



Střední průmyslová škola strojní
a elektrotechnická a Vyšší odborná škola,
Liberec 1, Masarykova 3

SJEDNOCENÍ UŽIVATELSKÝCH ROZHRANÍ WWW APLIKACÍ PRÁCE A PRAXE

Maturitní práce

Autor	Ondřej Svoboda
Obor	Technické lyceum
Vedoucí práce	Ing. Tomáš Kazda
Školní rok	2021/2022

Přihláška k maturitní práci

Jméno a příjmení studenta

Svoboda, Ondřej

Název práce

Sjednocení uživatelských rozhraní www aplikací PráCe a PraXe

Třída

L4

Školní rok

MP2021/22

Přidělené role

Vedoucí práce

Oponent

Podpis



Kazda, Tomáš

Obecná ustanovení	Vypracování a odevzdání práce proběhne v souladu s platnými normami (vyhláška 177/2009 Sb.) a aktuálním dokumentem "Pokyny k vypracování prací" vydaným školou.
	Práce bude hodnocena z hlediska jejího praktického využití, zvládnutí dokumentace po věcné i formální stránce a obhajoby celé práce. Student byl seznámen s kritérii hodnocení maturitní práce.
	Práce bude odevzdána ve dvou stejnopisech vázaných pevnou nebo kroužkovou vazbou.
	Veškeré náklady na MP včetně vyhotovení obou tištěných kopií si student hradí sám.
Licenční ujednání	Ve smyslu § 60 (Školní dílo) autorského zákona č. 121/2000 Sb. poskytují SPŠSE a VOŠ Liberec výhradní a neomezená práva k využití této mé maturitní práce.
	Bez svolení školy se zdržím jakéhokoliv komerčního využití mé práce.
	Pro výukové účely a prezentaci školy se vzdávám nároku na odměnu za užití díla.

Finanční rozvaha - odhad celkových nákladů

V Kč	Náklady celkem	Hrazené školou
Výrobní	0	0
Na služby	0	0

Jedná se o MP, jejíž vypracování si škola vyžádala? Ano – Ne

Podpis studenta (vyjadřuje souhlas s uvedenými údaji a ujednáními)

V Liberci 05.10.2021

Podpis



Konzultant

Práci podporuji

Podpis

Předmětová komise

Práci doporučuji

Podpis



Třídní učitel

Práci doporučuji

Podpis



Garant oboru

Práci doporučuji

Podpis



Ředitel školy

Práci doporučuji

Podpis



Zadání maturitní práce

Název

Sjednocení uživatelských rozhraní www aplikací PráCe a PraXe

Předmět

WEB

Téma

Cílem práce je optimalizace UI aplikace pro evidenci a vedení dlouhodobých prací. Dále úprava UX aplikace pro evidenci praxí tak, aby její rozhraní mělo stejné funkční (a případně vizuální) prvky jako aplikace pro dlouhodobé práce.

Návrh řešení by měl probíhat dle metodiky BEM a při vhodném zakomponování tak může vzniknout "katalog" komponent využitelných při tvorbě dalších školních aplikací...

Použité prostředky

IT

Cíle práce

1	Návrh úprav UI aplikace PráCe
2	Návrh úprav UI/UX aplikace PraXe
3	Realizace a implementace zvolených komponent webu

Osnova práce

1	Metodiky pro návrh a vývoj designu webových stránek
2	Specifika webových aplikací
3	Dokumentace vybrané metodiky
4	Analýza stávajícího stavu PráCe a PraXe
5	Identifikace a návrh komponent
6	Návrh designu a volba technické implementace
7	XD návrh společného layoutu
8	Realizace a nasazení vybraných komponent
9	Vytvoření katalogu včetně dokumentace komponent pro obecné použití

Anotace

Práce se zabývá optimalizací uživatelského rozhraní školní aplikace pro vedení a správu dlouhodobých prací. Dále také úpravou UX aplikace pro evidenci praxí, tak aby mělo stejné vizuální a funkční prvky. Měl by vzniknout katalog „komponentů“, neboli prvků, které budou využitelné při tvorbě dalších školních aplikací.

Summary

The thesis deals with the optimization of the user interface of a school application for conducting and managing long-term works. Furthermore, it also modifies the UX of the application for registering work experience so that it has the same visual and functional elements. A catalogue of "components", or elements that will be used in the creation of other school applications, should be created.

Čestné prohlášení

Prohlašuji, že jsem předkládanou maturitní práci vypracoval sám a uvedl jsem veškerou použitou literaturu a bibliografické citace.

V Liberci dne 13.03.2022

.....

Ondřej Svoboda

Obsah

Úvod.....	1
1 Specifika webových aplikací	2
1.1 Statické webové stránky.....	2
1.2 Webové aplikace.....	3
1.3 Front-end.....	3
1.3.1 React.....	4
1.3.2 Angular	4
1.4 Back-end	5
1.4.1 PHP	5
1.4.2 Node.js	5
1.4.3 ASP.NET	6
1.4.4 Ruby on Rails	6
2 Návrh a vývoj designu stránek.....	7
2.1.1 Layout.....	7
2.1.2 Typografie	7
3 Návrh úprav aplikací PráCe a PraXe	8
3.1 Analýza	8
3.1.1 Aplikace PraXe.....	8
3.1.2 Aplikace PráCe.....	10
3.2 Design	10
3.3 Návrh úprav.....	11
3.3.1 Fonty	11
3.3.2 Navigace.....	12
3.3.3 Barevná paleta.....	12
3.3.4 Design stránek.....	13
4 Návrh komponent webu	14
4.1 Identifikace komponent.....	14
4.2 Příprava a kódování	14

4.2.1	Závislosti.....	16
4.3	Stylování komponent.....	18
4.3.1	Styled Components.....	18
4.3.2	Bootstrap.....	20
4.3.3	CSS Modules	20
5	Katalog komponentů.....	21
5.1.1	Card.....	22
5.1.2	Table.....	22
5.1.3	Button	22
5.1.4	Message.....	23
5.1.5	Header	23
	Závěr.....	24
	Seznam zkratk a odborných výrazů.....	25
	Seznam obrázků.....	27
	Použité zdroje	28
A.	Seznam příložených souborů	I

Úvod

Již několik let se věnuji tvorbě webových stránek. Tato záliba vznikla ještě před začátkem mého studia na střední škole. Na škole jsem se naučil práci s jazyky HTML, CSS a základy Javascriptu a prohloubil své znalosti tvorby webových stránek. Minulý rok jsem měl možnost dělat jako ročníkovou práci web pro Libereckou geodetickou firmu Geoline CZ, což mi dalo nové zkušenosti v práci v reálném světě.

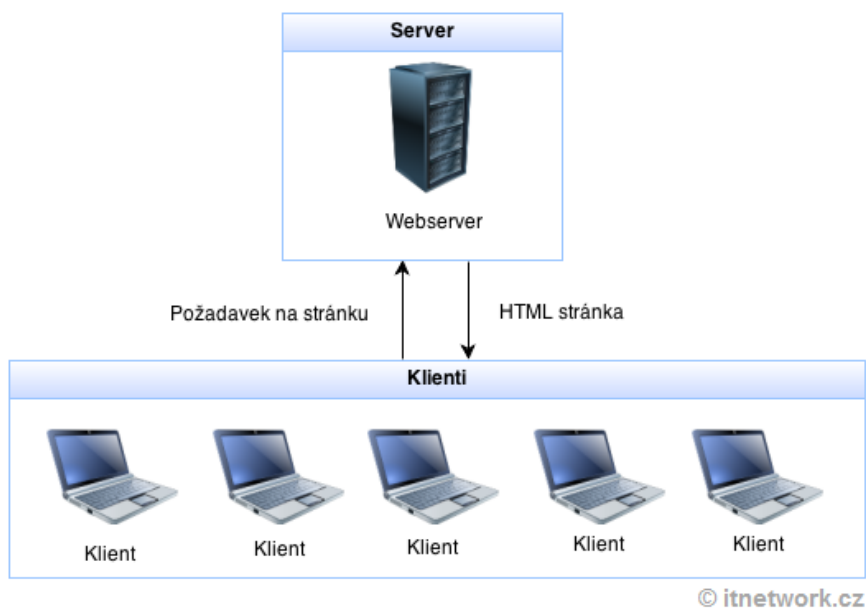
K tomuto tématu maturitní práce jsem se dostal už minulý rok během distanční výuky, když jsme začali se systémem zadávání školních dlouhodobých prací pracovat. V té době vzniklo ze strany pana Ing. Tomáše Kazdy možné zadání budoucí maturitní práce, které měl spočívat v revizi uživatelského této školní aplikace. Téma zůstalo volné a v příštím školním roce jsem se nakonec rozhodl si ho vybrat. Dalším důvodem, proč jsem si ho vybral bylo, že jsem se chtěl naučit s něčím pokročilejším, než je základní HTML/CSS, které běžně používáme pro tvorbu statických stránek na hodinách WEB.

V průběhu času se zadání práce mírně změnilo. Po další konzultaci s vedoucím práce byla k zadání také přidána revize rozhraní další školní aplikace pro správu praxí. Tyto aplikace jsou pro studenty a učitele nepřehledné, a zároveň mají úplně odlišná rozhraní, ne vždy respektující školní designový jazyk. Bude navrženo nové UI/UX rozhraní obou aplikací, a to bude nakonec realizováno. Novinkou je pro mě práce s Frameworkem React, na kterém jsou školní aplikace postavené. Myslím si, že se mi budou znalosti vývoje webových aplikací, v tomto případě konkrétně s využitím frameworku Reactu do budoucna hodit.

1 Specifika webových aplikací

Jako webové aplikace by se daly označit aplikace, které s počítačem komunikují prostřednictvím webového prohlížeče, který funguje jako klient. Většinou jde o komplexní webové stránky, nebo dokonce celé webové programy. Jejich výhodou jsou velmi nízké hardwarové požadavky a nezávislost na konkrétním počítači, kde není nutné instalovat a aktualizovat žádný software. Webové aplikace na rozdíl od běžných webových stránek nejsou statické a mohou zobrazovat odlišný obsah podle určitého uživatele. Mohou například v případě našich školních aplikací nebo různých internetových e-shopů pracovat ukládáním a vyjímáním dat z databází. Další možností využití webových aplikací je tam, kde se neustále mění obsah. Například u zpravodajských webů, kde pak není nutné při každé aktualizaci neustále měnit kód webové stránky. Webové aplikace jsou jedním z pilířů takzvaného Web 2.0. (1) (2) (3)

1.1 Statické webové stránky

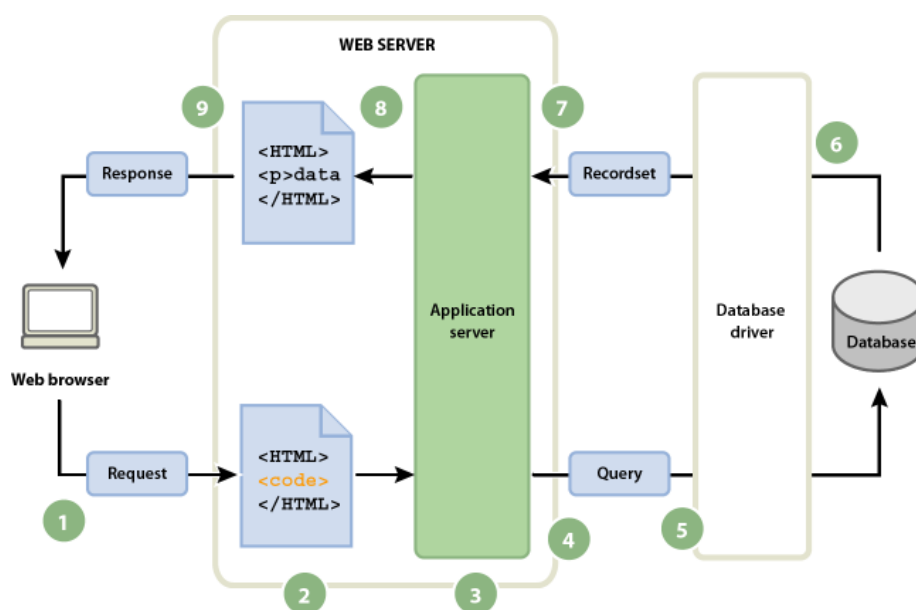


Obrázek 1- Diagram statické webové stránky (3)

Se vznikem internetu nejdříve přišly statické webové stránky ve formě jednoduchého HTML dokumentu. Tyto HTML stránky jsou uloženy na serveru, ke kterému se při zadání adresy prohlížeč připojí. Prohlížeč potom stránku stáhne a následně zobrazí, není už možné serveru nic odeslat zpět, a proto je obsah statický.

Výhodou zde je jednoduchost a především bezpečnost, uživatel se nemůže dostat k datům které na serveru nejsou uložena. V dnešní době se tyto stránky používají hlavně tam, kde je méně obsahu, který se nemusí často měnit a tím pádem je tvorba webové aplikace zbytečná. (3)

1.2 Webové aplikace



Obrázek 2 – Diagram webové aplikace, v tomto případě založené na PHP (2)

S rostoucí funkcí webů vznikly webové aplikace v podstatě jako náhrada klasických desktopových aplikací. V tomto případě to funguje tak, že se prohlížeč zeptá na určitou stránku, která je pomocí skriptů, které se spouští na straně serveru upravena podle požadavků uživatele. Většinou server pracuje s databází, z níž některá data požaduje nebo je do ní ukládá. Existuje mnoho programovacích jazyků, které mohou být pro serverový kód použity. Výhodou je zde už zmínění flexibility, tedy možnost obsah upravit a hned se projeví u všech uživatelů a kompatibility, webové aplikace používají prohlížeč, není nic nutné instalovat a fungují prakticky na všech zařízeních. (1) (2) (3)

Díky potřebám školních aplikací PráCe a PraXe jako je přistupování k datům z databází je samozřejmě nutné použít tuto filozofii.

1.3 Front-end

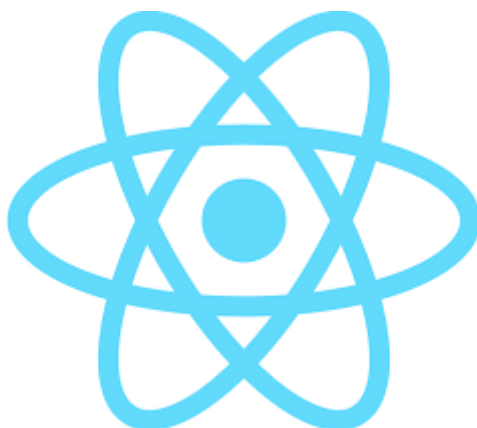
Front end je označení pro „přední vrstvu“ webové stránky, tedy to, co uživatelé vidí a s čím mohou interagovat. Většinou se zde používá jazyků HTML, CSS a JavaScript.

Do front endu pak patří veškerý obsah jako obrázky a grafika. Všechny tyto jazyky fungují na straně klienta, tedy v prohlížeči uživatele. Front-endem rozumíme hlavně uživatelské rozhraní. Často se pro tvorbu uživatelského rozhraní při tvorbě webových aplikací používají různé knihovny jako jsou například React, Angular nebo Vue.js.

1.3.1 React

React je Javascriptová knihovna pro tvorbu uživatelského rozhraní webových aplikací vyvíjená firmou Facebook (nyní už Meta). React aplikace se skládají z jednotlivých „komponent“, které představují jednotlivé třídy nebo funkce v Javascriptu. Pomocí Reactu se dá vytvářet komplexní UI a interaktivní webové aplikace. React používá virtuální DOM, pomocí kterého manipuluje s obsahem HTML dokumentu. Díky tomu v podstatě aplikace běží v prohlížeči místo na serveru. Je to právě knihovna, se kterou budu pracovat při tvorbě školních aplikací. (4) (5)

React používá vlastního rozšíření jazyka JavaScript, který upravuje strukturu komponentů. Umožňuje a zjednodušuje psaní HTML kódu do JavaScriptu. Soubory končí koncovkou `.jsx` místo `.js`. Zároveň umožňuje použití TypeScriptu, což je programovací jazyk od Microsoftu, který je v podstatě nadstavbou pro samotný JS.



Obrázek 3 - Logo React frameworku (6)

1.3.2 Angular

Angular je open-source platforma pro tvorbu webových aplikací vyvíjená týmem Googlu. Je velmi podobná Reactu. Rozdílem je, že používá jazyk TypeScript místo Javascriptu. Jedná se o nástupce AngularJS. (7) (8)

1.4 Back-end

Opakem front end je back end. Za back end je považována „zadní část“, tedy ta část kódu webové aplikace, která není uživateli přímo přístupná a většinou běží na serveru. V případě e-shopu se může jednat například o databáze a systém pro správu zboží. Existuje velké množství možností frameworků a programovacích, které lze při programování back endu využít. (9)

1.4.1 PHP

PHP (zkratka PHP Hypertext Preprocessor) je skriptovací programovací jazyk, používaný hlavně pro programování dynamických webů a webových aplikací na serverové straně webu. Jedná se o jeden z nejstarších a zároveň nejpoužívanějších jazyků pro vývoj na webu. PHP má ale své problémy, mezi některé patří například bezpečnost. Tyto a další problémy řeší některé frameworky PHP, jako například Laravel, které se stávají čím dál populárnější. Některé velké weby používající PHP jsou například Wikipedia nebo Facebook. (3) (10)

1.4.2 Node.js

Node.js je platforma pro tvorbu škálovatelných internetových aplikací a webových serverů. Programy Node.js jsou psané v JavaScriptu. Node.js umožňuje běžení JavaScript kódu mimo prohlížeč přímo na serveru webu, což zvyšuje rychlost a bezpečnost. Patří také mezi nejmodernější platformy a je velmi optimalizováno na asynchronní operace. Mezi velké uživatele Node.js patří například LinkedIn, Microsoft, Netflix, PayPal, Amazon AWS a další. Je možné ho provozovat na Windows i Linux serveru. (11)



Obrázek 4 - Node.js logo (12)

1.4.3 ASP.NET

Jedná se o sadu knihoven (framework) vyvíjený Microsoftem, která je součástí .NET frameworku. Umožňuje tvoření webových aplikací v několika jazycích, v dnešní době primárně C#. Je vhodný i pro velmi velké projekty jako webové portály nebo e-shopy. Je to nástupce staršího frameworku ASP, který je ale velmi odlišný. Aktuální verze je 4.8. ASP.NET není tolik rozšířený z důvodů nutnosti webhostingu, který ho podporuje a také je zde nutné řešit licence. Aplikace v ASP.NET je možné provozovat na i na Linux (Apache) serveru, ale většinou se používá hosting založený na Windows Serveru. (13) (14)

1.4.4 Ruby on Rails

Ruby on Rails je dalším moderním webovým frameworkem. Je vhodný pro vývoj webových aplikací propojených na databázi. Je založený na architektuře MVC (*model-view-controller*) a používá se zde programovacího jazyka Ruby. Ruby on Rails používá mnoho velkých firem jako například Twitch.tv, GitHub, Coinbase nebo Kickstarter. (15)

2 Návrh a vývoj designu stránek

Pod webdesign by se dala zařadit celkově tvorba webových stránek nebo webové aplikace. Měly by splňovat základní kritéria jako přicházení nových návštěvníků a jejich následné vrácení případně provedení konverze, jako je často například zakoupení produktu na e-shopu. Cílem mého redesignu školních aplikací je především upravení rozhraní tak, aby byla pro studenty a uživatele co nejlépe použitelná. Návrh stránek bych mohl rozdělit na dvě části, a to Layout a Typografii. (16)

2.1.1 Layout

Layout je hlavní částí uživatelského rozhraní. S nástupem mobilních zařízení na webu začal být používán responzivní webdesign. Ten je založen na CSS3 a reaguje na velké množství různých možností typů zařízení a velikostí obrazovky, tak aby byl co nejlépe optimalizovaný a aby vypadal a fungoval, jak má na všech zařízeních.

2.1.2 Typografie

Typografie se zabývá především textem na webu. Součástí typografie je nejen správná textace podle požadavků SEO, ale i typy a velikosti písem a například správné řádkování. Protože cílem mé práce je vytvoření aplikace, kde je obsah generovaný nebo načítaný ze školní databáze, nebudu se typografií příliš zabývat.

3 Návrh úprav aplikací PráCe a PraXe

Jedním z cílů méj maturitní práce byl návrh úprav školních aplikací PráCe a PraXe. Aplikace PráCe je určena na správu a hodnocení školních dlouhodobých prací jako jsou ročníkové práce či maturitní práce, a také umožňuje jejich dokumentaci, přidávání, upravování a hodnocení. Tato aplikace, nebo alespoň její nová verze, byla vytvořena panem učitelem Mgr. Michalem Stehlíkem. Aplikace PraXe je druhou školní webovou aplikací, která je určená na evidenci dlouhodobých praxí. Tato aplikace byla navržena a vytvořena v rámci maturitní práce v minulých letech bývalým studentem školy. Mým cílem je sjednotit tyto aplikace tak, aby měli stejné funkční a vizuální prvky.

3.1 Analýza

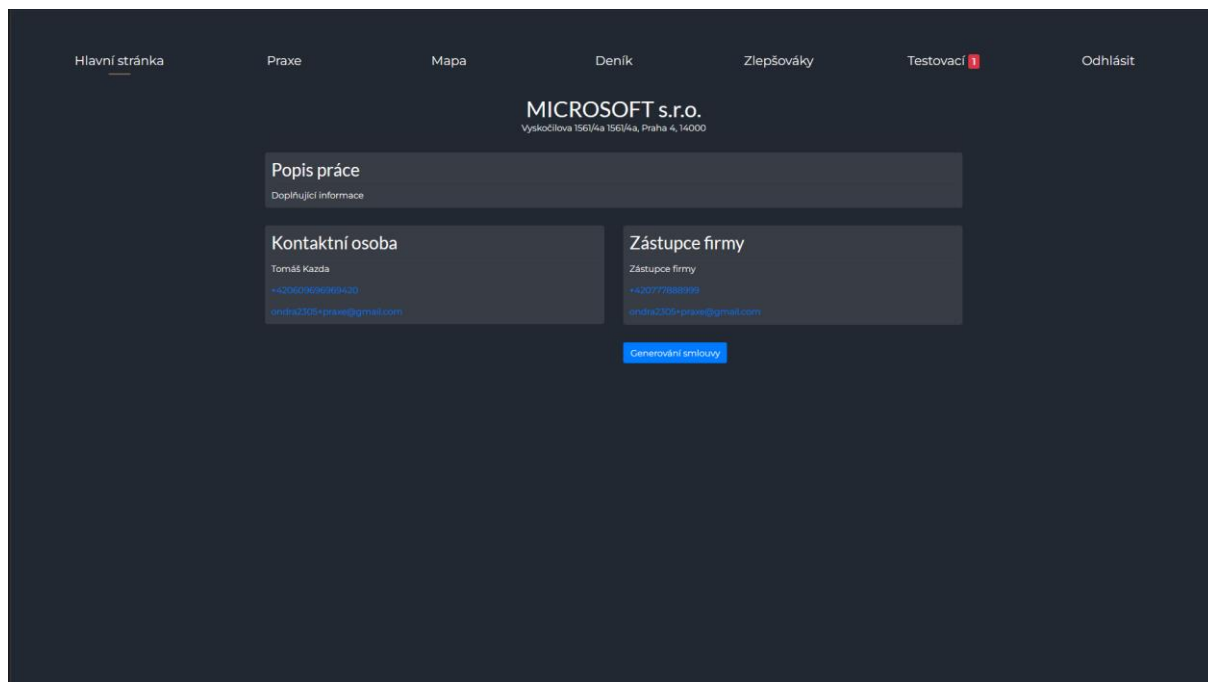
Pro analýzu aktuálního stavu školních aplikací jsem dostal přístupové údaje jak pro testovací učitelský účet, tak pro testovací účet studenta. Zároveň jsem dostal přístup do GitHub repozitáře aplikací. GitHub je služba využívající nástroje pro kontrolu verzí při kódování Git poskytující hosting v rámci takzvaných repozitářů. Komplikací je, že kód těchto aplikací není psán stejným způsobem, a to mi hlavně u aplikace PráCe komplikuje dělání změn. Proto bylo po dohodě s vedoucím práce stanoveno, že hlavním cílem bude právě aplikace PraXe a zaměřím se primárně na redesign a implementace v Reactu bude až posledním bodem.

3.1.1 Aplikace PraXe

Prvním nedostatkem aplikací je na první pohled úplně odlišný design. To byl první bod, který jsem musel zohlednit při návrhu úprav. Rozhodl jsem zde použít školní barvy a jako vzor jsem použil náš školní web. Dalším problémem PraXe je nesmyslnost rozložení prvků a celková nepřehlednost. Příkladem je například navigace v horní části stránky, kde je z nějakého důvodu součástí této navigace odkaz na správu účtu a odhlášení. Stránky také nemají žádné vizuální oddělení obsahu do hlavičky, obsahu a horní části stránky. Všechno doslova jen létá ve vzduchu na tmavém pozadí.

Během testování jsem také narazil na některé zásadní chyby. To je sice spíše věc back endu, ale je možné se některým vyhnout pouze úpravou UI/UX. Jednou z nich třeba je, že pokud vytvoříme hodnocení praxe, není možné ho už smazat, tlačítko na smazání prostě chybí. To vedlo k pracím, které mají popisek pouze „Nelze smazat“. Další chybou,

kteřou jsem během analýzy objevil je, že je možné zadat IČO firmy, která neexistuje a aplikace to umožní. Výsledkem je firma s názvem „Toto IČO jsme nenašli.“.

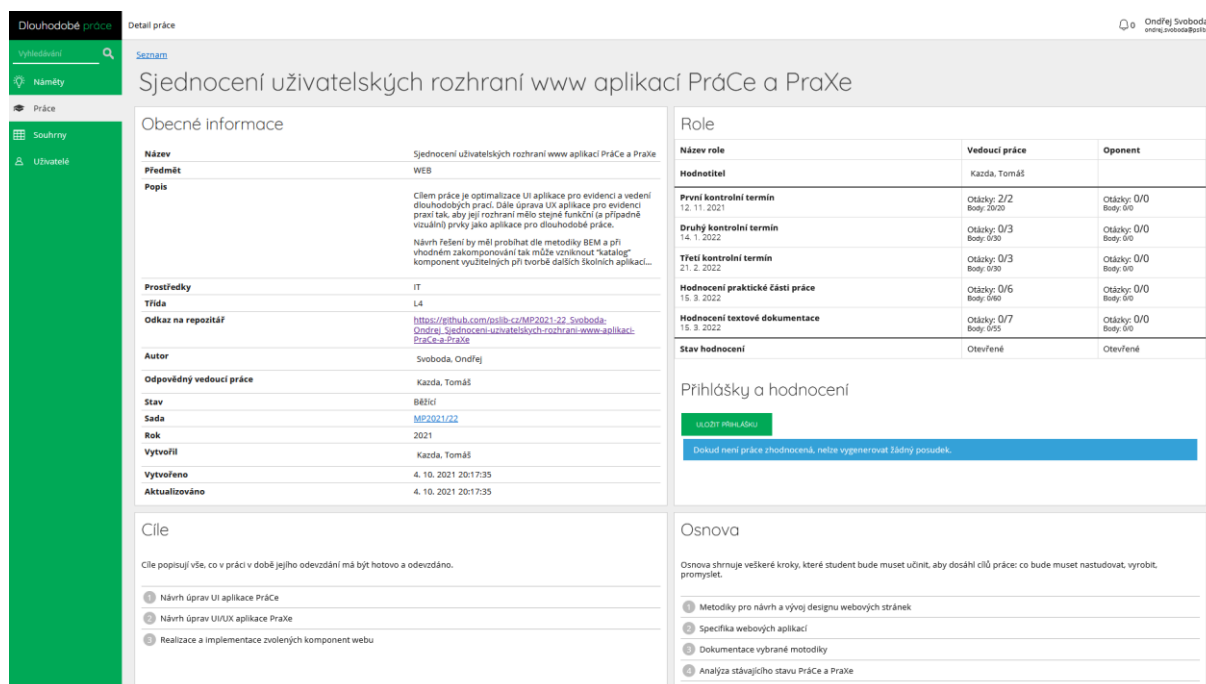


Obrázek 5 - Původní stav aplikace PraXe

Aplikace PraXe se skládá z následujících hlavních stránek:

- **Domů** – domovská stránka, představuje rozcestník
- **Praxe** – seznam všech praxí
- **Deník** – zde si studenti píšou hodiny na praxích
- **Zlepšováky** – určité bonusy, které mohou během praxe získat
- **Mapa** – mapa obsahující místa vykonávání praxí

3.1.2 Aplikace PráCe



Obrázek 6 - Původní stav aplikace PráCe

V aplikaci PráCe se zaměřím na vizuální změny. Mezi ně bych mimo změny barevné palety zařadil úpravu navigace, aby byla horizontální místo vertikální, a hlavně některé prvky zlepšující přehlednost. Příkladem jsou například nestylovované odkazy, které nahradím plnohodnotnými tlačítky. Další věci jsou příliš malé texty a příliš mnoho zbytečných informací v tabulkách na stránce s podrobnostmi o konkrétní práci.

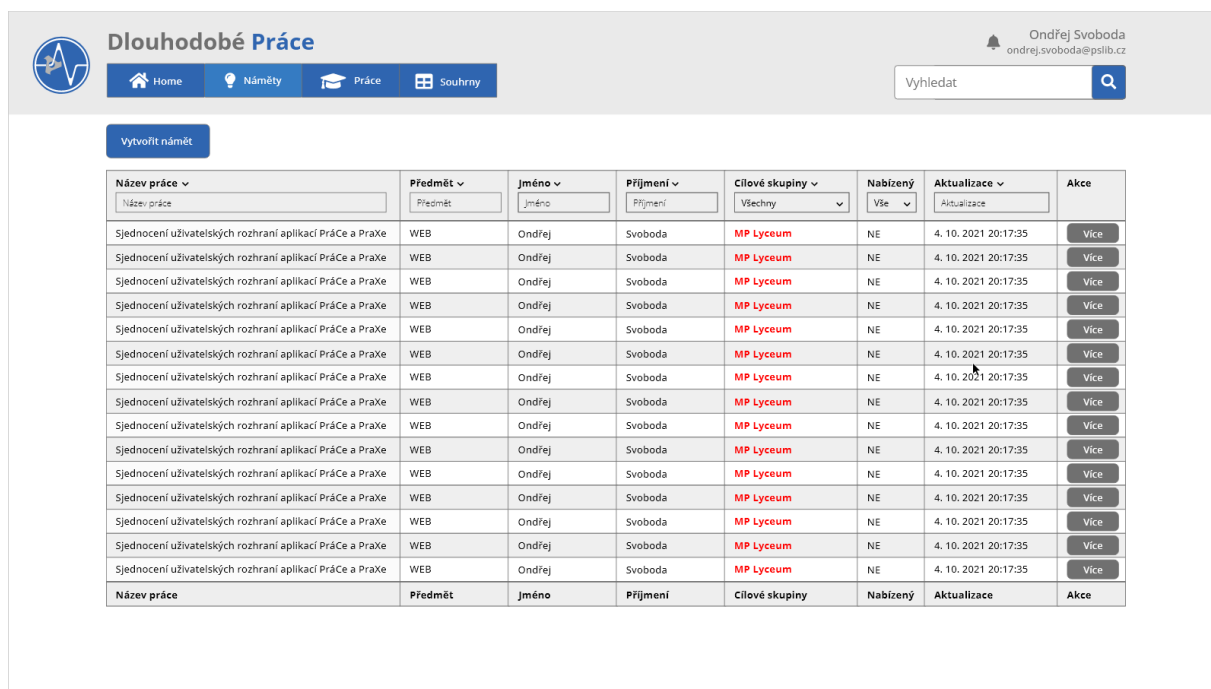
Aplikace PráCe se skládá z následujících hlavních stránek:

- **Domovská stránka**
- **Náměty**
- **Práce**
- **Souhrny** – zde je možné zobrazit stav prací vzhledem ke třídě

3.2 Design

Pro vytvoření sjednoceného designu a UI/UX obou aplikací jsem použil program Adobe XD, se kterým ve škole běžně pracujeme a mám v něm zkušenosti z předchozích prací. V programu Adobe XD jsem využil několika pluginů, hlavně pro vkládání ikon a placeholder textů Lorem Ipsum. Pro lepší přehlednost jsem design obou aplikací

navrhoval do jednoho souboru a jednotlivé aplikace PraXe a PráCe jsem rozdělil do samostatných plátén. V návrhu jsem použil zabudované funkce na spojování jednotlivých prvků do skupin, které lze poté pouze kopírovat a používat na více místech. Tyto skupiny budou představovat moje komponenty později při kódování. Podobně jsem využil i možnosti ukládání barev a velikostí textů. Sice bylo cíle práce aplikace nakódovat a vytvořit jakési demo, ale i tak jsem využil v návrhu takzvaného prototypingu, neboli propojení stránek, aby se návrh stal více interaktivním.



Obrázek 7 - Screenshot z návrhu v aplikaci Adobe XD

3.3 Návrh úprav

V této části se budu věnovat některým částem konkrétních změn, které jsem při vytváření návrhu implementoval.

3.3.1 Fonty

PraXe používají font Montserrat, a ještě k tomu k nepřiměřeně velkým velikostem. V obou případech se sice jedná o bezpatkové fonty, ale tento neodpovídá školnímu designovému manuálu, který používá font Open Sans, podobně jako v aplikaci PráCe. Tato aplikace naopak měla zase problémy s příliš malým písmem. V mojí upravené verzi jsem se proto rozhodl tento font použít s přiměřenou velikostí textu (16px).

Regular 400

Almost before we knew it, we had left the ground.

Obrázek 8 – Font Montserrat

Regular 400

Almost before we knew it, we had left the ground.

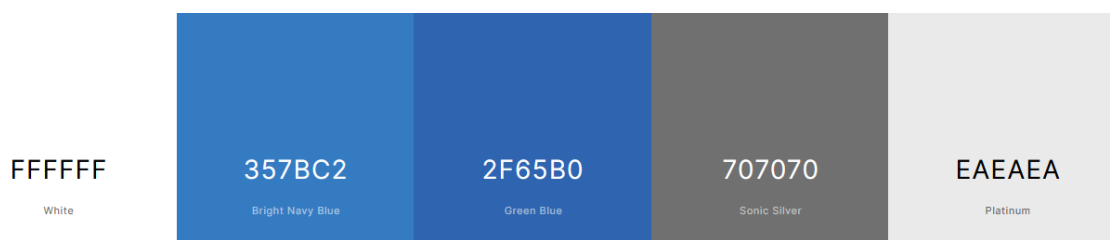
Obrázek 9 - Font Open Sans

3.3.2 Navigace

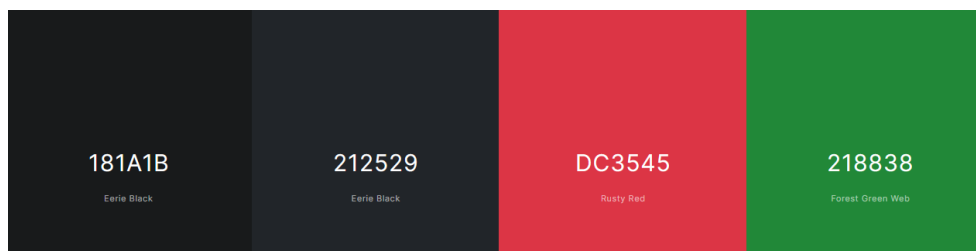
Jedním z prvků, který bylo nutné upravit byla navigace na stránkách. V původním stavu aplikace PráCe je využito postranní vertikální menu na levé straně obrazovky, na mobilní verzi se přesune do dolní části obrazovky. Naopak v aplikaci PraXe je použito horizontální navigace a na verzi pro mobilní zařízení se přepne do takzvaného „hamburger“ menu neboli rozbalovacího menu pomocí stisknutí tlačítka. Mně se sama o sobě nelíbila ani jedna z těchto variant, tak jsem se rozhodl využít upravenou verzi. Na desktopové verzi využívám klasické horizontální menu, podobné tomu v původní aplikaci PraXe a v mobilní verzi jsem se rozhodl použít menu v dolní části stránky, podobně jako v PráCe.

3.3.3 Barevná paleta

Barevnou paletu návrhu jsem upravit tak, aby reflektovala školní barvy, a celkově se podobala webu školy. Líbil se mi nápad použití tmavé verze webu z aplikace PraXe. Do návrhu jsem se proto rozhodl zakomponovat i tmavý režim, který je na internetu v poslední době velmi oblíbený, hlavně na mobilních zařízeních. Aplikace se bude v ideálním případě přepínat mezi světlým a tmavým režimem podle nastavení prohlížeče nebo systému uživatele. Světlá barevná paleta podobná školnímu webu zůstává jako výchozí nastavení. Jako hlavní barvy jsem použil školní modrou barvy *Bright Navy Blue* #357BC2 a *Green Blue* #2F65B0. Doplnkem pro tyto barvy jsou ve světlém režimu bílá (pozadí stránky) a další odstíny šedé barvy. V tmavém režimu jsem tyto barvy nahradil černou a tmavě šedými odstíny barev. Tuto barevnou paletu doplňují barvy jako zelená, červená a žlutá pro zvýrazněné prvky jako jsou tlačítka, zde jsou použil barvy buď už použité na školním webu, nebo jsem je doplnil pomocí online generátoru palet *Coolors.co*.



Obrázek 10 - Hlavní barevná paleta



Obrázek 11 - Přídavné barvy

3.3.4 Design stránek

Při úpravách designu stránek jsem se opět inspiroval existujícími aplikacemi a školním webem. Při designování stránek jsem se držel moderních konvencí jako je responzibilita webu čili aby fungoval stejně dobře na všech typech zařízení od malých obrazovek jako jsou telefony po velké jako jsou notebooky a desktopy. Design jsem vytvořil takzvaně Mobile First – s upřednostněním použití na mobilních zařízeních. Hodně času jsem ale strávil také nad desktopovou verzí pro počítače, protože vím, že na aplikace se dost často přistupuje i z desktopu, nejen ze strany učitelů. Verzi pro tablety jsem se příliš nevěnoval, takže jde jen o opravenou verzi pro notebooky. Důvodem je, že tablety postupně vymírají a mimo iPad je v dnešní době prakticky nikdo nepoužívá, takže by to bylo zbytečné.



Obrázek 12 - Zvolená desktopová navigace



Obrázek 13 - Zvolená mobilní aplikace

4 Návrh komponent webu

4.1 Identifikace komponent

Při tvorbě React aplikací je nutné aplikaci rozdělit do jednotlivých bloků neboli komponent. Komponentou může být například tlačítko, které lze potom v aplikaci na různých místech používat znovu. Jedná se o úplně nezávislé části kódu, které lze znovu použít v jiné části aplikace nebo dokonce v úplně odlišném projektu. Fungují jako funkce v JavaScriptu. V Reactu je možné vytvořit komponenty dvěma způsoby, pomocí třídy nebo pomocí JavaScript funkcí. Tyto komponenty se potom zapisují podle úrovně podrobnosti. (17) (18)

```
class HelloWorld extends React.Component {  
    render() { // Definice a vykreslení komponenty  
        return React.createElement('h1', null,  
            'Hello World!'); // Vrátí text „Hello World!“ v <h1></h1>  
    }  
}
```

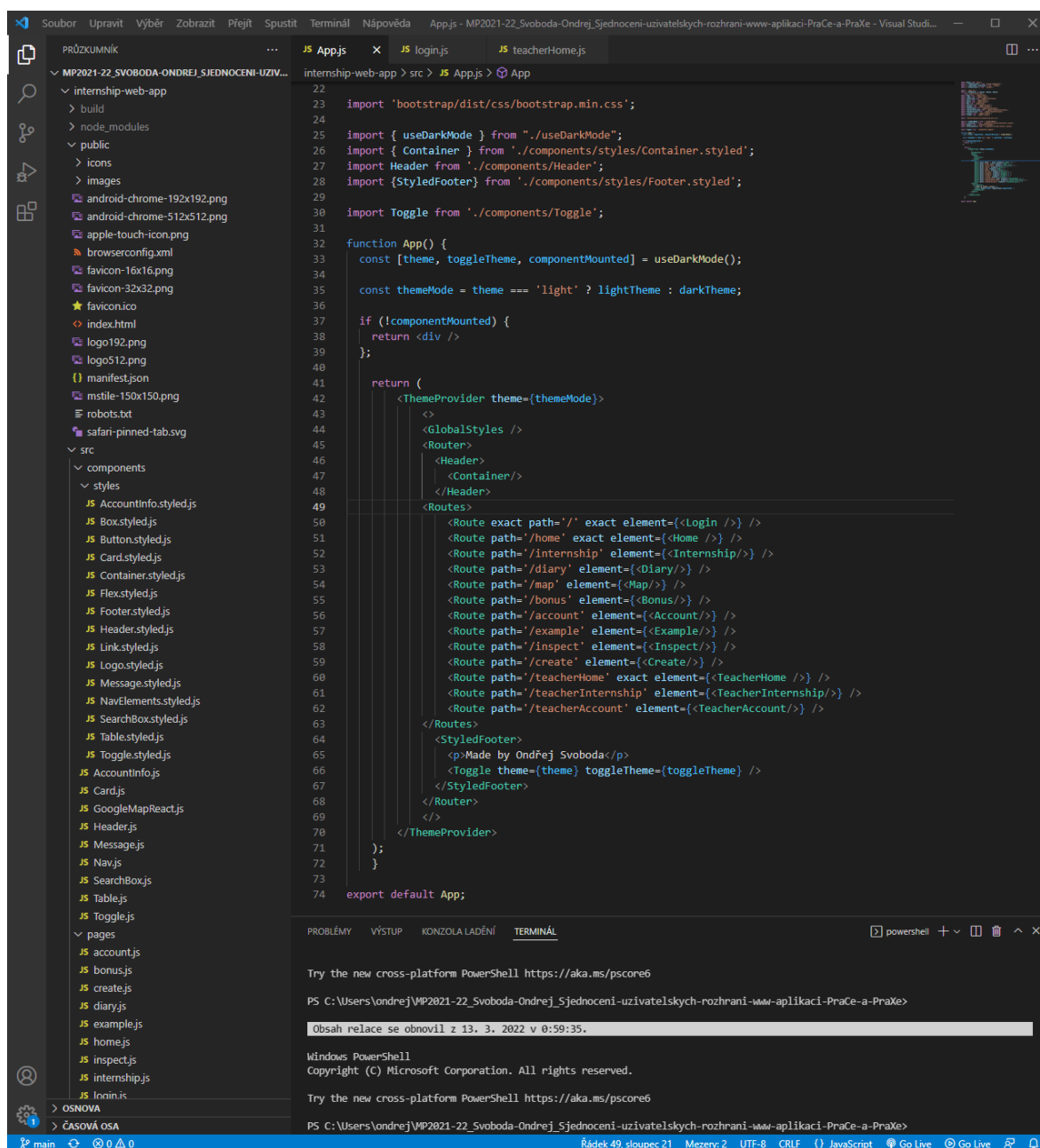
Ukázka zdrojového kódu ukazuje jednoduchou komponentu vytvořenou pomocí třídy, která pouze vrátí text „Hello World!“.

4.2 Příprava

Pro kódování jsem se rozhodl použít editor Visual Studio Code. Používám ho, protože s ním mám zkušenosti z hodin předmětu WEB a díky jeho různým funkcím a rozšířením, která mi proces kódování podstatně zjednodušují. Mezi výhodami VS Code je například integrovaná podpora Gitu, díky níž můžu jednoduše spolupracovat se vzdáleným repozitářem PSLIB na GitHubu. Další výhodou tady rozšíření *ES7+ React/Redux/React-Native snippets*, díky kterému můžeme používat různé zkratky pro napsání částí kódu, které se pořád opakují.

4.3 Kódování

Než jsem se vůbec do kódování pustil, rozhodl jsem se projít na internetu videokurzem na platformě YouTube, zaměřující se na framework React. Prvním krokem samotné tvorby aplikace byl samotné nainstalování frameworku React. Pro vytvoření aplikace jsem použil *create-react-app*, který umožňuje vytvoření React aplikace prakticky pomocí jednoho příkazu. Jedinou podmínkou je nainstalovaný Node.js server. Právě proto jsem se rozhodl pro vývoj použít Node.js, který jsem nainstaloval pomocí NVM. Po instalaci je už možné pokračovat na tvoření aplikace nebo instalovat další závislosti.



Obrázek 14 - Kódování ve VS Code

4.3.1 Závislosti

Závislosti představují většinou další Javascriptové knihovny upravené pro použití s Reactem, které jsem během kódování aplikací PráCe a PraXe využil. Závislosti se do prostředí Node.js instalují pomocí jednoho ze správců balíčků jako je npm nebo Yarn. Já využil npm, kde je příkaz nutný pro doinstalování nové závislosti *npm install [název závislosti]*. Tyto závislosti jsou v naprosté většině případů dostupné pod otevřenými open source licencemi jako je MIT nebo GPL, takže je možné je bez problému použít i pro komerční použití.

4.3.1.1 Create-react-app

Zmíněno výše, *create-react-app* umožňuje jednoduché vytvoření React aplikace a automaticky nainstaluje všechny nutné závislosti pro vývojové prostředí. Pomocí pouze jednoho příkazu *npx create-react-app my-app* vytvoříme jednoduše aplikaci, kde rovnou můžeme začít s kódováním. Souborová struktura aplikace bude vytvořená pro nás, stejně tak jako skripty pro testování nebo sestavení aplikace.

4.3.1.2 FortAwesome (FontAwesome)

Při tvorbě jsem implementoval React knihovnu Font Awesome, která umožňuje použití stejnojmenných ikon, kterých jsem využil při vytváření designu. React implementace Font Awesome umožňuje použití maker, díky kterým je možné se vyhnout přidáváním jednotlivých ikon manuálně a statickému importování všech (i nepoužitých) ikon. Je nutné ale doinstalovat další závislost, a to *babel-plugin-macros*. Tato makra mi ale způsobila značné množství problémů, dokonce jsem musel přeinstalovat všechny závislosti, jen abych tento problém vyřešil. Pro jednoduchost bych se dynamickému vkládání ikon v budoucnu radši vyvaroval.

4.3.1.3 React-gh-pages

React-gh-pages je nástroj obsahující skripty pro nasazení vaší aplikace na GitHub Pages hosting, který je na GitHubu dostupný pro všechny repozitáře. Já jsem se GitHub Pages rozhodl použít pro hostování demo verze mých aplikací. Při pokusu o použití *react-gh-pages* jsem ale narazil na řadu problému, které mi trvalo několik hodin vyřešit. Jedním z nich byl problém s přihlášením, protože není možné jednoduše použít heslo od GitHub účtu z důvodu bezpečnosti. Toto jsem relativně rychle vyřešil vytvořením jednorázového

tokenu, který se v tomto případě dá použít místo hesla. Větším problémem ale bylo to, že jsem potřeboval nasadit PráCe i PraXe na stejné GitHub pages a to rozbilo veškeré obsahy. Nakonec se mi povedlo na internetu najít řešení zahrnující přidání jakéhosi rozcestníku a upravení některých cest k souborům. Nasazení demo prostředí aplikací by bylo určitě jednodušší na vlastní doméně.

4.3.1.4 React Bootstrap

React Bootstrap představuje knihovnu Bootstrap, která umožňuje rychlé stylování z předpřipravených komponent a stylů, upravenou pro použití s Reactem. Nevyžaduje žádné další závislosti, jako je například JQuery. Použil jsem ho pouze v první části během testování, když jsem zkoušel funkčnost a vytvářel layout některých stránek.

4.3.1.5 React Leaflet

React Leaflet je implementace populární Javascript knihovny pro interaktivní mapy. Já jí používám v aplikaci PraXe na stránce Mapa pro zobrazení lokace míst, kde se vykonávají praxe. Umožňuje jednoduché přidání upravitelných ikon s popisky na zvýrazněná místa, jako jsou v našem případě právě místa konání praxí.

4.3.1.6 React Router

React Router je knihovna pro routování (směrování) na klientské nebo serverové straně. Umožňuje fungování odkazů a přesměrovávání uživatele na jiné stránky, upravení URL nebo použití různých URL parametrů. Já jsem jí konkrétně použil v navigaci, protože jsem potřeboval využít několik různých stránek mezi kterými se uživatel pohybuje pomocí různých tlačítek a odkazů. React Router používám v hlavním souboru *Reactu App.jsx*.

```
<Routes>
  <Route exact path="/" exact element={<Login />} />
    <Route path="/home" exact element={<Home />} />
    <Route path="/internship" element={<Internship/>} />
    <Route path="/diary" element={<Diary/>} />
    <Route path="/map" element={<Map/>} />
    <Route path="/bonus" element={<Bonus/>} />
    <Route path="/account" element={<Account/>} />
    <Route path="/example" element={<Example/>} />
    <Route path="/inspect" element={<Inspect/>} />
    <Route path="/create" element={<Create/>} />
    <Route
      path="/teacherHome"
      exact
      element={<TeacherHome />} />
    <Route
      path="/teacherInternship"
      element={<TeacherInternship/>} />
    <Route
      path="/teacherAccount"
      element={<TeacherAccount/>} />
</Routes>
```

V ukázce kódu je vidět příklad použití React Routeru v jedné z aplikací.

4.4 Stylování komponent

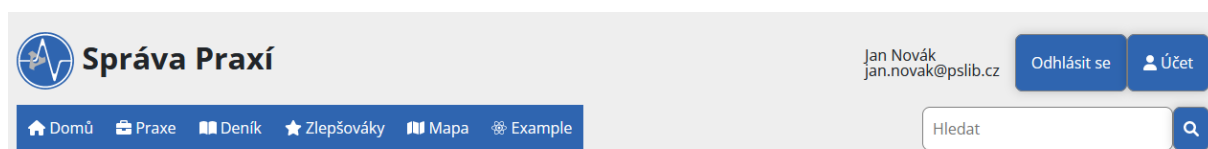
Existuje několik možností stylování komponent v Reactu. Některé používají pouze Javascript, jiné způsoby kombinují Javascript a CSS. Je možné použít jakýkoliv ze způsobů podle potřeb vývojáře a konkrétní webové aplikace. Já jsem z několika důvodů zvolil techniku Styled Components, kterou níže popisuji. (19) (20)

4.4.1 Styled Components

Styled Components je knihovna, která kombinuje JavaScript a CSS. Jedná se o „dependency“ – závislost, kterou je nutné doinstalovat. Umožňuje přímé psaní CSS kódu do komponent. To může být pro někoho výhodou, ale výsledný kód to komplikuje. Právě díky možnosti přímého psaní CSS kódu jsem se rozhodl tento framework použít při vytváření méj React aplikace. Každá komponenta představuje vlastní soubor a pomocí *props* (předávaných hodnot z jiné části kódu) je možné konkrétní komponentu poté

upravovat. Naopak se zde vůbec nepoužívají HTML třídy, Styled Components vytváří vlastní „virtuální“ třídy při vygenerování HTML kódu stránky. Následující kód ukazuje příklad použití Styled Components při tvorbě komponenty *Header* (hlavička stránky) (19) (21)

```
export default function Header() {
  return (
    <StyledHeader>
      <Container>
        <NavBlock>
          <NavContainer>
            <h1><Logo src="..." alt="PSLIB logo"/>
            Správa Praxí</h1>
            <Nav />
          </NavContainer>
          <AccountInfo />
        </NavBlock>
      </Container>
    </StyledHeader>
  )
}
```



Obrázek 15 - Výsledek stylování komponenty *Header* pomocí *Styled Components*

Styled Components implementuje využití takzvaných *props*, což jsou proměnné, pomocí kterých lze posílat určité parametry do komponenty. V praxi to potom umožňuje například vytvořit komponentu *Tlacitko*, které bude například modré a pomocí *props* změnit tuto barvu na například červenou.

```
export const StyledButton = styled.button`
  background-color: blue;

  &:hover {
    background-color: lightblue;
  }
  ${props => props.Red && css`
    background-color: red;
  `}
`
```

4.4.2 Bootstrap

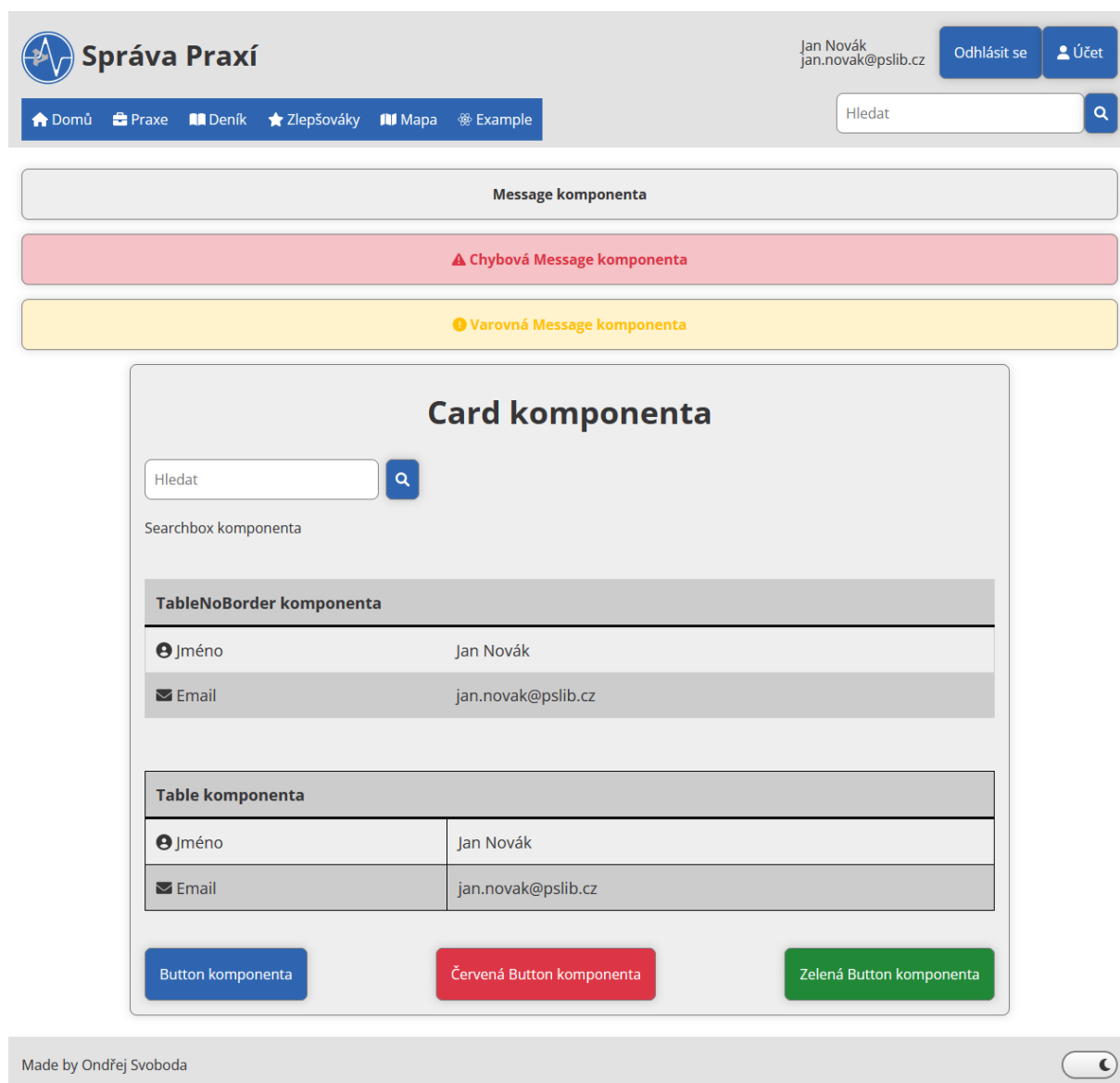
Bootstrap je knihovna, která umožňuje rychlé stylování stránek, protože jsou již předpřipravené, a my můžeme používat pouze generické styly podle dokumentace. Je ideální ho použít tam, kde řešíme část aplikace, kam se uživatel běžně nedostane, anebo pokud potřebujeme rychle vytvořit prototyp a není čas vytvořit vlastní design. Pro implementaci v Reactu je možné použít některou z Bootstrap knihoven, jako například *react-bootstrap*. Jsou zde předpřipravené komponenty jako například tabulky nebo formuláře, které lze jednoduše implementovat do naší aplikace. Na problémy můžeme narazit, pokud chceme pro nějaké komponenty použít vlastní styly, potom je vhodné se Bootstrapu vyhnout. Já jsem ho použil hlavně když jsem předběžně zkoušel rozložení prvků na stránce, ale potom jsem ho ve finální verzi odstranil, protože jsem potřeboval vlastní styly. (19) (20)

4.4.3 CSS Modules

Další možností je CSS Modules. V běžném statickém webu používáme HTML, které pak stylujeme pomocí CSS. V CSS Modules se HTML kód zapisuje v Javascriptu a CSS způsobem, na který jsme běžně zvyklí. Tyto CSS moduly se poté do Javascriptu importují a používá se tu klasických HTML selektorů pro stylování. Hlavním cílem této knihovny je, že se snaží vyhnout konfliktům v názvech selektorů, které by mohly vzniknout při větších projektech. Na rozdíl od běžného chování CSS se zde názvy selektorů vztahují pouze ke konkrétní komponentě. (20) (22) (23)

5 Katalog komponentů

Finálním výstupem práce měl být jakýsi katalog komponentů, z kterého bude možné jednotlivé komponenty vzít a implementovat je do budoucích školních aplikací postavených primárně na frameworku React. Tento katalog komponentů jsem nejdříve vytvořil jako samostatné plátno při návrhu designu aplikací v aplikaci Adobe XD. Při vytváření aplikace PraXe jsem zde vytvořil testovací stránku *Example*, na ukázkou jednotlivých komponentů při prezentaci demo aplikace.



Obrázek 16 - Ukázka katalogu komponentů u aplikace PraXe

5.1.1 Card

Hlavní komponentou je komponenta karty, která v podstatě představuje pozadí a ohraničení obsahové části jakékoliv stránky aplikací. Používám jí prakticky na všech stránkách a existují zde dvě varianty, klasická komponenta *Card* a její širší verze *Large*, kterou používám na stránkách, kde je tabulka se zobrazovanými daty (komponenta *Table*).

5.1.2 Table

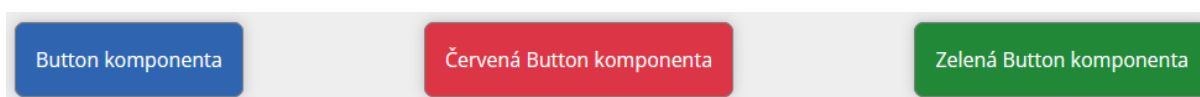
Komponenta tabulky je relativně přímočará. Záhloví a západní jsou stejně jako každý druhý řádek barevně oddělené, pro lepší orientaci při zobrazení velkého množství dat. Jedinou věcí, která se mění je záhlaví tabulky, kde může a nemusí být filtrování v jednotlivých sloupcích.

Firma	Adresa	Typ praxe	Student	Akce
Microsoft s.r.o.	Vyskočilova 1561/4a, 14000 Praha 4	Dlouhodobé praxe 4. ročníků	Ondřej Svoboda	<div> <div>Aktivovat</div> <div>Smazat</div> </div>
Další firma	Další adresa	Dlouhodobé praxe 4. ročníků	Další student	<div> <div>Přidat se</div> </div>
Další firma	Další adresa	Dlouhodobé praxe 4. ročníků	Další student	<div> <div>Přidat se</div> </div>
Další firma	Další adresa	Dlouhodobé praxe 4. ročníků	Další student	<div> <div>Přidat se</div> </div>
Další firma	Další adresa	Dlouhodobé praxe 4. ročníků	Další student	<div> <div>Přidat se</div> </div>
Firma	Adresa	Typ praxe	Student	Akce

Obrázek 17 - Příklad komponenty tabulky *Table*

5.1.3 Button

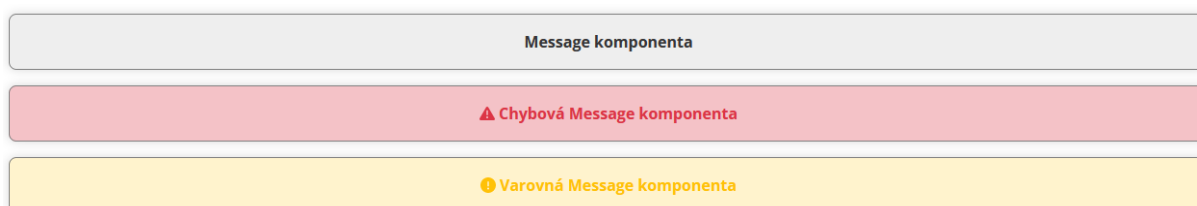
Komponentu *Button* neboli tlačítka používám hodně často, všude tam kde je potřeba navigovat na jinou stránku aplikace nebo otevřít menu. Vytvořil jsem čtyři varianty této komponenty a jediným rozdílem je barva. Tlačítka mohou být buď v modré a šedé barvě (primární a sekundární tlačítka), nebo pokud se jedná o podstatné akce s červeným nebo zeleným zvýrazněním.



Obrázek 18 - Ukázka tlačítkových komponent

5.1.4 Message

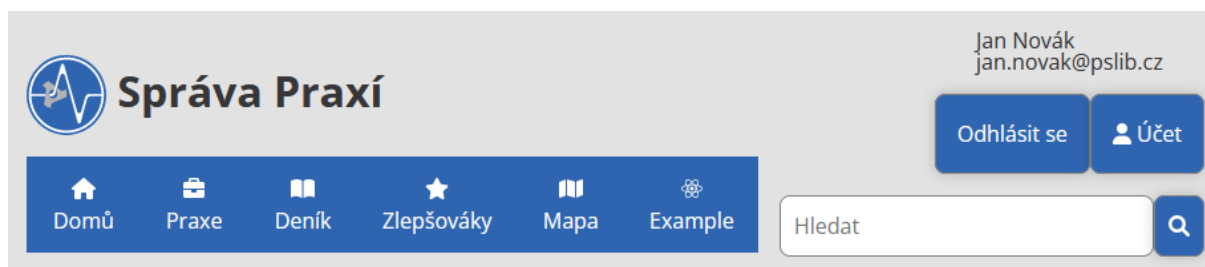
Další komponentou je *Message*, která představuje okno hlášky, které se zobrazí v určitých případech například při zadání špatného hesla nebo nenalezené stránky. Tady jsem vytvořil 3 varianty lišící se barvou a ikonami – šedou pro informační zprávy, žlutou pro varování a červenou pro chyby. Každou hlášku doprovází odpovídající ikona a krátký text.



Obrázek 19 - Ukázka Message komponenty

5.1.5 Header

Komponenta Menu představuje hlavní navigaci v horní části každé stránky webové aplikace. V případě aplikace PraXe se zde nacházejí stránky Domů, Praxe, Deník, Zlepšováky a Mapa. Po najetí na jedno z tlačítek v menu se zvýrazní jeho pozadí. Společně s komponentou *AccountInfo*, kde je možné prokliknout na profil uživatele nebo se odhlásit je součástí komponenty *Header*, která představuje celou hlavičku stránky.



Obrázek 20 - Ukázka komponenty Header a její součásti Menu a AccountInfo

Závěr

Nad prací jsem strávil několik desítek hodin. U práce se vyskytly výše zmíněné problémy se změnou zadání. Ty ale na práci neměly výrazný vliv, protože přišly před tím, než jsem začal pracovat na praktické části. Ze začátku jsem měl pochybnosti ohledně vybrání tohoto tématu maturitní práce, ale nakonec tyto moje obavy byly neopodstatněné. Myslím, že největším přínosem pro mě bylo získání zkušeností s frameworkem React, které jsem získal v podstatě z internetu a samouky, protože k tomuto tématu jsme se ve škole vůbec nedostali. Díky tomu jsem při vytváření aplikací někde nemusel zvolit optimální techniku, ale snažil jsem se držet dokumentace a běžných konvencí.

Myslím si, že se mi tyto znalosti budou hodit v dalším studiu na vysoké škole případně později v praxi nebo zaměstnání. Práce se mě i bavila a s frameworkem React jako takovým se mi pracuje dobře, takže ho při další práci na webové aplikaci určitě zvážím. Pokud by na práci zbylo více času daly by se vzniklé školní aplikace implementovat do produkčního prostředí, to ale nebylo pro mě hlavním cílem práce. Z implementace mého designu a propojení na školní databáze by mohla teoreticky vzniknout další ročníková práce nebo školní projekt. Doufám, že bude moje práce použita a rád bych svůj design školních aplikací PráCe a PraXe viděl nasazený a použitý v co nejbližší době.

Všechny hlavní cíle práce byly splněny a celkově jsem s výsledkem spokojený, ale mám nějaké výhrady k některým prvkům designu, co jsem vytvořil. Vzhledem k tomu, že jsem se nikdy nepovažoval za grafického designéra a spíše jsem se věnoval kódu, tak jsem s výsledkem spokojený. Snažil jsem se hlavně vyhnout chybám, které vznikly při vytváření původního uživatelského rozhraní školních aplikací, kde nebyl brán příliš ohled na funkčnost ze strany uživatele.

Seznam zkratk a odborných výrazů

HTML

Značkovací jazyk používaný pro tvorbu webových stránek propojených hypertextovými odkazy

CSS

Cascading Style Sheets – Kaskádové styly – kódovací jazyk, který se používá na webu pro stylování HTML elementů na webových stránkách

JavaScript

Skriptovací programovací jazyk, často používaný při tvorbě webových stránek, kde ovládá interaktivní prvky. Spouští se často na straně klienta, ale může se využívat i na straně serveru

SEO

Search Engine Optimization – optimalizace obsahu webu pro vyhledávače jako je Google nebo Seznam, cílem je dostat se co nejvýše v žebříčku vyhledávání

UI/UX

Zkratka UI označuje termín *User Interface* neboli uživatelské rozhraní (vzhledová část) a UX představuje spíše základy označuje *User Experience* neboli zážitek uživatele při používání

Layout

V případě webových stránek označuje formát stránky, celkové rozvržení prvků a jejich velikost, při kódování se potom používá kaskádových stylů CSS

Framework

Česky – Aplikační rámec – soubor programů, API nebo knihoven, které upravují postupy vývoje nebo při něm pomáhají

GitHub

Služba využívající nástroje pro kontrolu verzí při kódování Git poskytující hosting v rámci takzvaných repositářů

GitHub Pages

Hosting webového projektu přímo z repozitáře na stránce GitHub, všechny změny v repozitáři se automaticky publikují a aktualizují na GitHub Pages

Lorem Ipsum

Nesmyslný výplňový text používaný mimo jiné při navrhování grafického designu

Dependency

Česky – závislost – část kódu třetí strany, často knihovna, na které jiný kód závisí a nemůže bez ní fungovat

Seznam obrázků

Obrázek 1- Diagram statické webové stránky (3)	2
Obrázek 2 – Diagram webové aplikace, v tomto případě založené na PHP (2).....	3
Obrázek 3 - Logo React frameworku (6)	4
Obrázek 4 - Node.js logo (12)	5
Obrázek 5 - Původní stav aplikace PraXe.....	9
Obrázek 6 - Původní stav aplikace PráCe.....	10
Obrázek 7 - Screenshot z návrhu v aplikaci Adobe XD	11
Obrázek 8 – Font Montserrat.....	12
Obrázek 9 - Font Open Sans	12
Obrázek 10 - Hlavní barevná paleta.....	13
Obrázek 11 - Přídavné barvy	13
Obrázek 12 - Zvolená desktopová navigace.....	13
Obrázek 13 - Zvolená mobilní aplikace.....	13
Obrázek 14 - Výsledek stylování komponenty Header pomocí Styled Components	19
Obrázek 15 - Ukázka katalogu komponentů u aplikace PraXe	21
Obrázek 16 - Příklad komponenty tabulky Table	22
Obrázek 17 - Ukázka tlačítkových komponent	23
Obrázek 18 - Ukázka Message komponenty	23
Obrázek 19 - Ukázka komponenty Header a její součásti Menu a AccountInfo	23

Použité zdroje

1. **Wikimedia Foundation.** Webová aplikace - Wikipedie. *Wikipedie*. [Online] 16. 6. 2019. [Citace: 7. 3. 2022.]
https://cs.wikipedia.org/wiki/Webov%C3%A1_aplikace.
2. **Adobe.** Co jsou to webové aplikace a dynamické webové stránky? *Adobe*. [Online] 19. 5. 2021. [Citace: 7. 3. 2021.]
<https://helpx.adobe.com/cz/dreamweaver/using/web-applications.html>.
3. **Čápka, David.** Lekce 1 - Úvod do PHP a webových aplikací. *itnetwork.cz*. [Online] 12. 2. 2021. [Citace: 7. 3. 2022.] <https://www.itnetwork.cz/php/zaklady/php-tutorial-uvod-do-webovych-aplikaci>.
4. **VOXCAFE s.r.o.** Lekce 1. - React - úvod. *MIND BLOG*. [Online] 02. 10. 2021. [Citace: 13. 1. 2022.] <https://www.voxcafe.cz/mindblog/clanky/programovani/lekce1.html>.
5. **Máca, Jindřich.** Lekce 1 - Úvod do React. *itnetwork.cz*. [Online] [Citace: 13. 1. 2022.]
<https://www.itnetwork.cz/javascript/react/zaklady/uvod-do-react/>.
6. **Facebook.** File:React-icon.svg. *Wikimedia Commons*. [Online] 25. 1. 2022. [Citace: 13. 3. 2022.] <https://commons.wikimedia.org/wiki/File:React-icon.svg>.
7. **Máca, Jindřich.** Lekce 1 - Úvod do Angular frameworku . *itnetwork.cz*. [Online] 19. 2. 2021. [Citace: 7. 3. 2022.]
<https://www.itnetwork.cz/javascript/angular/zaklady/uvod-do-angular-frameworku>.
8. **Wikimedia Foundation.** Angular (web framework) - Wikipedia. *Wikipedia*. [Online] 15. 2. 2022. [Citace: 7. 3. 2022.]
[https://en.wikipedia.org/wiki/Angular_\(web_framework\)](https://en.wikipedia.org/wiki/Angular_(web_framework)).
9. **MDN contributors.** Introduction to the server side - Learn web development. *MDN Web Docs*. [Online] 18. 2. 2022. [Citace: 7. 3. 2022.]
https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Introduction.
10. **Wikimedia Foundation.** PHP - Wikipedie. *Wikipedie*. [Online] 16. 2. 2022. [Citace: 7. 3. 2022.] <https://cs.wikipedia.org/wiki/PHP>.

11. —. Node.js - Wikipedia. *Wikipedia*. [Online] 4. 3. 2022. [Citace: 7. 3. 2022.]
<https://en.wikipedia.org/wiki/Node.js>.
12. **authors, node.js**. File:Node.js logo.svg. *Wikimedia Commons*. [Online] 1. 4. 2017.
[Citace: 13. 3. 2022.] https://commons.wikimedia.org/wiki/File:Node.js_logo.svg.
13. **Wikimedia Foundation**. ASP.NET - Wikipedie. *Wikipedie*. [Online] 4. 8. 2021.
[Citace: 7. 3. 2022.] <https://cs.wikipedia.org/wiki/ASP.NET>.
14. **Čápka, David**. Lekce 1 - Úvod do ASP.NET . *itnetwork.cz*. [Online] 28. 11. 2020.
[Citace: 7. 3. 2022.] <https://www.itnetwork.cz/csharp/asp-net-core/zaklady/tutorial-uvod-do-asp-dot-net>.
15. **Wikimedia Foundation**. Ruby on Rails - Wikipedie. *Wikipedie*. [Online] 27. 5. 2021.
[Citace: 7. 3. 2022.] https://cs.wikipedia.org/wiki/Ruby_on_Rails.
16. —. Webdesign - Wikipedie. *Wikipedie*. [Online] 6. 9. 2021. [Citace: 7. 3. 2022.]
<https://cs.wikipedia.org/wiki/Webdesign>.
17. **W3Schools** . React Components. *W3Schools Online Web Tutorials*. [Online] [Citace: 7. 3. 2022.] https://www.w3schools.com/react/react_components.asp.
18. **VOXCAFE s.r.o.** Lekce 6. - React - CSS. *Blog / VOXCAFE*. [Online] 17. 3. 2022. [Citace: 7. 3. 2022.]
<https://www.voxcafe.cz/mindblog/clanky/programovani/lekce6.html>.
19. **ReactGirls**. React: Stylování komponent. | by ReactGirls | Medium. *Medium*. [Online] 30. 9. 2021. [Citace: 7. 3. 2022.] <https://reactgirls.medium.com/react-stylov%C3%A1n%C3%AD-komponent-fec293ce5b3d>.
20. **hunter_js**. 8 Ways to Style React Components. *GeeksforGeeks*. [Online] 17. 1. 2022.
[Citace: 7. 3. 2022.] <https://www.geeksforgeeks.org/8-ways-to-style-react-components/>.
21. **Contributors, styled-components**. styled-components/styled-components. *GitHub*. [Online] 12. 5. 2021. [Citace: 7. 3. 2022.] <https://github.com/styled-components/styled-components/blob/main/README.md>.
22. **Cruz, Rafael**. <https://medium.com/@ralph1786/using-css-modules-in-react-app-c2079eadbb87>. [Online] 11. 10. 2019. [Citace: 7. 3. 2022.]

<https://medium.com/@ralph1786/using-css-modules-in-react-app-c2079eadbb87>.

23. **Hamedani, Mosh.** Using CSS Modules in React. *Programming with Mosh*. [Online] 25. 3. 2019. [Citace: 7. 3. 2022.] <https://programmingwithmosh.com/react/css-modules-react/>.

A. Seznam přiložených souborů

Na přiloženém datovém nosiči se nacházejí následující soubory a složky:

- **MP2021-22_Svoboda-Ondrej_Sjednoceni-uzivatelskych-rozhрани-www-aplikaci-PraCe-a-PraXe.docx** – *Textová práce ve formátu MS Word*
- **MP2021-22_Svoboda-Ondrej_Sjednoceni-uzivatelskych-rozhрани-www-aplikaci-PraCe-a-PraXe.pdf** – *Textová práce ve formátu PDF*
- **Sjednocení uživatelských rozhraní WWW aplikací PraCe a PraXe.xd** – *Návrh designu v aplikaci Adobe XD*
- **Sjednoceni_uzivatelsky_rozhрани_WWW_aplikaci_PraCe_a_PraXe-Navrh_export.zip** – *Exportovaný design do souboru PDF*
- **Sjednoceni_uzivatelsky_rozhрани_WWW_aplikaci_PraCe_a_PraXe-Zdrojovy_kod.zip** – *Veškerý zdrojový kód*