

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**

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 25.02.2022

Ondřej Svoboda

Obsah

[Úvod 1](#_Toc96644551)

[1 Specifika webových aplikací 2](#_Toc96644552)

[1.1 Statické webové stránky 2](#_Toc96644553)

[1.1 Webové aplikace 3](#_Toc96644554)

[1.2 Front end 3](#_Toc96644555)

[1.2.1 React 4](#_Toc96644556)

[1.2.2 Angular 4](#_Toc96644557)

[1.3 Backend 4](#_Toc96644558)

[1.3.1 PHP 5](#_Toc96644559)

[1.3.2 Node.js 5](#_Toc96644560)

[1.3.3 ASP.NET 5](#_Toc96644561)

[1.3.4 Ruby on Rails 5](#_Toc96644562)

[2 Návrh a vývoj designu stránek 6](#_Toc96644563)

[2.1 Layout 6](#_Toc96644564)

[2.2 Typografie 6](#_Toc96644565)

[2.1 Metodiky vývoje webu 6](#_Toc96644566)

[2.1.1 Waterfall 6](#_Toc96644567)

[2.1.2 Agile 7](#_Toc96644568)

[3 Návrh úprav aplikací PráCe a PraXe 8](#_Toc96644569)

[3.1 Analýza 8](#_Toc96644570)

[3.2 Design 9](#_Toc96644571)

[4 Návrh komponent v Reactu 12](#_Toc96644572)

[4.1 Komponenta 12](#_Toc96644573)

[4.2 Stylování komponent 13](#_Toc96644574)

[4.2.1 Styled Components 13](#_Toc96644575)

[4.2.2 Bootstrap 14](#_Toc96644576)

[4.2.3 CSS Modules 14](#_Toc96644577)

[Závěr 15](#_Toc96644578)

[Seznam zkratek a odborných výrazů 16](#_Toc96644579)

[Seznam obrázků 17](#_Toc96644580)

[Použité zdroje 18](#_Toc96644581)

[A. Seznam přiložených souborů I](#_Toc96644582)

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 JS a prohloubil svoje 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.

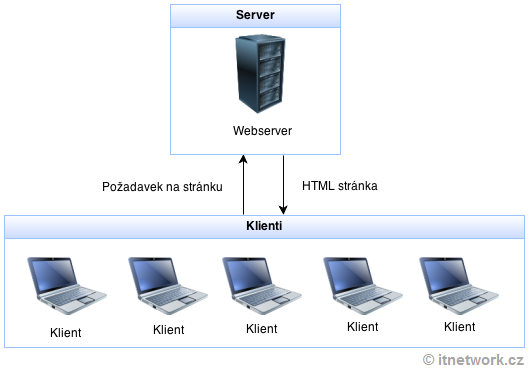
K tomuto tématu maturitní práce jsem se dostal už minulý rok během distanční výuky, když jsme začali se systémem zadávání školních dlouhodobých prací pracovat. V té době vzniklo ze strany pana Ing. Tomáše Kazdy možné zadání budoucí maturitní práce, které měl spočívat v revizi uživatelského této školní aplikace. Téma zůstalo volné a v příštím školním roce jsem se nakonec rozhodl si ho vybrat. Dalším důvodem, proč jsem si téma 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, a ne vždy respektující školní designový jazyk. Bude navrhnuto 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 Reactu do budoucna hodit.

# 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 můžou zobrazovat odlišný obsah podle určitého uživatele. Můžou například v případě našich školních aplikací nebo různých internetových e-shopů pracovat ukládáním a vyjímáním dat z databází. Další možností využití webových aplikacích je tam, kde se neustále mění obsah. Například u zpravodajských webů, kde pak není nutné při každé aktualizaci neustále měnit HTML kód webové stránky. Webové aplikace jsou jedním z pilířů takzvaného Web 2.0.

## Statické webové stránky

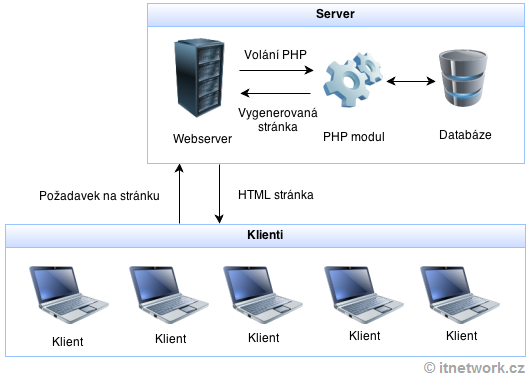


Obrázek 1- Diagram statické webové stránky

Se vznikem internetu nejdříve přišly statické webové stránky ve formě jednoduchého HTML dokumentu. Tyto HTML stránky jsou uložené 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.

## Webové aplikace



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

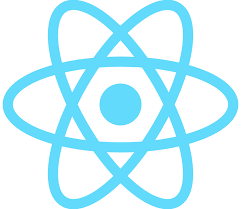
S rostoucí funkčností webu vznikly webové aplikace v podstatě jako náhrada klasických desktopových aplikací. V tomto případě to funguje tak, že se prohlížeč zeptá na určitou stránku, která je pomocí skriptů, které běží na straně serveru upravena podle požadavků uživatele. Většinou server pracuje s databází,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

## Front end

Frontend je označení pro „přední vrstu“ webové stránky, tedy to, co uživatelé vidí a s čím můžou interagovat. Většinou se zde používá jazyků HTML, CSS a JavaScript. Do frontendu 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.

### 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. Právě React je knihovna, se kterou budu pracovat při tvorbě školních aplikací.



Obrázek 3 - Logo React frameworku

### Angular

Angular je opensource platforma pro tvorbů webových aplikací vyvíjená týmem Googlu. Je velmi podobná Reactu. Rozdílem je, že používá jazyk TypeScript místo JavaScriptu.

## Backend

Opakem Front-endu je Backend. Za Backend 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 Backendu využít.

### 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 a mezi některé patří například bezpečnost, protože je náchylné například na SQL-Injection útoky. Tyto a další problémy řeší některé frameworky PHP, jako například Laravel, které se stávají čím dál populárnějšími. Mezi některé velké weby používající PHP patří například Wikipedia nebo Facebook.

### 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.

### 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.

### 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.

# 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 vracení případě provedení konverze, jako je například zakoupení produktu na e-shopu.

## Layout

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

## Typografie

Typografie se zabývá především textem na webu. Patří do ní nejen správná textace podle požadavků SEO, ale i typy a velikosti písem a například správné řádkování.

## Metodiky vývoje webu

### Waterfall

Waterfall neboli vodopádový model je metodika vývoje, kde vývoj považuje jako neustálý tok (jako tok vody ve vodopádu), který postupně prochází jednotlivých fázemi vývoje. Obsahuje 7 fází:

* Specifikace požadavků
* Návrh
* Implementace
* Integrace
* Testování
* Instalace
* Údržba

Tento model je ale kritizován hned z několika důvodů. Jedním je, že je v praxi u jakéhokoli většího projektu nemožné tímto způsobem postupovat a jednotlivou fázi dokončit dříve, než se začne pracovat na další. Naopak je výhodou může být jednoduchost nebo důraz na dokumentaci v začátečních fázích.

### Agile

Metody Agile jsou založené na postupném, inkrementálním vývoji. Vychází z kritiky vodopádového modelu, který nedokáže reagovat na problémy a změny, které vzniknou během vývoje. Zakládá na rychlém vývoji a následném otestování a upravení podle zpětné vazby.

# Návrh úprav aplikací PráCe a PraXe

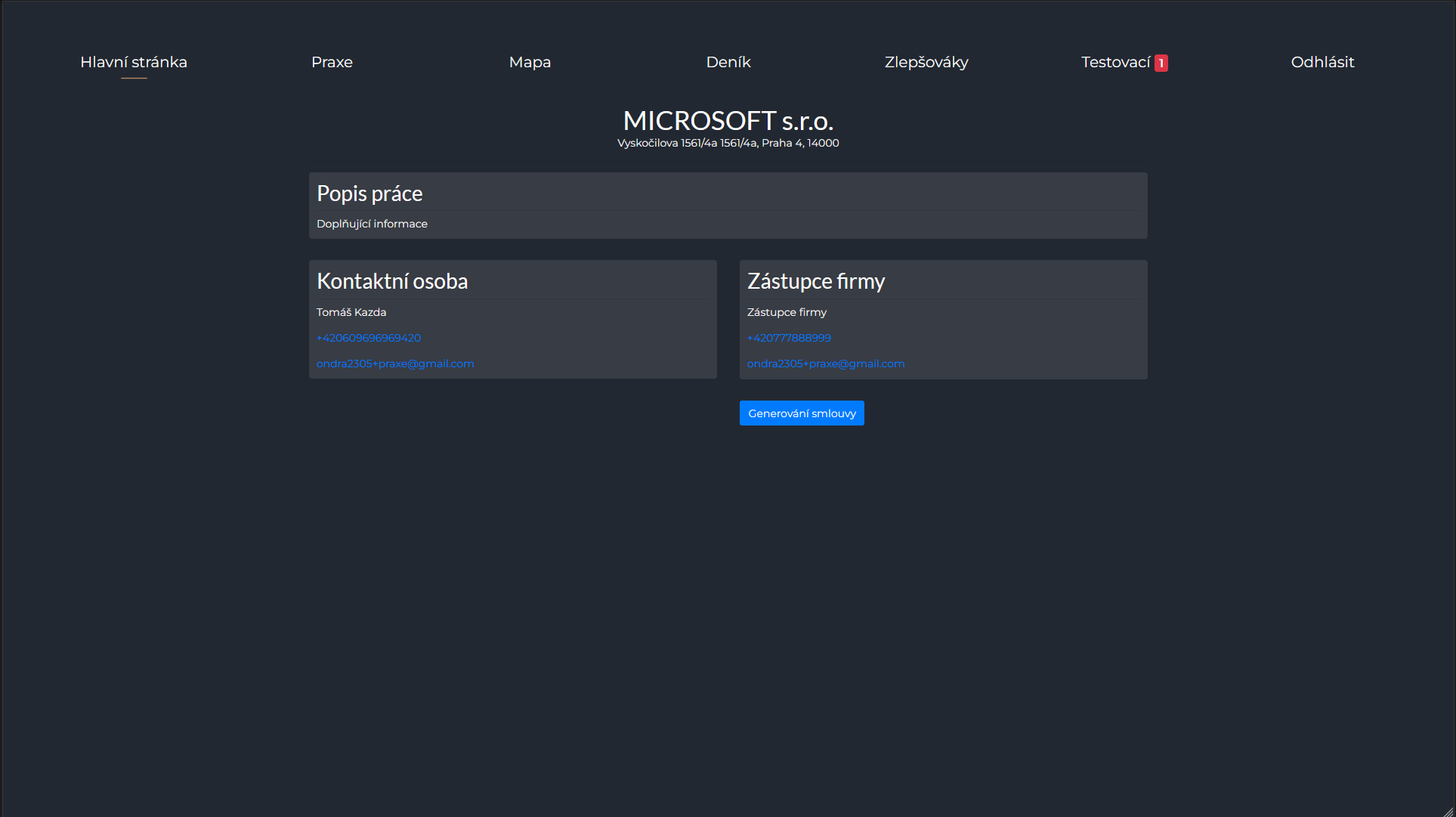
Aplikace PráCe je určená na správu a hodnocení školních dlouhodobých prací jako jsou ročníkové práce a maturitní práce. Aplikace PraXe je druhou školní webovou aplikací, která je určená na evidenci praxí. Cílem mojí práce je sjednotit aplikace tak, aby měli stejné funkční a vizuální prvky.

## Analýza

Obsah obrázku text

Popis byl vytvořen automaticky

Obrázek 4 - Původní stav aplikace PráCe



Obrázek 5 - Původní stav aplikace PraXe

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. 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 bude až posledním bodem.

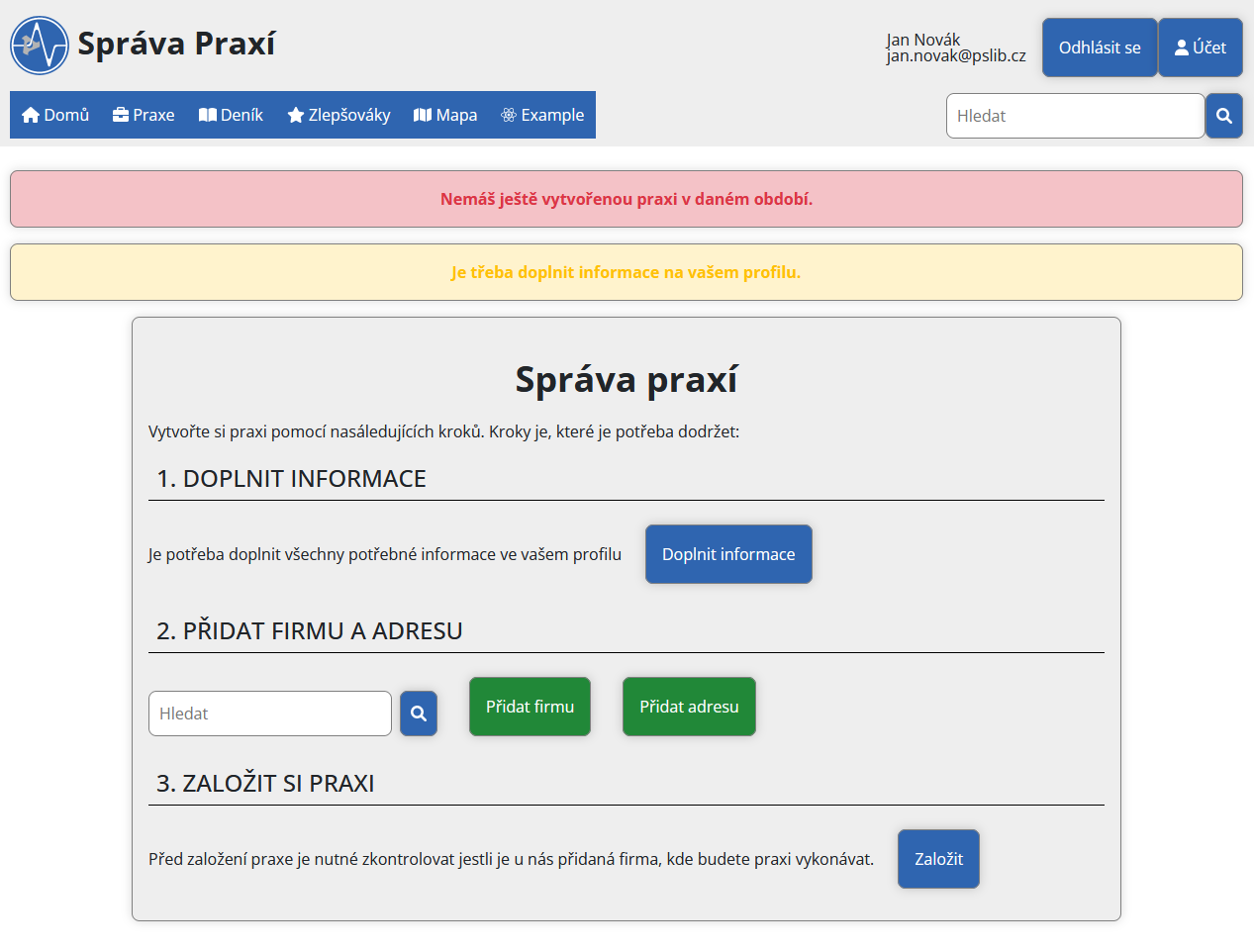
Prvním nedostatkem aplikací je na první pohled úplně odlišný design. To byl první bod, který jsem musel zohlednit při návrhu úprav. Rozhodl jsem zde použít školní barvy (tmavě modrá #2F65B0 a šedá #707070) a jako vzor jsem použil náš školní web.

Hlavním cílem změn UX bude aplikace PraXe, která má v tomto ohledu značné nedostatky. Jde hlavně o nesmyslnost rozložení prvků a celková nepřehlednost. Během testování jsem také narazil na některé zásadní chyby. Jednou z nich je, že pokud vytvoříme hodnocení praxe, není možné ho už smazat. To vedlo k pracím, které mají popisek pouze „Nelze smazat“. Další chybou, kterou 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 nanašli.“. Těmto chybám jsem se pokusil zabránit právě redesignem UX.

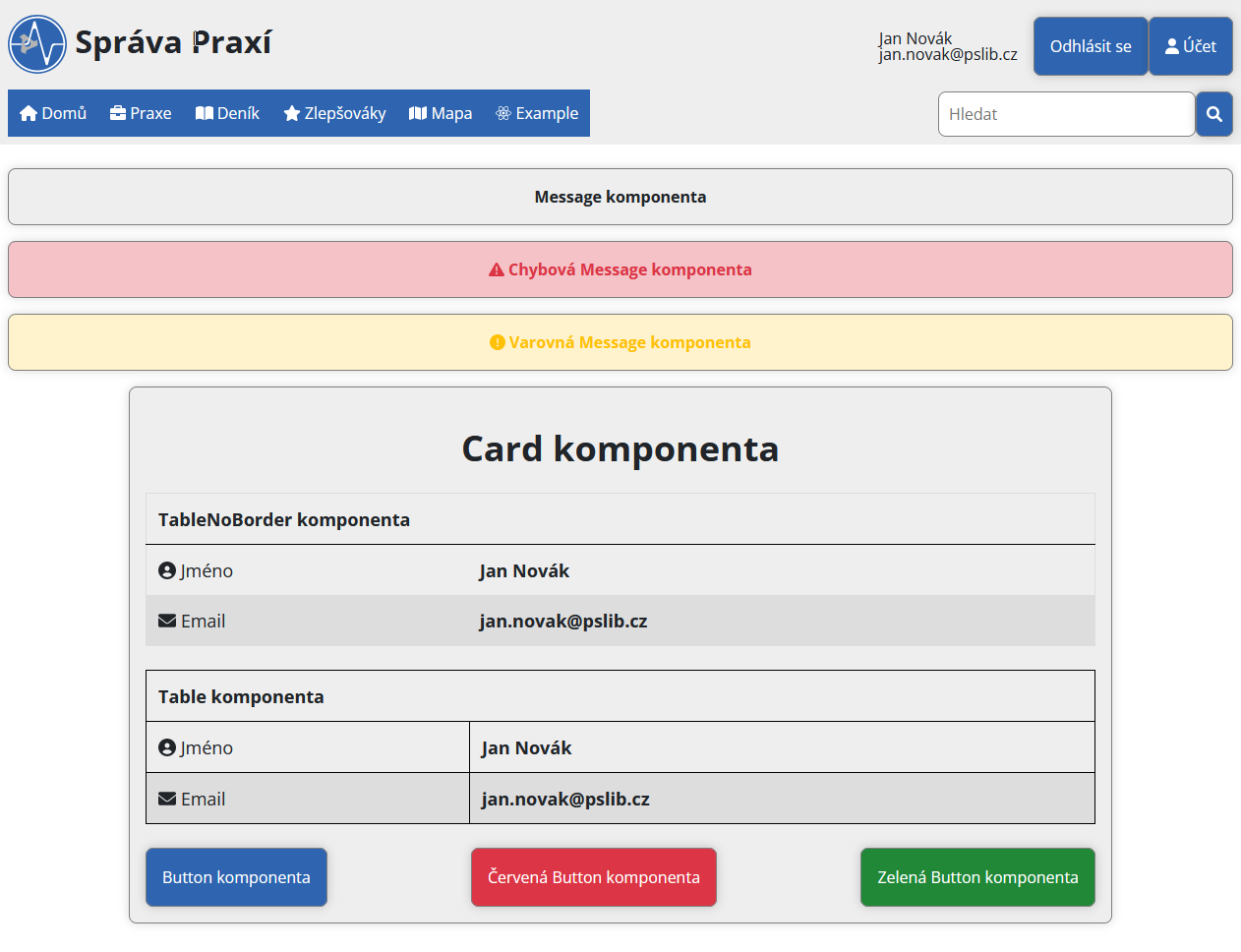
V aplikaci PráCe se změřím na vizuální změny, jako je výše zmíněná změna barevné palety, zvýraznění některých tlačítek, změna rozložení navigace webu

## 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. Pro lepší přehlednost jsem design obou aplikací navrhoval do jednoho souboru. V XD návrhu jsem vytvořil samostatné plátno jako náhled katalogu komponent, které vznikly.



Obrázek 6 - Redesign aplikace PraXe



Obrázek 7 - Ukázka vytvořených komponent

## Oauth

# Návrh komponent v Reactu

Prvním krokem byl samotné nainstalování Reactu. 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 (dependencies) pomocí jednoho ze správců balíčků pro prostředí Node.js jako je třeba například npm nebo Yarn.

Další závislosti, které jsem při tvorbě aplikace použil:

* **Create-react-app**
  + Zmíněno výše, umožňuje jednoduché vytvoření React aplikace, automaticky nainstaluje všechny nutné závislosti
* **FontAwesome**
  + React implementace populární knihovny ikon FontAwesome
* **Styled Components**
  + Knihovna na stylování v Reactu, podrobnosti popisuji níže
* **React-gh-pages**
  + Scripty na nasazení React aplikace na GitHub Pages hosting, který můžu použít jako interaktivní demo aplikace Praxe
* **React-boostrap**
  + Reactová implementace populární knihovny stylů Boostrap, použil jsem jí hlavně pro testování
* **Leaftet**
  + Komponenty React pro Leaflet (OpenStreetMaps), používá se v sekci aplikace Mapa, kde budou vidět sídla jednotlivých firem a probíhající praxe

## Komponenta

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.

class HelloWorld extends React.Component {

render() { // Vykreslení komponenty

// Vykreslení vrátí element <h1> s textem "Hello World"

return React.createElement('h1', null, 'Hello World!');

}

}

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

## Stylování komponent

Existuje několik možností stylování komponent v Reactu. Některé používání pouze JavaScript, jiné způsoby kombinují JavaScript a CSS. Je možné použít jakýkoli ze způsobů podle potřeb vývojáře a konkrétní webové aplikace.

### 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 mojí React aplikace. Každá komponenta představuje vlastní soubor a pomocí props (předaných hodnot) je možné konkrétní komponentu poté upravovat. Naopak se zde vůbec nepoužívají HTML třídy, Styled Components vytvoří vlastní „virtuální“ třídu při vygenerování HTML kódu stránky. Následující kód ukazuje příklad použítí Styled Components při tvorbě komponenty Header (hlavička stránky)

export default function Header() {

return (

<StyledHeader>

<Container>

<NavBlock>

<NavContainer>

<h1><Logo src="..." alt="PSLIB logo"/> Správa Praxí</h1>

<Nav />

</NavContainer>

<AccountInfo />

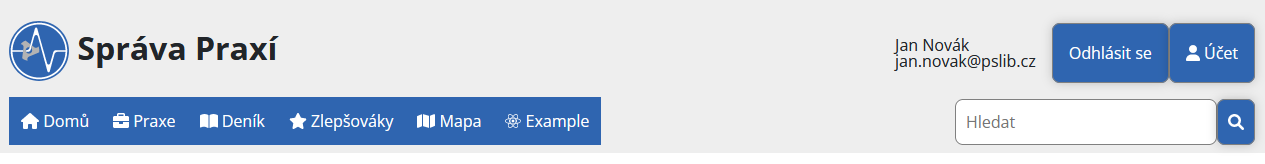
</NavBlock>

</Container>

</StyledHeader>

)

}



Obrázek 8 - Výsledek stylování komponenty Header pomocí Styled Components

Styled Components implementuje využití takzvaných „props“, což jsou proměnné pomoc kterých lze posílat určité parametry do komponenty. V praxi to pomotom umoňuje 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;

`}

### Bootstrap

Boostrap 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 použit některou z Boostrap knihoven, jako například react-bootstrap. Jsou zde předpřipravené komponenty jako například tabulky nebo formuláře, které lze jednoduše implementovat do React aplikace. Na problémy můžeme narazit, pokud chceme pro nějaké komponenty použít vlastní styly, potom je vhodné se Boostrapu vyhnout. Já jsem Bootstrap použil hlavně když jsem předběžně zkoušel rozložení prvků na stránce, ale potom jsem ho ve finální verzi odstranil, protože jsem potřeboval vlastní styly.

### 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 CSS modules je, že se snaží vyhnout konfliktům v názvech selektorů, které můžou vzniknout při větších projektech. V CSS Modules se na rozdíl od běžného chování CSS názvy selektorů vztahují pouze ke konkrétní komponentě.

Závěr

Nad prací jsem strávil několik desítek hodin. Myslím, že největším přínosem pro mě bylo získání zkušeností s Reactem, které jsem získal v podstatě z internetu a samouky, protože k tomuto tématu jsme se ve škole vůbec nedostali.

Seznam zkratek a odborných výrazů

HTML

HyperText Markup Language – značkovací jazyk používaný pro tvorbu webových stránek.

CSS

SEO

Framework

Seznam obrázků

[Obrázek 1- Diagram statické webové stránky 2](#_Toc96644583)

[Obrázek 2 – Diagram webové aplikace, v tomto případě založené na PHP 3](#_Toc96644584)

[Obrázek 3 - Logo React frameworku 4](#_Toc96644585)

[Obrázek 4 - Původní stav aplikace PráCe 8](#_Toc96644586)

[Obrázek 5 - Původní stav aplikace PraXe 9](#_Toc96644587)

[Obrázek 6 - Redesign aplikace PraXe 10](#_Toc96644588)

[Obrázek 7 - Ukázka vytvořených komponent 11](#_Toc96644589)

[Obrázek 8 - Výsledek stylování komponenty Header pomocí Styled Components 14](#_Toc96644590)

Použité zdroje

1. **[Online] https://cs.wikipedia.org/wiki/Webov%C3%A1\_aplikace.**

**2. [Online] https://helpx.adobe.com/cz/dreamweaver/using/web-applications.html.**

**3. [Online] https://www.itnetwork.cz/php/zaklady/php-tutorial-uvod-do-webovych-aplikaci.**

**4. [Online] https://cs.wikipedia.org/wiki/PHP.**

**5. [Online] https://en.wikipedia.org/wiki/Node.js.**

**6. [Online] https://developer.mozilla.org/en-US/docs/Learn/Server-side/First\_steps/Introduction.**

**7. [Online] https://cs.wikipedia.org/wiki/Vodop%C3%A1dov%C3%BD\_model.**

**8. Máca, Jindřich. Lekce 1 - Úvod do React . *itnetwork.cz.* [Online] [Citace: 13. 01 2022.] https://www.itnetwork.cz/javascript/react/zaklady/uvod-do-react/.**

**9. VOXCAFE s.r.o. Lekce 1. - React - úvod. *MIND BLOG.* [Online] 02. 10 2021. [Citace: 13. 01 2022.] https://www.voxcafe.cz/mindblog/clanky/programovani/lekce1.html.**

**10. https://cs.wikipedia.org/wiki/ASP.NET. [Online] https://cs.wikipedia.org/wiki/ASP.NET.**

**11. https://www.itnetwork.cz/csharp/asp-net-core/zaklady/tutorial-uvod-do-asp-dot-net. [Online] https://www.itnetwork.cz/csharp/asp-net-core/zaklady/tutorial-uvod-do-asp-dot-net.**

**12. https://cs.wikipedia.org/wiki/Ruby\_on\_Rails. [Online] https://cs.wikipedia.org/wiki/Ruby\_on\_Rails.**

**13. https://www.geeksforgeeks.org/8-ways-to-style-react-components/. [Online] https://www.geeksforgeeks.org/8-ways-to-style-react-components/.**

**14. https://reactgirls.medium.com/react-stylov%C3%A1n%C3%AD-komponent-fec293ce5b3d. [Online] https://reactgirls.medium.com/react-stylov%C3%A1n%C3%AD-komponent-fec293ce5b3d.**

**15. https://github.com/styled-components/styled-components. [Online] https://github.com/styled-components/styled-components.**

**16. *https://www.w3schools.com/react/react\_components.asp.* [Online] https://www.w3schools.com/react/react\_components.asp.**

**17. *https://medium.com/@ralph1786/using-css-modules-in-react-app-c2079eadbb87.* [Online] https://medium.com/@ralph1786/using-css-modules-in-react-app-c2079eadbb87.**

**18. *https://programmingwithmosh.com/react/css-modules-react/.* [Online] https://programmingwithmosh.com/react/css-modules-react/.**

**19. *https://en.wikipedia.org/wiki/Angular\_(web\_framework).* [Online] https://en.wikipedia.org/wiki/Angular\_(web\_framework).**

**20. *https://www.itnetwork.cz/javascript/angular/zaklady/uvod-do-angular-frameworku.* [Online] https://www.itnetwork.cz/javascript/angular/zaklady/uvod-do-angular-frameworku.**

**21. *https://cs.wikipedia.org/wiki/Webdesign.* [Online] https://cs.wikipedia.org/wiki/Webdesign.**

1. Seznam přiložených souborů

Na přiloženém datovém nosiči se nacházejí následující soubory a složky: