/kurzy/digitalni-akademie-web>>
- [17] (2023, Aug) Kam na střední školu? [Online]. Available: <<https://www.stredniskoly.cz>>
- [18] (2023, Aug) Vyber si z nabídky 204 vysokých škol. [Online]. Available: <<https://www.vysokeskoly.cz>>
- [19] (2023, Aug) Trainee program v lokalitě Česko. [Online]. Available: <<https://cz.linkedin.com/jobs/trainee-program-jobs?position=1&pageNum=0>>
- [20] (2023, Aug) David Šetek - hackni svou budoucnost. [Online]. Available: <<https://www.youtube.com/@hacknisvoubudoucnost>>
- [21] (2023, Aug) Rob web s yablkom. [Online]. Available: <<https://www.youtube.com/@RobWebSK>>
- [22] (2023, Aug) Co je to gamifikace? [Online]. Available: <<https://www.youtube.com/@jaknaweby>>
- [23] (2023, Aug) developedbyed. [Online]. Available: <<https://www.youtube.com/@developedbyed>>
- [24] (2023, Aug) Co je to gamifikace? [Online]. Available: <<https://it-slovník.cz/pojem/gamifikace/>>



- [25] (2023, Aug) Create stories, games, and animations. [Online]. Available: <https://scratch.mit.edu>
- [26] (2023, Aug) Learn to code through the power of play. [Online]. Available: <https://codecombat.com>
- [27] (2023, Aug) Play coding games. have fun. build skills. [Online]. Available: <https://www.codingame.com/start/>
- [28] (2023, Aug) Achieve mastery through challenge. [Online]. Available: <https://www.codewars.com>
- [29] (2023, Aug) Improving lives through learning. [Online]. Available: <https://www.udemy.com>
- [30] (2023, Aug) Learn without limits. [Online]. Available: <https://www.coursera.org>
- [31] (2023, Aug) Codecademy. [Online]. Available: <https://www.codecademy.com>
- [32] (2023, Aug) Learn to code — for free. [Online]. Available: <https://www.freecodecamp.org>
- [33] M. Laurenčík, *Tvorba www stránek v HTML a CSS*. Grada, 2019.
- [34] J. Řezáč, *Web ostrý jako břitva*. Web ostrý jako břitva, 2014.
- [35] M. Bureš, R. Miroslav, and D. Michal, *Efektivní testování softwaru*. Grada, 2016.
- [36] (2023, Aug) Smashing magazine. [Online]. Available: <https://www.smashingmagazine.com>
- [37] (2023, Aug) Frontend mag. [Online]. Available: <https://www.frontendmag.com>
- [38] (2023, Aug) Google scholar. [Online]. Available: <https://scholar.google.com>
- [39] (2023, Aug) File types indexable by google. [Online]. Available: <https://developers.google.com/search/docs/crawling-indexing/indexable-file-types>
- [40] (2023, Aug) Stack overflow. [Online]. Available: <https://stackoverflow.com>
- [41] (2023, Aug) IntelliJ idea. [Online]. Available: <https://www.jetbrains.com/idea/>
- [42] (2023, Aug) Visual studio code. [Online]. Available: <https://code.visualstudio.com>

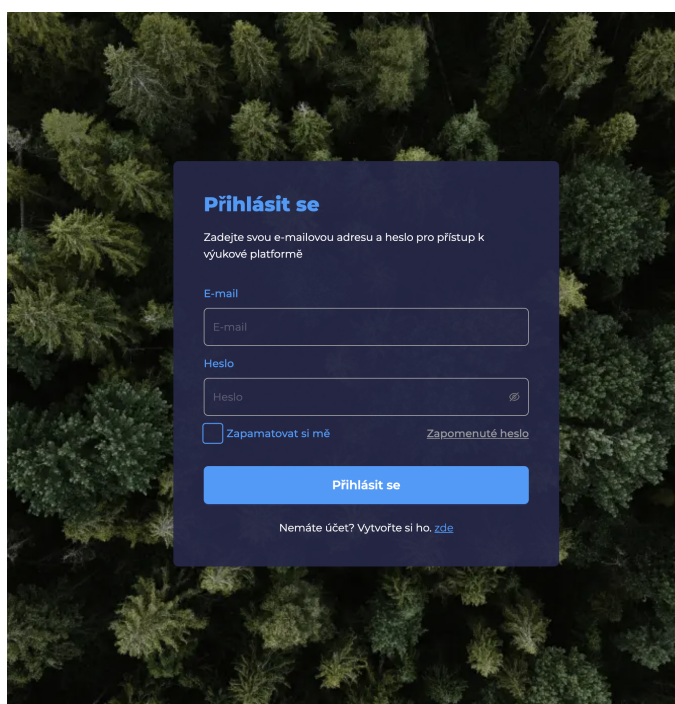
- [43] (2023, Aug) JsFiddle. [Online]. Available: <<https://jsfiddle.net>>
- [44] (2023, Aug) Codepen. [Online]. Available: <<https://codepen.io>>
- [45] (2023, Aug) Codesandbox. [Online]. Available: <<https://codesandbox.io>>
- [46] (2023, Aug) Playcode. [Online]. Available: <<https://playcode.io>>
- [47] (2023, Aug) Live share. [Online]. Available: <<https://marketplace.visualstudio.com/items?itemName=MS-vsliveshare.vsliveshare>>
- [48] (2023, Aug) Code with me. [Online]. Available: <<https://www.jetbrains.com/code-with-me/>>
- [49] (2023, Aug) Tutorials point. [Online]. Available: <<https://www.tutorialspoint.com/codingground.htm>>
- [50] (2023, Aug) Jdoodle. [Online]. Available: <<https://www.jdoodle.com/online-java-compiler/>>
- [51] (2023, Aug) Python online compiler. [Online]. Available: <<https://www.programiz.com/python-programming/online-compiler/>>
- [52] (2023, Aug) Onlinegdb. [Online]. Available: <<https://www.onlinegdb.com>>
- [53] (2023, Aug) Online java ide. [Online]. Available: <<https://www.online-java.com>>
- [54] (2023, Aug) Online python ide. [Online]. Available: <<https://www.online-python.com>>
- [55] (2023, Aug) Sql online. [Online]. Available: <<https://sqliteonline.com>>
- [56] J. Evers, “Javascript opens doors to browser-based attacks,” *CNET*, Aug 2023. [Online]. Available: <<https://www.cnet.com/news/privacy/javascript-opens-doors-to-browser-based-attacks/>>
- [57] (2023, Aug) Webassembly. [Online]. Available: <<https://developer.mozilla.org/en-US/docs/WebAssembly>>
- [58] M. Levlin. (2020) Dom benchmark comparison of the front-end javascript frameworks react, angular, vue, and svelte. [Online]. Available: <[https://www.doria.fi/bitstream/handle/10024/177433/levlin\\_mattias.pdf?seque](https://www.doria.fi/bitstream/handle/10024/177433/levlin_mattias.pdf?seque)>
- [59] (2023, Aug) Typescript. [Online]. Available: <<https://www.typescriptlang.org>>
- [60] (2023, Aug) React. [Online]. Available: <<https://react.dev>>

- [61] (2023, Aug) Redux. [Online]. Available: <<https://redux.js.org>>
- [62] (2023, Aug) Ant design 5.0. [Online]. Available: <<https://ant.design>>
- [63] (2023, Aug) Css for the <component> age. [Online]. Available: <<https://styled-components.com>>
- [64] (2023, Aug) Monaco - the editor of the web. [Online]. Available: <<https://microsoft.github.io/monaco-editor/>>
- [65] (2023, Aug) Codemirror. [Online]. Available: <<https://codemirror.net>>
- [66] E. Corporation DBA Tiny Technologies. (2023, Aug) Tinymce. [Online]. Available: <<https://github.com/tinymce/tinymce>>
- [67] Meta. (2023, Aug) Draft.js. [Online]. Available: <<https://github.com/facebookarchive/draft-js>>
- [68] (2023, Aug) Prosemirror. [Online]. Available: <<https://prosemirror.net>>
- [69] J. Chen and M. Byron. (2023, Aug) Quill. [Online]. Available: <<https://quilljs.com>>
- [70] (2023, Aug) Můj cloud. [Online]. Available: <<https://www.muji.cloud>>
- [71] (2023, Aug) Wedos. [Online]. Available: <<https://www.wedos.com/cs/webhosting/>>
- [72] (2023, Aug) ebola.cz. [Online]. Available: <<https://www.ebola.cz>>
- [73] (2023, Aug) Active24.cz. [Online]. Available: <<https://www.active24.cz/webhosting>>
- [74] (2023, Aug) Český hosting. [Online]. Available: <<https://www.cesky-hosting.cz/webhosting/>>
- [75] (2023, Aug) Digitalocean. [Online]. Available: <<https://www.digitalocean.com>>
- [76] (2023, Aug) Render. [Online]. Available: <<https://render.com>>
- [77] (2023, Aug) Heroku. [Online]. Available: <<https://www.heroku.com/>>
- [78] (2023, Aug) Azure. [Online]. Available: <<https://azure.microsoft.com/en-us/>>
- [79] (2023, Aug) Aws. [Online]. Available: <<https://aws.amazon.com/ec2/>>
- [80] (2023, Aug) Node.js. [Online]. Available: <<https://nodejs.org>>

- [81] (2023, Aug) The api and real-time application framework. [Online]. Available: <<https://feathersjs.com>>
- [82] L. P. Pinca and contributors. (2023, Aug) ws: a node.js websocket library. [Online]. Available: <<https://github.com/websockets/ws>>
- [83] S.-y. Guo, M. Ficarra, and G. Kevin. (2024) The json object. [Online]. Available: <<https://tc39.es/ecma262/multipage/structured-data.html#sec-json-object>>
- [84] L. Rui and D. Li, “A new operational transformation framework for real-time group editors,” pp. 307 – 319, 2007.
- [85] N. Smith and G. Joseph. (2023, Aug) Sharedb. [Online]. Available: <<https://github.com/share/sharedb>>
- [86] (2023, Aug) What is postgresql? [Online]. Available: <<https://www.postgresql.org/about/>>
- [87] (2023, Aug) Bitbucket. [Online]. Available: <<https://bitbucket.org/product/>>
- [88] (2023, Aug) Github. [Online]. Available: <<https://github.com/about>>
- [89] B. Buxton, *Sketching User Experiences*. Morgan Kaufmann, 2010.
- [90] C. Ware, *Information Visualization, perception for design*. Morgan Kaufmann, 2019.
- [91] E. Goodman, K. Mike, and M. Andrea, *Observing the User Experience*, 2nd ed. Morgan Kaufmann, 2012.
- [92] T. Koomen, B. Broekman, L. van der Aalst, and M. Vroon, *TMap next: for result-driven testing*. Uitgeverij kleine Uil, 2013.
- [93] M. Disman, *Jak se vyrábí sociologická znalost: Příručka pro uživatele*. Karolinum, 2002.

# Příloha A

## Screenshots platformy



Obrázek A.1: Screenshot přihlašovacího formuláře

## A. Screenshoty platformy

The registration form is titled "Zaregistrujte se" and is set against a dark blue background with a forest image. It contains the following fields and elements:

- Křestní jméno:** Input field with placeholder "Křestní jméno".
- Příjmení:** Input field with placeholder "Příjmení".
- E-mail:** Input field with placeholder "E-mail".
- Heslo:** Input field with placeholder "Heslo" and a visibility icon.
- Potvrdit heslo:** Input field with placeholder "Potvrdit heslo" and a visibility icon.
- Consent:** A checkbox followed by the text "Souhlasím s [Podmínky používání](#) and [Zásady ochrany osobních údajů](#)".
- Submit:** A large blue button labeled "Zaregistrujte se".
- Link:** Text "Už máte účet? Přihlaste se [zde](#)".

Obrázek A.2: Screenshot formuláře pro registraci lektora

The screenshot shows the "JavaScript" course administration page. The left sidebar contains a navigation menu with items like "JavaScript", "Administrace", "Úvod", "Proměně", "Tvorba www stránek", "PHP", "PHP 2", and "HTML". The main content area is titled "JavaScript" and shows a table of lessons.

Stav	Název	Popis	Akce
✓	Úvod	Jak to všechno funguje	<a href="#">Upravit</a> <a href="#">Odstranit</a>
	Proměně	Základní stavební kámen	<a href="#">Upravit</a> <a href="#">Odstranit</a>

Buttons: "Vytvořit kurz", "Lekce", "Studenti", "Přidat lekci".

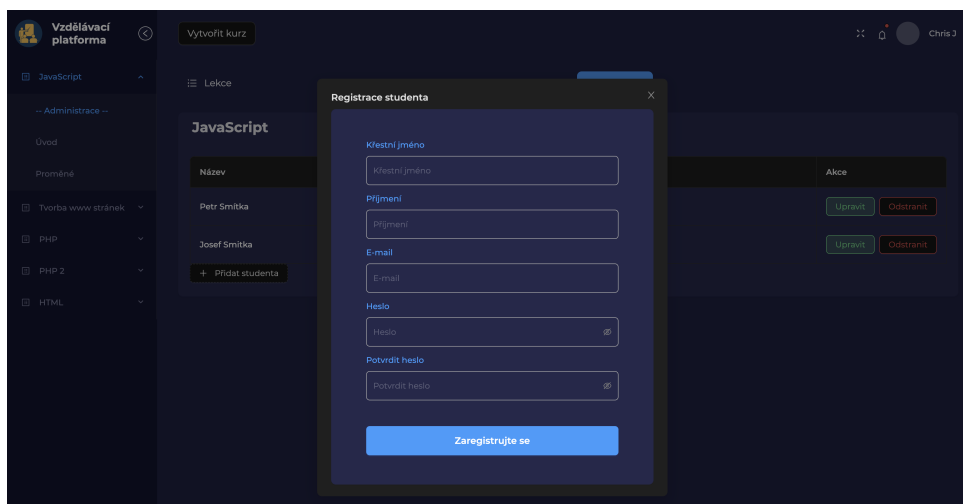
Obrázek A.3: Screenshot administrace lekcí

The screenshot shows the "JavaScript" course administration page for students. The left sidebar is the same as in the previous screenshot. The main content area is titled "JavaScript" and shows a table of students.

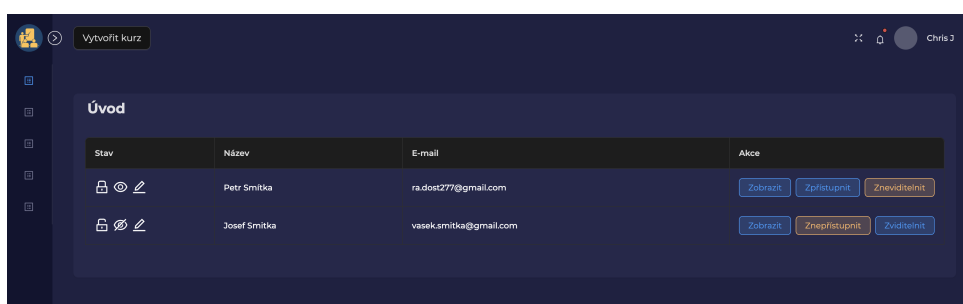
Název	E-mail	Akce
Petr Smitka	ra.dest277@gmail.com	<a href="#">Upravit</a> <a href="#">Odstranit</a>
Josef Smitka	vasek.smitka@gmail.com	<a href="#">Upravit</a> <a href="#">Odstranit</a>

Buttons: "Vytvořit kurz", "Lekce", "Studenti", "Přidat studenta".

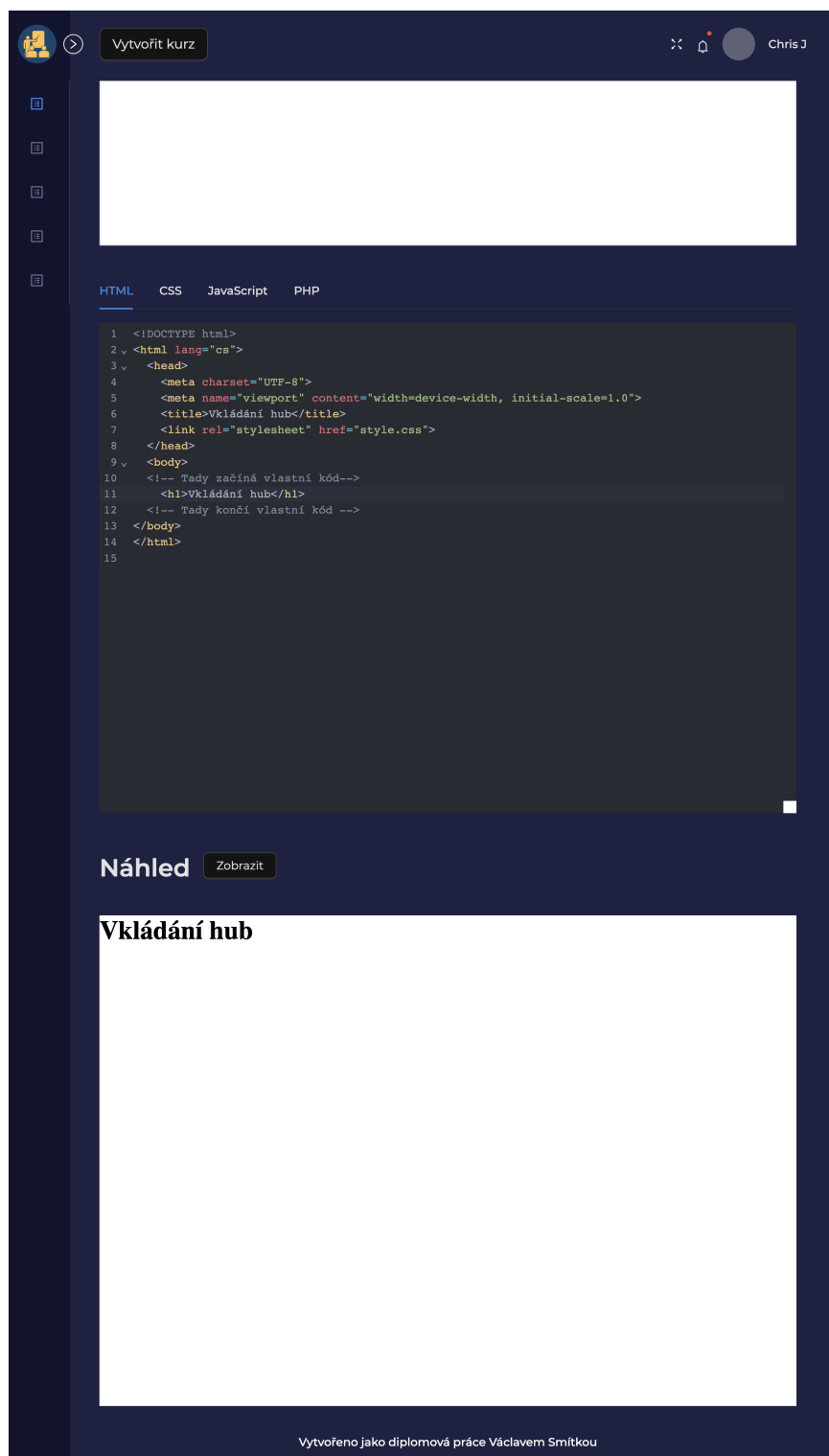
Obrázek A.4: Screenshot administrace studentů



Obrázek A.5: Screenshot registračního formuláře studenta

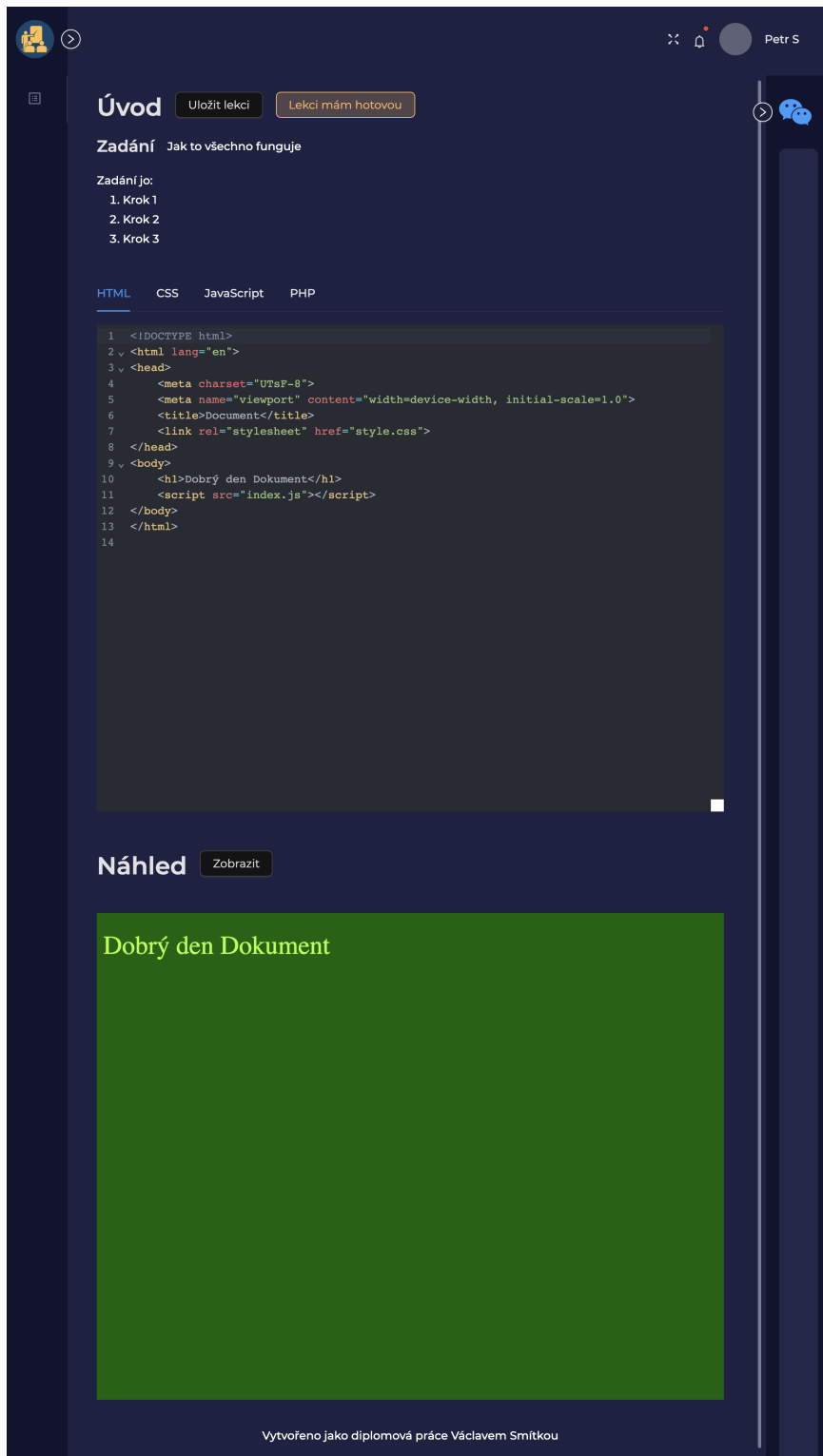


Obrázek A.6: Screenshot detailu lekce z pohledu lektora

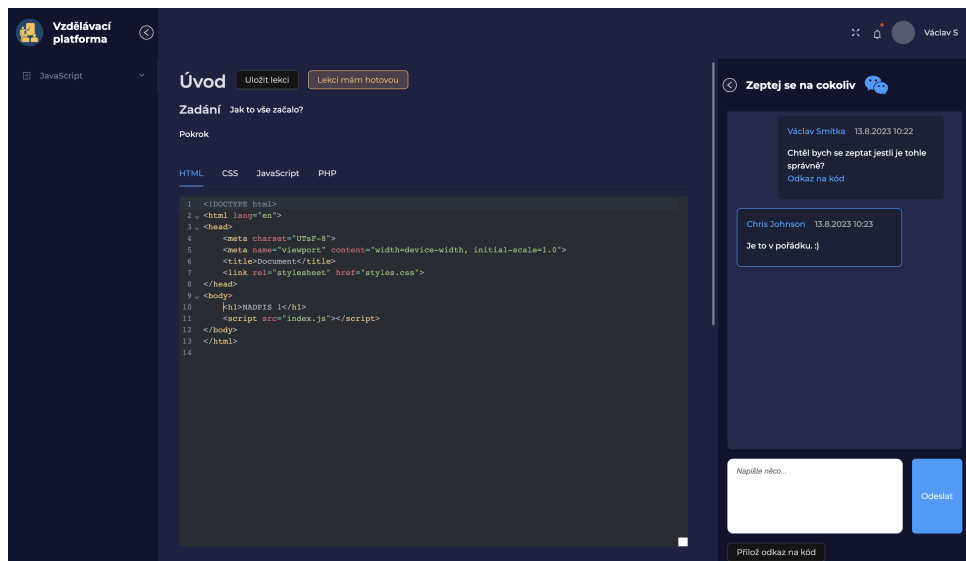


Obrázek A.7: Screenshot editoru z pohledu lektora při přípravě cvičení





Obrázek A.8: Screenshot editoru z pohledu studenta



Obrázek A.9: Screenshot chatovacího vlákna z pohledu studenta

## Vítejte na Vzdělávací platformě

Do platformy se připojíte pomocí tohoto emailu a heslo vám předá lektor.

[Přihlaste se do platformy](#)

Obrázek A.10: Screenshot mailu příchozí po registraci studenta

## Student Petr Smítka dokončil lekci Úvod

[Otevřít řešení](#)

Obrázek A.11: Screenshot mailu příchozí po odevzdání cvičení

```
import { Knex } from 'knex'

export async function up(knex: Knex): Promise<void> {
  await knex.schema.createTable('users', (table) => { ...
  })

  await knex.schema.createTable('courses', (table) => { ...
  })

  await knex.schema.createTable('lessons', (table) => { ...
  })

  await knex.schema.createTable('course-users', (table) => { ...
  })

  await knex.schema.createTable('lesson-users', (table) => { ...
  })

  await knex.schema.createTable('notification', (table) => { ...
  })

  await knex.schema.createTable('messages', (table) => { ...
  })
}

export async function down(knex: Knex): Promise<void> {
  await knex.schema.dropTable('lessons')
  await knex.schema.dropTable('courses')
  await knex.schema.dropTable('users')
  await knex.schema.dropTable('course-users')
  await knex.schema.dropTable('lesson-users')
  await knex.schema.dropTable('notification')
  await knex.schema.dropTable('messages')
}
```

Obrázek A.12: Screenshot struktury migračního souboru pro založení databáze



## Příloha B

### Testovací scénáře

#### B.1 Registrace lektora

ID	TC-1
Název	Registrace lektora
Aktér	Lektor
Předpoklad	Znalost url adresy platformy
Účel	Lze se zaregistrovat a přihlásit do platformy jako lektor
Výsledek	Úspěšné přihlášení do platformy jako lektor

##### B.1.1 Kroky

ID	Akce	Očekávaný výsledek
TC-1-1	Otevření webové stránky platformy v Chrome prohlížeči	Zobrazí se stránka s možností přihlásit se a zaregistrovat
TC-1-2	Vyplnění svého e-mailu a hesla: 123 a kliknutí na tlačítko přihlásit	Objeví se červená notifikace informující o nemožnosti se přihlásit
TC-1-3	Kliknutí na tlačítko registrovat se zde	Překlikne se na novou stránkou s registračním formulářem
TC-1-4	Vyplnění formuláře se svým e-mailem a heslem: 123 a kliknutí na tlačítko odeslat	Překlikne se na stránku s přihlašovací formulářem
TC-1-5	Vyplnění svého e-mailu a hesla: 123 a kliknutí na tlačítko přihlásit	Překlikne se na úvodní stránku lektora, odkud je možné vytvářet a spravovat kurzy

## B.2 Tvorba kurzu

ID	TC-2
Název	Tvorba kurzu
Aktér	Lektor
Předpoklad	Vytvořený lektorský účet
Účel	Lze projít úspěšně procesem tvorby kurzu
Výsledek	Úspěšné vytvoření kurzu včetně lekcí

### B.2.1 Kroky

ID	Akce	Očekávaný výsledek
TC-2-1	Přihlášení v Chrome prohlížeči	Zobrazí se úvodní stránku lektora odkud je možné vytvářet a spravovat kurzy
TC-2-2	Kliknutí na tlačítko vytvořit kurz	Objeví stránka s možností zadat jméno, popis a vybrat šablonu kurzu
TC-2-3	Kliknutí na tlačítko další	Zabarví se červeně jméno a popis kurzu a zobrazí se hláška, že údaje jsou povinné
TC-2-4	Vyplnění jména, popisku a kliknutí na tlačítko další	Zobrazí se další část formuláře, kde se zadávají lekce
TC-2-5	Kliknutí na tlačítko přidat lekci a kliknutí na tlačítko další	Zabarví se červeně jméno a popis kurzu a zobrazí se hláška, že údaje jsou povinné
TC-2-6	Vyplnění názvu a popisku lekce a kliknutí na další	Zobrazí se shrnutí obsahující vyplněné informace v předšlých krocích
TC-2-7	Kliknutí na tlačítko hotovo	Zobrazí se úvodní stránka platformy pro lektora a v bočním panelu se zobrazí záložka nového kurzu

## B.3 Tvorba kurzu se šablonou

ID	TC-3
Název	Tvorba kurzu se šablonou
Aktér	Lektor
Předpoklad	Vytvořený lektorský účet a alespoň jeden kurz
Účel	Lze projít úspěšně procesem tvorby kurzu včetně využití šablony
Výsledek	Úspěšné vytvoření kurzu včetně lekcí

### B.3.1 Kroky

ID	Akce	Očekávaný výsledek
TC-3-1	Přihlášení v Chrome prohlížeči	Zobrazí se úvodní stránku lektora, odkud je možné vytvářet a spravovat kurzy
TC-3-2	Kliknutí na tlačítko vytvořit kurz	Objeví se stránka s možností zadat jméno, popis a vybrat šablonu kurzu
TC-3-3	Vyplnění jména, popisku a kliknutí na tlačítko další	Zobrazí se další část formuláře, kde se zadávají lekce
TC-3-4	Kliknutí na tlačítko předchozí	Zobrazí se předchozí stránka s vyplněnými údaji
TC-3-5	Ve výběru šablon se jedna vybere a klikne se na tlačítko další	Na další stránce se zobrazí názvy a popisky lekcí vybraného kurzu
TC-3-6	Kliknutí na tlačítko další	Zobrazí se stránka se shrnutím všech vyplněných údajů včetně informace o vybrané šabloně
TC-3-7	Kliknutí na tlačítko předchozí	Zobrazí se stránka s lekcemi, tak, jak byly připravené
TC-3-8	Kliknutí na tlačítko přidat lekci a kliknutí na tlačítko další	Zabarví se červeně jméno a popis kurzu a zobrazí se hláška, že údaje jsou povinné
TC-3-9	Vyplnění názvu a popisku lekce a kliknutí na další	Zobrazí se shrnutí obsahující všechny vyplněné informace v předešlých krocích
TC-3-10	Kliknutí na tlačítko hotovo	Zobrazí se úvodní stránka platformy pro lektora a v bočním panelu se zobrazí záložka nového kurzu

## B.4 Vložení studenta do kurzu

<b>ID</b>	<b>TC-4</b>
Název	Vložení studenta do kurzu
Aktér	Lektor
Předpoklad	Vytvořený lektorský účet a alespoň jeden vytvořený kurz
Účel	Lze přidat studenta do kurzu
Výsledek	Student se zobrazí v seznamu studentů i v seznamu řešitelů jednotlivých lekcí

### B.4.1 Kroky

ID	Akce	Očekávaný výsledek
TC-4-1	Přihlášení v Chrome prohlížeči	Zobrazí se úvodní stránka lektora, odkud je možné vytvářet a spravovat kurzy
TC-4-2	V bočním panelu vybrat kurz a kliknout na administraci	Zobrazí se tabulka se seznamem lekcí kurzu – musí obsahovat přesně tolik lekcí, kolik jich bylo vytvořeno
TC-4-3	Kliknutí na tlačítko studenti	Zobrazí se tabulka se seznamem studentů – bude prázdná
TC-4-4	Kliknutí na tlačítko přidat studenta	Zobrazí se modální okno s registračním formulářem pro studenta
TC-4-5	Kliknutí na tlačítko zaregistruj se	Zabarví se červeně jméno a popis kurzu a zobrazí se hláška, že údaje jsou povinné
TC-4-6	Vyplnění formuláře s rozdílnými hesly	Zabarví se červeně druhé vstupní pole pro heslo s hláškou, že hesla se musí shodovat
TC-4-7	Vyplnění nevalidního e-mailu	Zabarví se červeně vstupní pole pro e-mail a zobrazí se hláška, že se jedná o nevalidní e-mail
TC-4-8	Vyplnění správně všech údajů a kliknutí na tlačítko registruj se	Modální okno se zavře a v tabulce se zobrazí nový uživatel



## B.5 Přípravení lekce

ID	TC-5
Název	Přípravení lekce
Aktér	Lektor
Předpoklad	Vytvořený lektorský účet a alespoň jeden vytvořený kurz
Účel	Lze vytvořit zadání lekce a připravit šablonu kódu
Výsledek	Lekce se nastaví jako připravena

### B.5.1 Kroky

ID	Akce	Očekávaný výsledek
TC-5-1	Přihlášení v Chrome prohlížeči	Zobrazí se úvodní stránka lektora, odkud je možné vytvářet a spravovat kurzy
TC-5-2	V bočním panelu vybrat kurz a kliknout na administraci	Zobrazí se tabulka se seznamem lekcí kurzu – musí obsahovat přesně tolik lekcí, kolik jich bylo vytvořeno. Zároveň všechny lekce v prvním sloupečku tabulky mají ikonu tužky
TC-5-3	Kliknutí na tlačítko upravit u první lekce	Zobrazí se stránka s editorem. Odpovídá název a popis lekce.
TC-5-4	Vyplnění obecného textu do zadání a odchod z platformy na stránku administrace kurzu	Ikona tužky je stále v prvním sloupečku u prvního cvičení
TC-5-5	Kliknutí na tlačítko upravit u první lekce	Zobrazí se stránka s editorem. Odpovídá název a popis lekce a zadaný text z kroku TC-5-4.
TC-5-6	Vyplnění zadání včetně seznamu a vyplnění pár řádků kódu do kód editoru a následné kliknutí na tlačítko uložit lekci	Zobrazí se v pravém horním rohu zelené notifikace uvědomující o úspěšném uložení lekce
TC-5-7	Odchod z platformy na stránku administrace kurzu	Ikona tužky je stále v prvním sloupečku u prvního cvičení
TC-5-8	Kliknutí na tlačítko upravit lekci	Zobrazí se stránka editoru včetně všech vyplněných náležitostí ve stejném formátu

ID	Akce	Očekávaný výsledek
TC-5-9	Kliknutí na tlačítko lekce je připravena a odchod na stránku administrace kurzu	Po kliknutí na tlačítko lekce je připravena se objeví zelené notifikace uvědomující o uložení a přípravě lekce. Na stránce seznamu lekcí je místo tužky ikona fajfky

## B.6 Editování a viditelnost

ID	TC-6
Název	Editování a viditelnost
Aktér	Lektor
Předpoklad	Vytvořený lektorský účet, vytvořený alespoň jeden kurz a přidán alespoň jeden student
Účel	Lze zviditelnit, zneviditelnit, zakázat a dovolit editování studentovi lektorem
Výsledek	Podle kliknutí se mění styl tlačítek v administrátorské sekci a zároveň se podle toho chová studentské prostředí

### B.6.1 Kroky

ID	Akce	Očekávaný výsledek
TC-6-1	Přihlášení v Chrome prohlížeči	Zobrazí se úvodní stránka lektora, odkud je možné vytvářet a spravovat kurzy
TC-6-2	V bočním panelu vybrat kurz a kliknout na název první lekce	Zobrazí se tabulka se seznamem studentů kuru – alespoň jeden. V prvním sloupečku všichni studenti mají ikonky: zavřený zámeček, otevřené oko a otazník.
TC-6-3	Kliknutí na tlačítko zobrazit	Zobrazí se stránka s editorem. Název, popis, text zadání a obsah editoru odpovídá zadaným údajům v testu B.5
TC-6-4	Překliknutí zpět na administraci lekce	Vše zůstalo stejně jako v kroku TC-6-2

ID	Akce	Očekávaný výsledek
TC-6-5	Kliknutí na tlačítko zneviditelnit	Ikona oka se vymění za přeškrtnuté oko a místo oranžového tlačítka zneviditelnit se zobrazí modré tlačítko zviditelnit.
TC-6-6	Kliknutí na tlačítko zviditelnit	Ikona oka se vymění za oko a místo modrého tlačítka zviditelnit se zobrazí oranžové tlačítko zneviditelnit
TC-6-7	Kliknutí na tlačítko zpřístupnit	Ikona zavřeného zámku se změní na otevřený zámek a místo modrého tlačítka zpřístupnit se zobrazí oranžové tlačítko znepřístupnit
TC-6-8	Kliknutí na tlačítko znepřístupnit	Ikona zavřeného zámku se změní na zavřený zámek a místo oranžového tlačítka znepřístupnit se zobrazí modré tlačítko zpřístupnit
TC-6-9	Kliknutí na tlačítko zobrazit	Zobrazí se stránka s editorem. Název, popis, text zadání a obsah editoru odpovídá zadaným údajům v testu B.5
TC-6-10	Překliknutí zpět na administraci lekce	Vše zůstalo stejné jako v kroku TC-6-2
TC-6-7	Kliknutí na tlačítko zpřístupnit	Ikona zavřeného zámku se změní na otevřený zámek a místo modrého tlačítka zpřístupnit se zobrazí oranžové tlačítko znepřístupnit



## Příloha C

### Zpětná vazba z testování

Nasbíraná zpětná vazba účastníků ze všech částí testování je rozdělena podle toho, čeho se zpětná vazba týká a je seřazena podle priority.

#### C.1 Chyby

Priorita chyby je určována podle kritičnosti. Pokud se chyba týká základních funkcionalit platformy, zabraňující studentovi pracovat na zadaném úkolu, je označena jako vysoká. Když se jedná o drobný nedostatek, který spíše činí práci studenta méně pohodlnou, je označena za nízkou.

Účastník	Setkání/Scénář	Priorita	Popis
Míša	1. setkání	Vysoká	Chyba při ukládání kódu zapříčinila rozhození formátování a rozbití HTML struktury
Míša	1. setkání	Vysoká	Chyba při formátování vnořených seznamů v zadání – seznam nevytvoří podseznam
Vítek	1. setkání	Střední	Při napsání delší zprávy se přeskočí odesílací tlačítko pod textové pole
Vítek	1. setkání	Střední	V profilu se zobrazují kolonky pro adresu
Dominik	TC-2	Střední	Při zavření levého menu nelze rozlišit mezi lekce
Dominik	TC-2	Střední	Nízký kontrast tlačítka pro přidání lekce
Dominik	TC-4	Střední	Nízký kontrast tlačítka pro přidání studenta
Dominik	TC-6	Střední	Po aktualizaci stavu se změní i pořadí studentů v tabulce

## C.2 Podněty k vylepšení

Účastník	Setkání/Scénář	Priorita	Popis
Petra	1. setkání	Vysoká	Po přijetí zprávy se nezobrazí žádná notifikace
Tomáš	TC-1	Vysoká	Obohacení úvodní lektorovi stránky o přehled postupu studentů skrz kurz
Petra	1. setkání	Střední	Aktualizace náhledu v reálném čase
Tomáš	1. setkání	Střední	Aktualizace náhledu pomocí klávesové zkratky
Dominik	TC-5	Střední	5000 znaků pro sepsání zadání je málo
Dominik	2. setkání	Nízká	Změna ikony otazníku na něco víc odpovídajícího rozdělané práci
Dominik	TC-6	Nízká	Změnit barvy tlačítek
Tomáš	2. setkání	Nízká	Přidání světlého tématu
Dominik	1. setkání	Nízká	Zobrazovat nějaký progress bar nebo jiným způsobem ukázat studentovi kolik ho toho ještě čeká a kolik toho zvládl
Veronika	1. setkání	Nízká	Přidat náročnost cvičení

## Příloha D

### Struktura příloh elektronické verze práce

Součástí elektronické verze práce jsou zdrojové soubory Vzdělávací platformy. V souboru „README.md“ je návod jak platformu zprovoznit. Před lokálním spuštěním platformy je nutné si vytvořit soubory pro proměnné prostředí:

- .env
- src/server/.env

Strukturu lze zkopírovat z příložených souborů s názvy „.env.example“ ve stejném adresáři. Proměnná „NODE\_ENV“ musí být nastavena na „dev“ pro použití SQLite místo PostgreSQL.