



Semestrální práce z předmětu KIV/EITM

Progresivní webová aplikace - aplikace k exkurzím do muzea

Jakub Koderaj, Jiří Noháč, Jaroslav Hrubý
koderaj@students.zcu.cz, jnohac@students.zcu.cz,
hrubyj@students.zcu.cz

22. května 2023

Obsah

1	Zadání	2
2	Implementace	3
2.1	Nasazení	3
3	Uživatelská příručka	5
3.1	Instalace	5
3.1.1	Pomocí Docker	5
3.1.2	Pomocí XAMPP webserveru	6
3.2	Používání	7
3.2.1	Simulace PWA	8
4	Závěr	9

1 Zadání

Implementujte progresivní webovou aplikaci (PWA) pro prohlížení exkurzí muzea. Aplikace umožní uživateli prohlížení exponátů. Aplikace dále umožní přihlášenému uživateli provést test o znalostech ohledně exponátů. Přihlášený uživatel si bude moci prohlédnout výsledky svých testů. Uživatel bude schopen si exponáty prohlédnout i v případě, že se odpojí od sítě. Přihlášený uživatel bude moci vyplnit test i v případě, že je odpojený od sítě a při znovu připojení se výsledky zapíší do databázového serveru aplikace.

2 Implementace

Implementace aplikace byla provedena pomocí PHP frameworku *Laravel* <https://laravel.com/>. Funkcionalita, která umožňuje z klasické webové aplikace vytvořit progresivní webovou aplikaci, je zajištěna pomocí volně dostupného pluginu do Laravel frameworku <https://github.com/silvionolleite/laravel-pwa>. Backend aplikace je tedy řešen pomocí programovacího jazyka PHP s využitím relační databáze *MySQL*. Frontend je řešen pomocí Laravel šablonovacího systému *Blade*, stylování aplikace pomocí CSS frameworku *Bootstrap 5*, a obsluha progresivní webové aplikace pomocí JavaScriptu.

Hlavní implementace, která z webové aplikace dělá progresivní webovou aplikaci, se nachází v souboru `public/serviceworker.js`. ServiceWorker registruje posluchače pro různé události. Při otevření aplikace dojde k instalaci ServiceWorkeru, při kterém dojde k uložení všech dostupných webových stránek do lokálního úložiště (do *cache*) klienta. Při zaslání HTTP requestu klientem na server, je nejprve ServiceWorkerem zkontrolováno zda je uživatel připojen k síti (tedy je online) a pokud ano, tak je požadavek přeposlán na aplikační server. V případě, že klient online není tak jedná-li se o POST request, je daný request uložen do lokálního úložiště *indexedDB*. V případě GET requestů jsou pak jednotlivé stránky vyhledávány v cachi klienta a při nalezení požadované stránky je mu zobrazena stránka z lokálního úložiště. Pokud uživatel nebyl online a nově se připojil k síti, jsou při jakémkoliv novém požadavku zaslány veškeré uložené POST requesty na aplikační server.

2.1 Nasazení

V rámci semestrální práce bylo také zkušeno nasazení aplikace pomocí poskytovatele cloudových služeb *DigitalOcean* <https://www.digitalocean.com/>. Aplikace běží na pronajatém webovém serveru s IP adresou 164.92.137.14¹. Aplikace je na serveru nasazena pomocí platformy Docker, přičemž byl použit návod <https://www.digitalocean.com/community/tutorials/how-to-set-up-laravel-nginx-and-mysql-with-docker-compose-on-ubuntu-20-04>.

¹Jelikož samozřejmě hosting není zadarmo, tak aplikace na této adrese bude dostupná pouze do odevzdání této práce. Aplikace zde bude tedy dostupná maximálně pouze do 31.6.2023

Pro funkcionalitu aplikace jako progresivní webové aplikace, je ovšem pro registraci ServiceWorkeru nutné aby bylo připojení k aplikaci zabezpečené. Pro lokální rozchození aplikace to samozřejmě není problém, ovšem pro aplikaci nasazenou na webovém serveru je nutné aby na serveru byl aktivován *SSL* certifikát. Byl na server nahrán vlastně podepsaný *SSL* certifikát, který samozřejmě není bezpečný, takže při připojení na IP adresu z webového prohlížeče je zobrazeno upozornění, že přístup na adresu <https://164.92.137.14> není zabezpečený a ServiceWorker není registrován.

Existuje způsob jak toto zabezpečení obejít a to spuštěním webového prohlížeče *Google Chrome* následujícím příkazem²:

```
1 chrome.exe --profile-directory="Profile 1" --ignore-certificate-errors --unsafely-treat-insecure-origin-as-secure=https://164.92.137.14
```

který přepínači nastaví prohlížeči Chromu, aby se choval k adrese <https://164.92.137.14> jako k bezpečné. V tomto případě se pak ServiceWorker úspěšně instaluje a je tak možné simulovat funkcionalitu PWA viz 3.2.1. Ovšem problém je, že po instalaci aplikace jako nativní, stále nefunguje registrace ServiceWorkeru, jelikož aplikace registruje adresu jako nezabezpečenou. A to i v případě, že jsou nativní aplikaci explicitně nastaveny stejné přepínače jako v případě spouštění Chromu. Na tento problém jsme řešení ovšem nenašli.

²<https://stackoverflow.com/questions/38728176/can-you-use-a-service-worker-with-a-self-signed-certificate>

3 Uživatelská příručka

3.1 Instalace

3.1.1 Pomocí Docker

Pro lokální instalaci pomocí platformy *Docker* je nejprve potřeba nainstalovat následující prerekvizity:

- Docker — volně dostupný ke stažení pro různé operační systémy na URL adrese <https://docs.docker.com/get-docker/>
- Composer — volně dostupný pro stažení na URL adrese <https://getcomposer.org/download/>

Po instalaci prerekvizit je potřeba stažení zdrojových souborů aplikace pomocí příkazů:

```
1 ~/ $ git clone https://github.com/dumby7/eitm-sp.git
2 ~/ $ cd eitm-sp
```

Instalace PHP závislostí pomocí Composer

```
1 ~/eitm-sp $ composer install
```

Vytvoření systémových proměnných aplikace pomocí kopírování předpřipraveného souboru s příkladem nastavení¹:

```
1 ~/eitm-sp $ cp .env.example .env
```

Spuštění Docker containerů, bude zde uveden příklad pomocí Laravel utility pro ovládání Docker containerů *Sail*², je ovšem možné použít klasické Docker příkazy:

```
1 ~/eitm-sp $ ./vendor/bin/sail up -d
```

Pro usnadnění je možné nastavit alias pro příkaz sail podle návodu <https://laravel.com/docs/10.x/sail#configuring-a-shell-alias>.

Překlad CSS a JS:

```
1 ~/eitm-sp $ sail npm install && sail npm run build
```

Vygenerování aplikačního klíče:

¹Upravte nastavení v souboru `.env` podle potřeby

²

```
1 ~/eitm-sp$ sail artisan key:generate
```

Vytvoření databázových tabulek:

```
1 ~/eitm-sp$ sail artisan migrate
```

Naplnění tabulek základními daty:

```
1 ~/eitm-sp$ sail artisan db:seed
```

Po instalaci je aplikace dostupná na URL adrese <http://localhost/>. Pro práci s databází pomocí uživatelského rozhraní je možné použít *PH-MyAdmin* na adrese <http://localhost:8001/>.

3.1.2 Pomocí XAMPP webserveru

Pro lokální instalaci pomocí webserveru *XAMPP* je nejprve potřeba nainstalovat následující prerekvizity:

- Composer — volně dostupný pro stažení na URL adrese <https://getcomposer.org/download/>
- Node.js — volně dostupný pro stažení na URL adrese <https://nodejs.org/en/download>
- XAMPP — volně dostupný pro stažení na URL adrese <https://www.apachefriends.org/download.html>

Po instalaci prerekvizit je potřeba stažení zdrojových souborů aplikace pomocí příkazů:

```
1 ~/$ git clone https://github.com/dumby7/eitm-sp.git
2 ~/$ cd eitm-sp
```

Instalace PHP závislostí pomocí Composer

```
1 ~/eitm-sp$ composer install
```

Vytvoření systémových proměnných aplikace pomocí kopírování předpřipraveného souboru s příkladem nastavení:

```
1 ~/eitm-sp$ cp .env.example .env
```

Nastavení připojení k lokální databázi pomocí změn v `.env` souboru:

```
1 DB_CONNECTION=mysql
2 DB_HOST=127.0.0.1
3 DB_PORT=3306
4 DB_DATABASE=eitm_sp
5 DB_USERNAME=root
6 DB_PASSWORD=
```

Překlad CSS a JS:

```
1 npm install
2 npm run build
```

Vygenerování aplikačního klíče:

```
1 ~/eitm-sp$ php artisan key:generate
```

Vytvoření databázových tabulek:

```
1 ~/eitm-sp$ php artisan migrate
```

Naplnění tabulek základními daty:

```
1 ~/eitm-sp$ php artisan db:seed
```

Spuštění aplikačního serveru:

```
1 ~/eitm-sp$ php artisan serve
```

Aplikační server je dostupný z URL adresy definované v souboru `.env` například tedy <http://localhost/>. Pro práci s databází pomocí uživatelského rozhraní je možné použít *PHPMyAdmin* na adrese <http://localhost/phpmyadmin>.

3.2 Používání

Nepřihlášený uživatel uvidí na úvodní obrazovce aplikace v hlavním menu 3 položky: **Exhibits**, **Login** a **Register**.

Kliknutím na položku **Exhibits** je uživatel přesunut na obrazovku, ve které se nachází tabulka s jednotlivými Exponáty, které aplikace Muzea nabízí. Kliknutím na položku exponátu v tabulce je uživatel přesunut na detailní zobrazení daného exponátu. Na obrazovku s detailem exponátu se uživatel také může dostat pomocí zadání unikátního názvu exponátu do vyhledávacího pole a následným kliknutím na tlačítko **Submit**.

Kliknutím na položku **Register** je uživatel přepnut na obrazovku s registrací, ve které může vytvořit nový uživatelský profil. Po vytvoření profilu je možné se do aplikace přihlásit kliknutím na položku **Login** a vyplnění formuláře s přihlašovacími údaji. Přihlášenému uživateli se dále v hlavním menu zobrazí následující dvě položky: **Test** a **Results**.

Přepnutím na záložku **Test** může uživatel vyplnit otázky týkající se exponátů dostupných na záložce **Exhibits**. Test obsahuje několik otázek, na které je možné vybrat odpověď kliknutím na předem vybranou odpověď a případnou změnou odpovědi vybráním z dostupných možností. Vybrané odpovědi je pak možné potvrdit pomocí kliknutí na tlačítko **Submit**.

Záložka **Results** obsahuje přehled všech výsledků testů přihlášeného uživatele.

3.2.1 Simulace PWA

Aplikace je prgresivní, to znamená, že kromě používání aplikace z webového prohlížeče je ji také možné nainstalovat jako nativní aplikaci. Při otevření aplikace ve webovém prohlížeči³ se na pravé straně URL adresy zobrazí tlačítko **+**, na které je možné kliknout a nainstalovat tak aplikace přímo na systém. Aplikaci je pak možné spouštět přímo ze systému.

Jakožto progresivní webová aplikace, poskytuje aplikace také možnost částečného fungování i v případě že uživatel není připojen k internetu. Ne-přihlášený uživatel tak může prohlížet exponáty i poté co se odpojí od síťového připojení. Přihlášený uživatel navíc může dále vyplňovat testy, jejichž výsledky budou po zpětném připojení k síti poslány na aplikační server.

Pro simulaci offline módu při používání lokálního webserveru (tedy aplikace běží na localhostu), je možné v prohlížeči pomocí DevTools. DevTools otevřeme klávesou **F12**, na záložce **Network** je možné vybrat hodnotu **Offline** kliknutím na rozbalovací menu **No throttling**.

³Vyzkoušené jsou pouze prohlížeče *Google Chrome* a *Microsoft Edge*, například prohlížeč *Mozilla Firefox* progresivní webové aplikace nepodporuje.

4 Závěr

V rámci semestrální práce byla vytvořena jednoduchá progresivní webová aplikace. Aplikaci je z prohlížeče možné nainstalovat i jako nativní aplikaci. Aplikace také poskytuje některou funkcionalitu i v případě, že se uživatel odpojí od síťového připojení. Jelikož se jedná pouze o *proof of concept* je funkcionalita aplikace velice jednoduchá. Pro skutečné používání aplikace by bylo potřebné dodělat další funkcionality.