

DELTA – Střední škola informatiky a ekonomie, Základní škola a  
Mateřská škola s.r.o.  
Ke Kamenci 151, PARDUBICE

## **Galerie Hřišť**

**Autor:** Marek Pavlíček

**Třída:** 4.A

**Studijní obor:** 18-20-M/01 Informační technologie

**Školní rok:** 2020/2021

# Zadání maturitního projektu z infromatických předmětů

Jméno a příjmení: *Marek Pavlíček*  
Školní rok: *2021/2022*  
Třída: *4.A*  
Obor: *Infomační technologie 18-20-M/01*

Téma práce: *Galerie Golfových Hřišť*  
Vedoucí práce: *Bc. Vlad'ka Janů*

## **Způsob zpracování, cíle práce, pokyny k obsahu a rozsahu práce:**

Cílem projektu je vytvořit webovou aplikaci pro prohlížení golfových hřišť. Bude obsahovat vyhledávání, třídění podle délky nebo třeba kraje, mapu s markery umístění hřišť, zobrazení detailu hřiště s mapou a fotkami, ukládání oblíbených hřišť. Internetové stránky budou mít 2 rozhraní, administrátorské a uživatelské. Uživatel bude moci přidat foto a další zajímavosti. Administrátor bude moci spravovat fotografie. Můj systém bude obsahovat proklik na rozpis turnajů na daném hřišti, uvidíte předpověď počasí a budete sem moci zapsat svůj výsledek.

## **Použité základní technologie:**

- Typescript
- React
- Next.js
- Firebase
- Google Maps API

## **Stručný časový harmonogram (s daty a konkretizovanými úkoly):**

**Konec září:** Dokončení návrhu struktury a designu stránek, vytvoření databázového modulu

**Říjen–Listopad:** Vytvoření Landing page, dynamická stránka pro zobrazení detailu hřiště, využití api

**Prosinec:** Kontaktní stránka, doladění úloh z předchozích měsíců

**15. leden:** Doladění práce, oprava posledních chyb, testování práce

**Leden–Březen:** Práce na dokumentaci

# **Prohlášení**

Prohlašuji, že jsem svou práci vypracoval samostatně, výhradně s použitím uvedené literatury.

## **Poděkování**

Děkuji vedoucímu projektu Bc. Vlad'ce Janů za vedení projektu a za nápady, které mi pomohly obohatit tento projekt. Dále bych chtěl poděkovat mým spolužákům za pomoc při testování funkčnosti projektu.

## **Anotace**

Cílem projektu je webová aplikace pro přidávání uživatelských fotografií vybraných golfových hřišť. Uživatelé mohou v aplikaci psát komentáře pod dané hřiště a zjistit tak jestli je v dobrém stavu. Projekt je tedy mířen na úzkou skupinu lidí, která hraje golf.

## **Klíčová slova**

Webová aplikace, Firebase, Next.js, React, Google mapy, Komentáře, Golfová hřiště

## **Anotace**

Target of this project is to make web application for uploading images of chosen golf courses. Users can write comments under selected course to find out about its state. You always want to know if the course where you want to play is in good shape. This project aims at small group of people, that play golf.

## **Keywords**

Web application, Firebase, Next.js, React, Google maps, Comments, Golf courses

# Obsah

|                                      |    |
|--------------------------------------|----|
| 1. Úvod .....                        | 8  |
| 1.1. Osobní zkušenost .....          | 8  |
| 2. Funkce.....                       | 8  |
| 2.1. Landing Page.....               | 8  |
| 2.1.1. Zobrazení a vyhledávání ..... | 8  |
| 2.1.1.2. Seznam .....                | 9  |
| 2.2. Detail page .....               | 10 |
| 2.2.1. Detail .....                  | 11 |
| 2.2.2. Hodnocení.....                | 11 |
| 2.2.3. Počasí .....                  | 12 |
| 2.2.4. Komentáře .....               | 13 |
| 2.2.5. Fotogalerie.....              | 13 |
| 2.3. Admin page .....                | 15 |
| 2.4. Navigační menu .....            | 16 |
| 2.4.1. Login/Logout.....             | 16 |
| 2.4.2. Přidat Fotku.....             | 16 |
| 2.4.3. Admin.....                    | 16 |
| 2.4.4. Kontakt.....                  | 16 |
| 2.5. Responzivita .....              | 17 |
| 3. Architektura .....                | 19 |
| 3.1. Frontend.....                   | 19 |
| 3.1.1. Google mapy.....              | 19 |
| 3.1.2. OpenWeatherMap.....           | 20 |
| 3.1.3. React Popup.....              | 20 |
| 3.2. Backend.....                    | 20 |
| 3.2.1. Databáze .....                | 20 |
| 3.2.2. Cloud storage.....            | 23 |
| 3.2.3. Firebase Authentication ..... | 23 |
| 4. Technologie .....                 | 24 |
| 4.1. React.....                      | 24 |
| 4.1.1. Komponenty .....              | 24 |

|        |                                   |    |
|--------|-----------------------------------|----|
| 4.2.   | Node.js.....                      | 24 |
| 4.3.   | Next.js.....                      | 25 |
| 4.3.1. | Automatické rozdělování kódu..... | 25 |
| 4.3.2. | SSR – server-side rendering.....  | 25 |
| 4.3.3. | HMR – Hot module replacement..... | 25 |
| 4.4.   | Typescript.....                   | 25 |
| 4.5.   | Firebase.....                     | 25 |
| 4.5.1. | Firebase Authentication .....     | 25 |
| 4.5.2. | Cloud Firestore.....              | 26 |
| 4.5.3. | Cloud Storage .....               | 26 |
| 4.6.   | styled-components.....            | 26 |
| 4.7.   | OpenWeatherMap .....              | 26 |
| 5.     | Závěr .....                       | 27 |

# 1. Úvod

Projekt Galerie Hřišť pomáhá golfistům v rozhodování, na jaké hřiště si jít zahrát. Většinou je více turnajů v jeden den a každý chce vědět, které ze hřišť je v lepším stavu. Všude na internetu jsou vidět pouze profesionální fotografie, které nemusí být aktuální. Tento projekt umožňuje přidat amatérské fotografie, tak aby je viděla širší veřejnost. Fotografie mohou obsahovat popisek zadaný uživatelem. V komentářové sekci u každého hřiště bude možné řešit různé výhody nebo nedostatky hřiště.

## 1.1. Osobní zkušenost

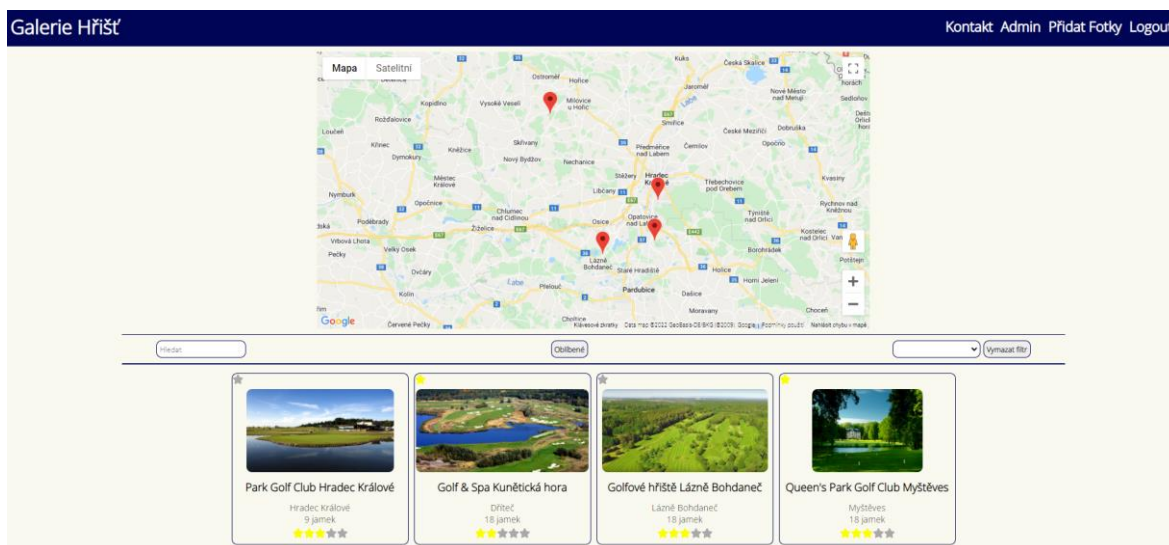
Já osobně jsem se setkal s rozdílem reality a fotografií. Jedno ze hřišť vypadalo na fotkách jinak, než jaká byla realita. Na fotkách vypadalo, že mají krásný udržovaný trávník, ale nejspíše nebyly aktuální. V realu to bylo jako hrát na louce, která nemá zavlažování, a proto nebyl vidět rozdíl mezi vyšší a nižší trávou.

## 2. Funkce

### 2.1. Landing Page

#### 2.1.1. Zobrazení a vyhledávání

Obě zobrazení jsou na jedné stránce pro lepší přehlednost. Prvním zobrazením je mapa a druhým je seznam, jak můžete vidět na obrázku číslo 1. O obou více mluvím v dalších odstavcích.



Obrázek 1 Ukázka celé Landing page

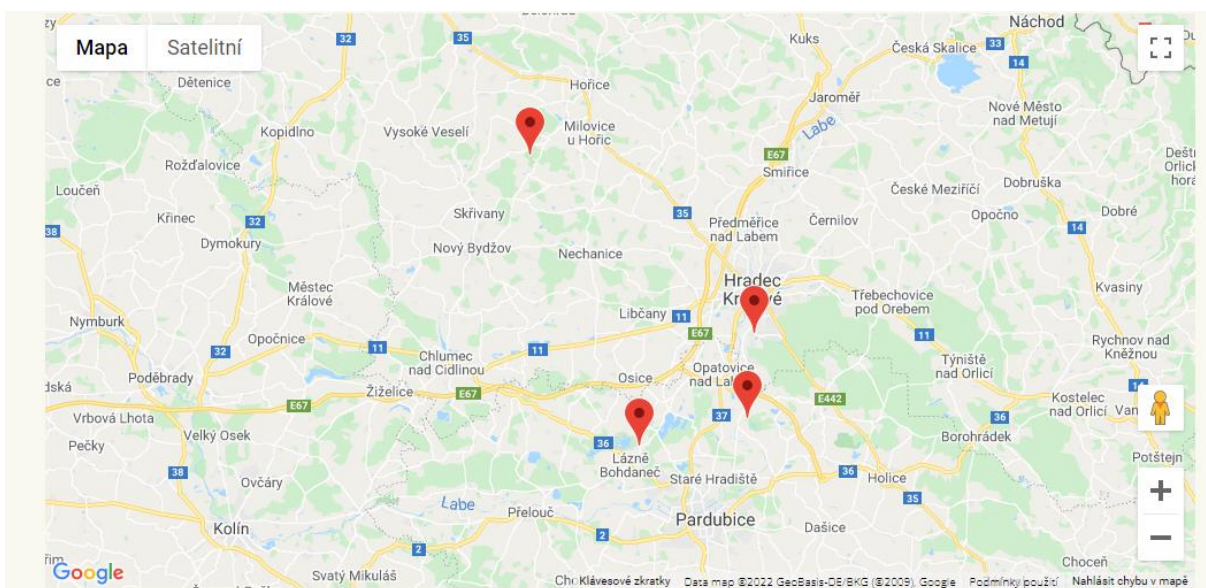
#### 2.1.1.1. Mapa

V horní části stránky je mapa, na které jsou markery na místech hřišť, jak můžete vidět na obrázku číslo 2. Každé hřiště má svůj marker na mapě.

Po kliknutí na marker se zobrazí stejné informace jako jsou u hřišť v seznamu. Kliknutím na křížek v pravém horním rohu zavře marker. Kdekoliv jinde v otevřeném



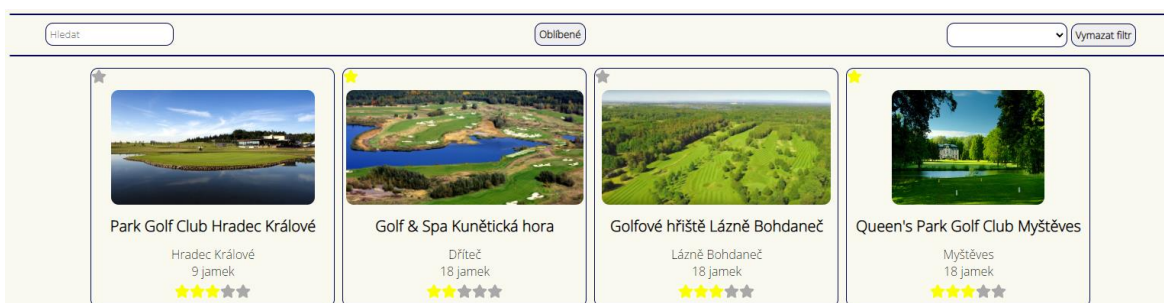
markeru vás kliknutí přesměruje na detail hřiště. Markery jsou intuitivní, jednoduché na použití a uživatelsky příjemné.



Obrázek 2 Mapa s markery

### 2.1.1.2. Seznam

V spodní části stránky je seznam hřišť, který můžete vidět na obrázku číslo 3, a ve kterém je každé hřiště s profesionální fotkou, místem a počtem jamek. Seznam je možné dynamicky rozšiřovat, to znamená, že po přidání nového hřiště do databáze bude hned viditelné. Po kliknutí na vybrané hřiště se člověk přesune na jeho detail.



Obrázek 3 Seznam hřišť

Tímto seznamem lze filtrovat dvěma způsoby. Pouze „Oblíbené“ využívá komunikaci s databází pro rychlejší funkci webu.

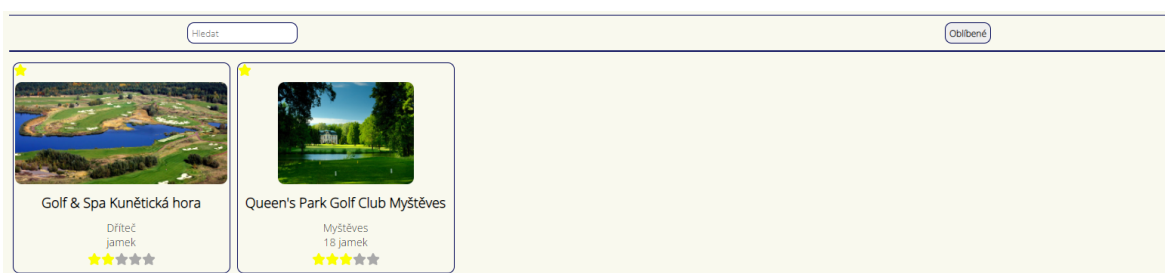
- Textovým vyhledáváním
  - Vyskytuje se v levé části pásu filtrů. Jakmile uživatel napíše písmeno, které daná hřiště obsahují zobrazí to jen je, jak je vidět na obrázku číslo

4. Stejně tak jakmile umaže nějaké písmena z textu, tak už to může zobrazit více hřišť.



Obrázek 4 Seznam po využití textového vyhledávání

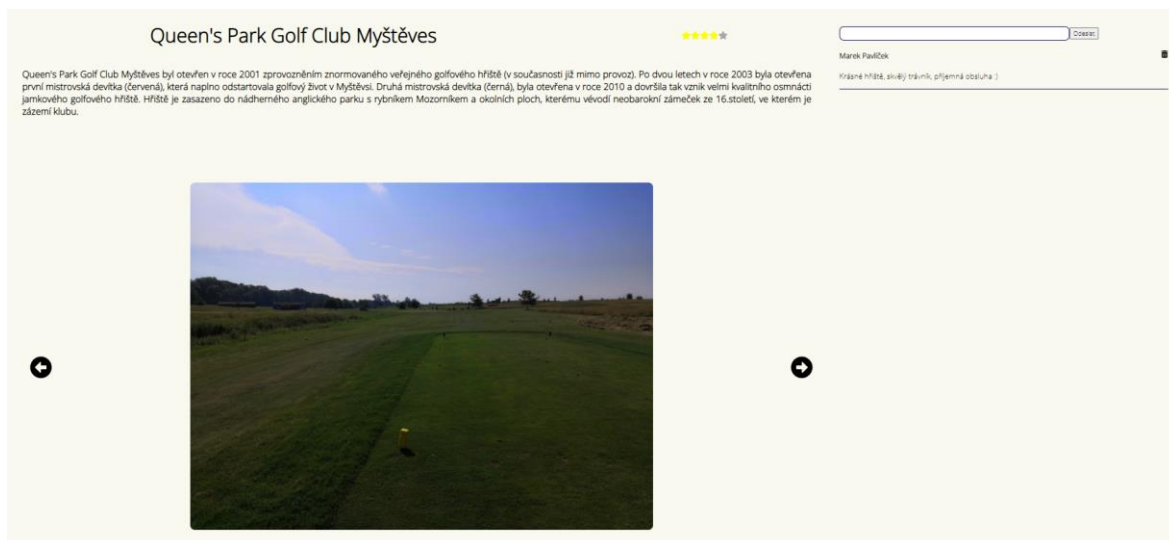
- Oblíbené
  - Po kliknutí na hvězdu u hřiště si člověk uloží dané hřiště do oblíbených a po znovunačtení stránky bude u hřiště hvězda žlutá. Následně mezi ostatními filtry je tlačítko „Oblíbené“, po kliknutí na něj se zobrazí pouze oblíbená hřiště, jak můžete vidět na obrázku číslo 5. Opětovné kliknutí filtr vypne.



Obrázek 5 Seznam po využití filtru oblíbených

## 2.2. Detail page

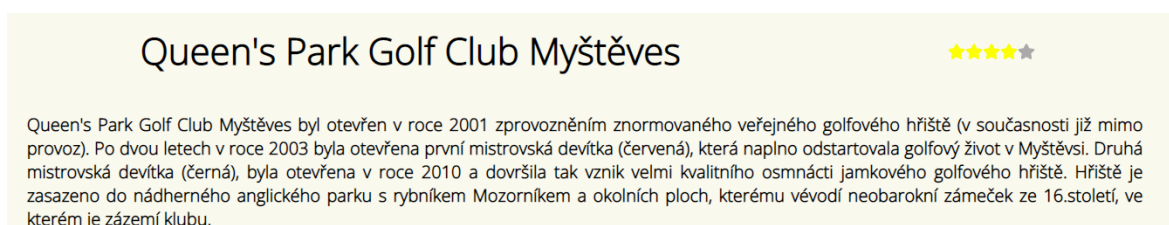
Celou detailní stránku můžete vidět na obrázku číslo 6.



Obrázek 6 Ukázka celé detail page

### 2.2.1. Detail

V levé části stránky se nachází samotný detail, kde jsou zajímavosti o hřišti (kdo a kdy jej založil, kdo jej designoval, lokalita...) a jeho název pro dobrou identifikaci na stránce jakého hřiště se nacházíte, jak můžete vidět na obrázku číslo 7. Obsahuje zároveň hodnocení, které je popsáno v dalším odstavci.



Obrázek 7 Popis hřiště

### 2.2.2. Hodnocení

Tato část je viditelná pouze pokud je uživatel přihlášen. Vyskytuje se vedle názvu stránky. Hřišti je možné dát od 1 do 5 hvězd. Počet obdržovaných hvězd je přímo úměrný počtu žlutých hvězd po kliknutí na jednu z nich. Do databáze se uloží hodnocení navázané na ID uživatele, aby nedocházelo k vytváření nových objektů. Po návratu na hřiště uvidíte vaše předchozí hodnocení. Hodnocení všech uživatelů se průměruje a je zobrazeno jak na této stránce, tak v seznamu hřišť a můžete vidět oba případy na obrázcích 8 a 9.



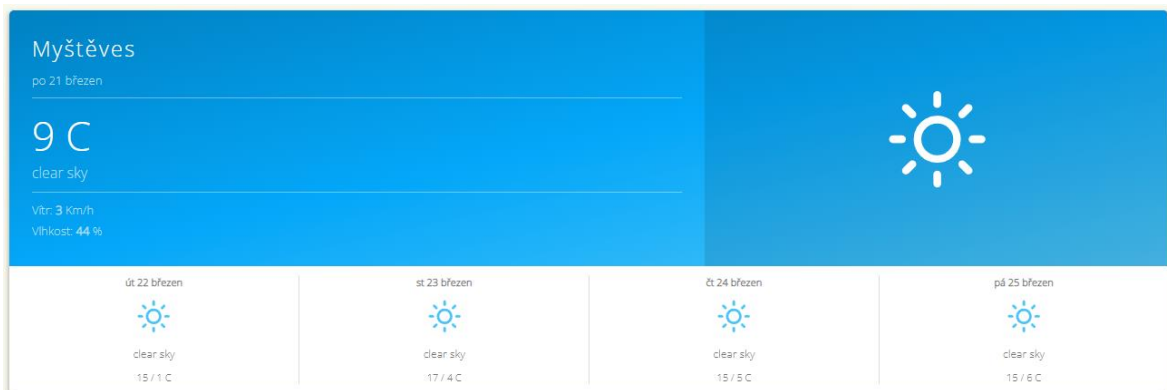
Obrázek 8 Hodnocení na detail page



Obrázek 9 Průměrné hodnocení na Landing page

### 2.2.3. Počasí

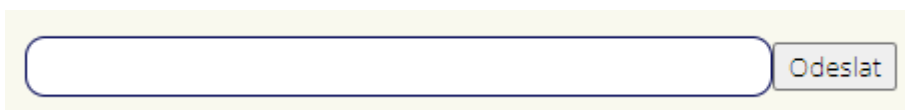
K zobrazení předpovědi počasí využívám knihovnu react-open-weather, která získává data z OpenWeatherMap API. React-open-weather tato data zpracovává pomocí API klíče a poskytuje elegantně vypadající rozhraní pro zobrazení dat, které je možné si upravit, jak můžete vidět na obrázku číslo 10. Tato knihovna neposkytuje rozhraní pro jazyk Typescript, proto jsem musel použít Javascript.



Obrázek 10 Počasí

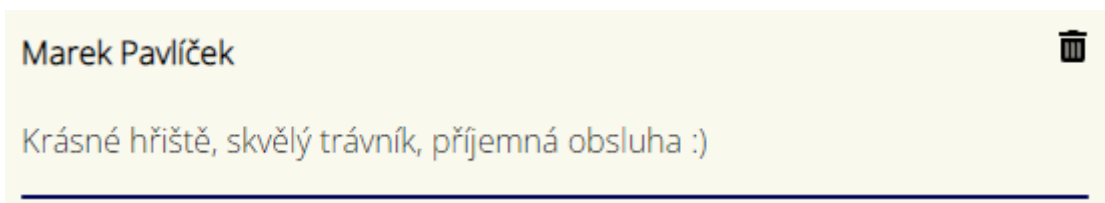
## 2.2.4. Komentáře

Ke každému hřišti je možné napsat komentář. Pole pro napsání komentáře se zobrazí pouze přihlášenému uživateli, jak můžete vidět na obrázku číslo 11. Uživatel může napsat jakýkoliv komentář, pozitivní či negativní. Pro odeslání musí kliknout na tlačítko „Odeslat“. Poté co jej odešle se mu zobrazí na stránce.



Obrázek 11 Pole pro přidání komentáře

Po odeslání komentáře může člověk svůj komentář vidět po zapsání do databáze, jak můžete vidět na obrázku 12 a může jej do jednoho dne smazat. Toto časové omezení je zde z důvodu, aby se dalo podívat do minulosti, jestli hřiště dříve bylo dobré a nyní již ne nebo naopak. Administrátor může mazat jakékoliv komentáře, kvůli možným explicitním výrazům.

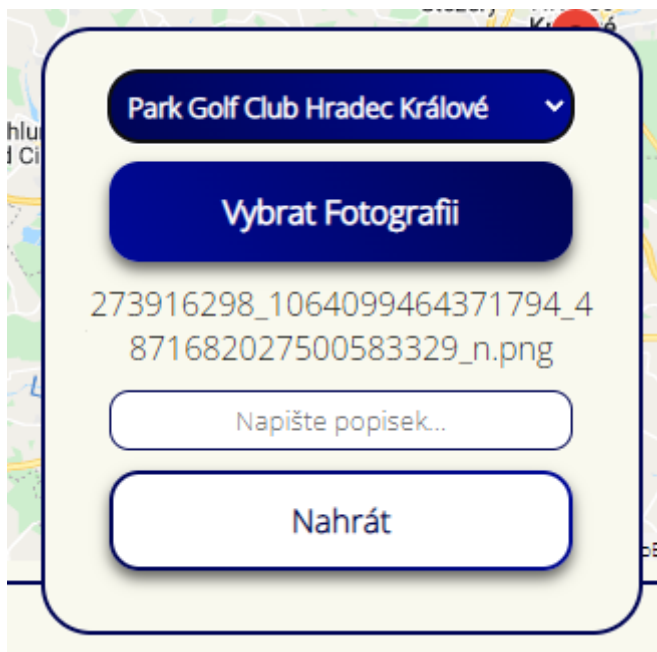


Obrázek 12 Komentář

## 2.2.5. Fotogalerie

Stěžejní částí projektu je uživatelská fotogalerie. Přidávat fotografie mohou pouze přihlášení uživatelé, kterým se v navigačním menu ukáže tlačítko „Přidat fotku“.

Po kliknutí na již zmíněné tlačítko se zobrazí formulář pro přidání fotografie, který můžete vidět na obrázku číslo 13.

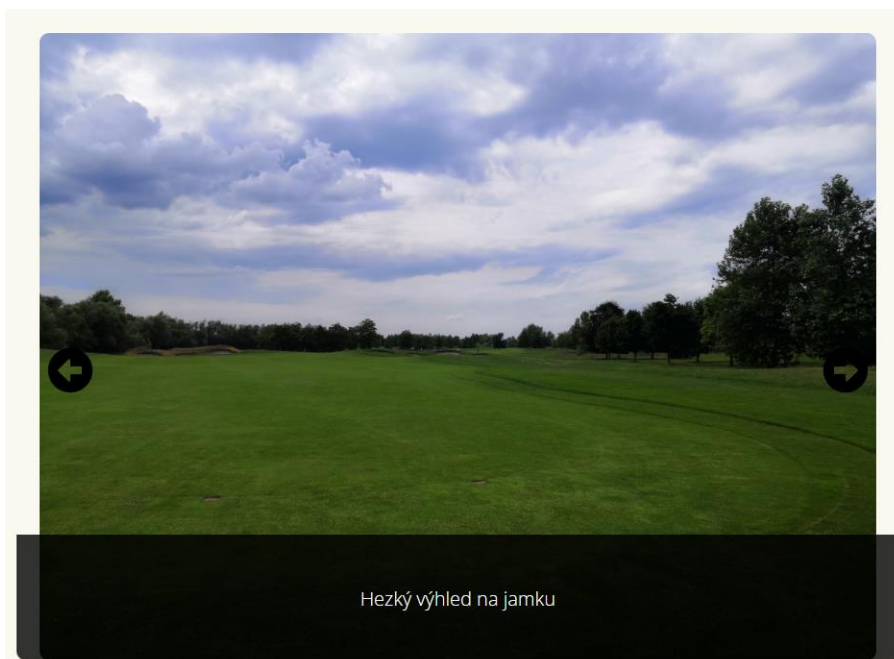


Obrázek 13 Přidání fotografie

Uživatel si vybere v selectu jedno z hřišť. Následně si kliknutím na tlačítko „Vybrat fotografii“ vybere fotografii, kterou chce přidat. Může napsat komentář a poté již jen klikne na tlačítko „Přidat“, kdy se přidá fotografie a uživatel počká na schválení administrátorem.

Po přidání fotografie ji admin na své stránce zkontroluje, zda je závadná, či ne. Pokud uzná za vhodné, že fotografie nebo její popis je nezávadný a potvrdí její zobrazení, uživatel ji uvidí mezi ostatními fotografiemi.

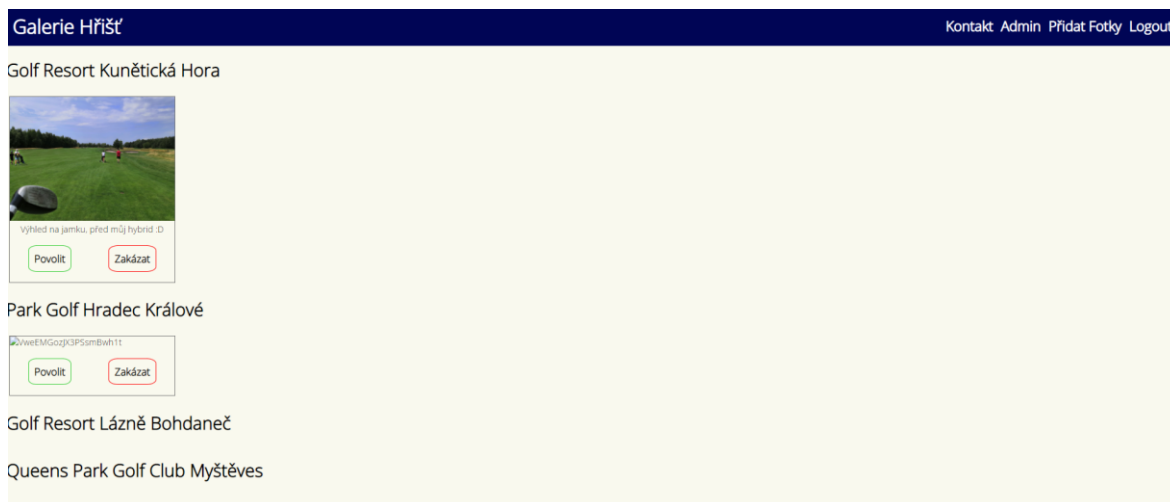
Fotografie se zobrazují v posuvníku (anglicky slider). V jednu chvíli se zobrazuje jedna fotografie s jejím popisem a můžete nimi projíždět šipkami po stránkách daných fotografií, jak můžete vidět na obrázku číslo 14.



Obrázek 14 Fotografie s popiskem v posuvníku

## 2.3. Admin page

Na tuto stránku se může dostat pouze administrátor. Pouze on ji vidí v navigačním menu. Pokud se na ni pokusí dostat uživatel bez administrátorských práv, tak bude přeměrován na Landing page. Jak vypadá tato stránka můžete vidět na obrázku číslo 15.



Obrázek 15 Ukázka celé admin page

Všechny fotografie, které uživatelé přidají musí být schváleny na této stránce. Administrátor vidí fotografii a pod ní popisek a má na výběr, jestli ji chce povolit nebo zakázat. Pokud ji povolí bude normálně viditelná na stránce vybraného hřiště.

## 2.4. Navigační menu

Navigační menu se vyskytuje na vrchu každé stránky. Na obrázku číslo 16 můžete vidět, jak vypadá před přihlášením a na obrázku číslo 17 jak po něm. V levé části je název stránky „Galerie Hřišť“, kterým se můžete prokliknout na Landing page. O odkazech v pravé části budu mluvit v příštích odstavcích.



Obrázek 16 Navbar před přihlášením



Obrázek 17 Navbar po přihlášení

### 2.4.1. Login/Logout

Jeden z odkazů je pro přihlášení uživatele, aby mohl využívat funkce stránky. Po kliknutí na odkaz se uživateli ukáže vyskakovací okno pro přihlášení. Ve vyskakovacím okně se uživatel přihlásí přes jeho Google účet.

Poté když je uživatel přihlášen se odkaz na přihlášení změní na odhlášení. Po kliknutí je uživatel odhlášen a tento web si automaticky pomatuje přes který účet se přihlásil a při příštím přihlašování již nebude muset zadávat své údaje.

### 2.4.2. Přidat Fotku

Tento odkaz je již zmíněn v odstavci „2.5.5. Fotogalerie“.

### 2.4.3. Admin

Toto je odkaz na stránku zmíněnou v odstavci „2.3. Admin page“.

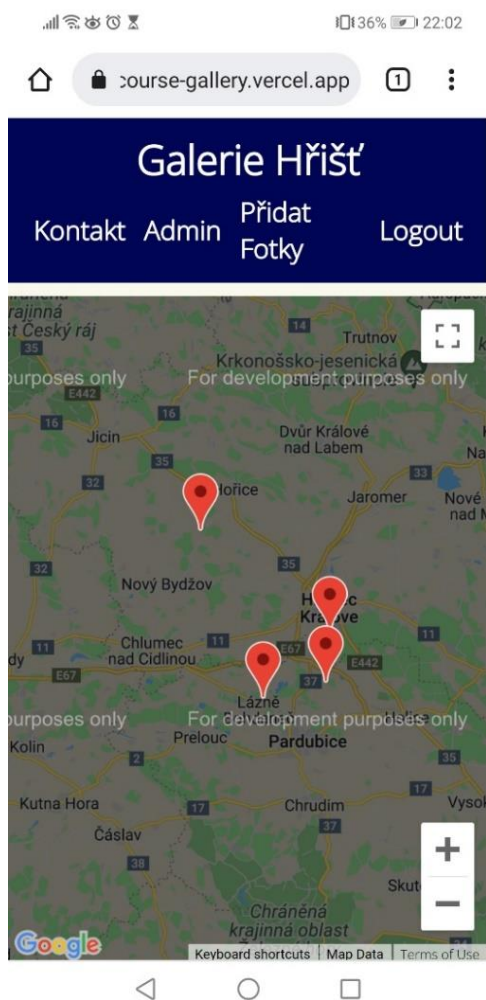
### 2.4.4. Kontakt

Proklikem na tento odkaz se dostane uživatel na můj Facebook, kde mě může kontaktovat.

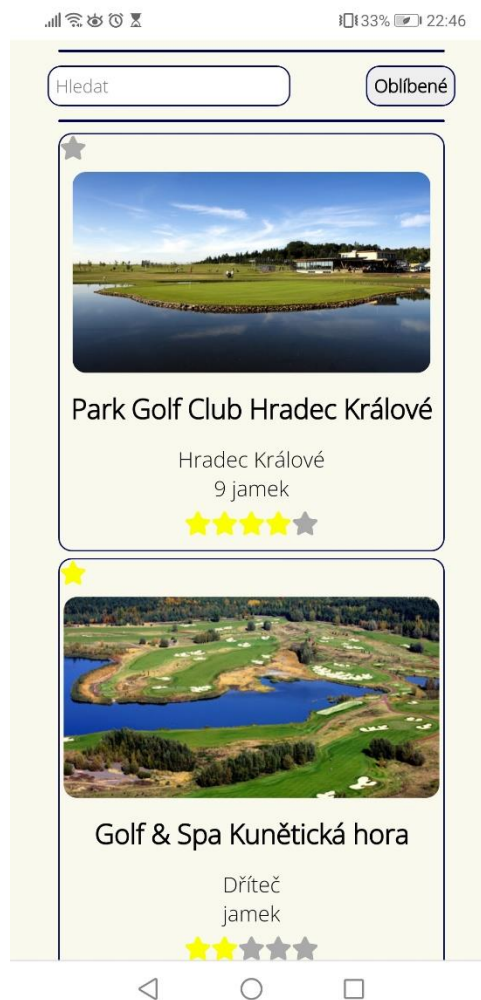


## 2.5. Responzivita

Stránky pro uživatele jsou responzivní jak pro menší desktopové zařízení, tak pro mobilní telefony nebo dokonce tablety. Zde na obrázcích číslo 18 a 19 můžete vidět responzivní Landing page.



Obrázek 19 Mapa a navigační menu v mobilní responzi



Obrázek 18 Seznam s filtry v mobilní responzi

Na obrázcích 20 a 21 můžete vidět jak se tok stránky, který byl zleva do prava rozdělen na 2 části, přesunul na klasickou stránku, kde je vše pod sebou.

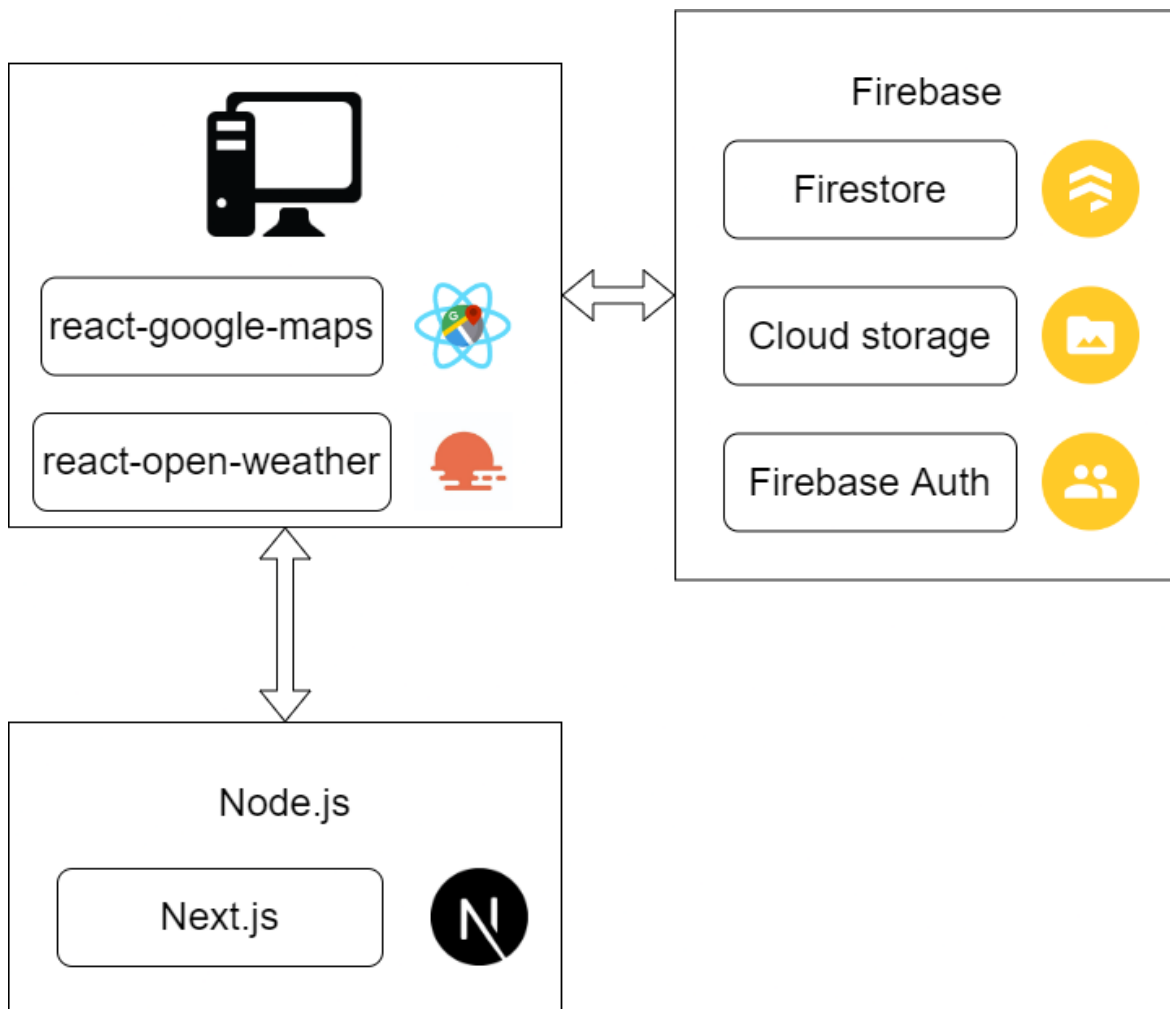


Obrázek 21 Detail s hodnocením v mobilní responzi



Obrázek 20 Posuvník a komentáře v mobilní responzi

## 3. Architektura



Obrázek 22 Architektura aplikace

### 3.1. Frontend

Frontend aplikace využívá framework Next.js, který je nadstavbou JavaScriptové knihovny React. Tento framework umožňuje vývoj webových aplikací. Hlavními výhodami tohoto frameworku jsou rychlý vývoj, rychlost načítání stránek nebo řešení problémů, protože pokud máte nějaký problém, tak na něj většinou už na internetu někdo narazil.

#### 3.1.1. Google maps

Na stránce [console.cloud.google.com](https://console.cloud.google.com) jsem si založil projekt díky kterému jsem získal api klíč pro mou mapu. Tato služba je bezplatná po určitou dobu a Google vás včas upozorní, kdy máte začít platit.

Mapu získávám z knihovny pro React zvané `@react-google-maps/api`. Tato knihovna poskytuje jednoduchou implementaci map od společnosti Google do vaší stránky. Je to přepsaná původní `react-google-maps` knihovna, která bohužel již nemá podporu. Tato knihovna má hodně možností pro developery, jak si upravovat svou mapu. Markery používám defaultní a v komponentě s mapou jim předávám potřebná data z databáze.

### 3.1.2. OpenWeatherMap

OpenWeatherMap je firma, která poskytuje různorodá data, která se točí kolem počasí. Já konkrétně využívám týdenní předpověď, která je osekána o nějaká přebytečná data. API klíč je pro mé potřeby zdarma. Kdybych chtěl tento web rozšířit např. pro celou republiku, tak bych si musel zaplatit více přístupů denně.

React-open-weather je knihovna která vezme data z API pomocí mého klíče a dosadí je do funkce `useOpenWeather()`. Tato funkce přijímá parametry `key`, který přijímá API klíč, `lat`, `lon`, `lang` a `unit`. `lat`, `lon` jsem dosadil z databáze a `unit` jsem nastavil na `metric`. Pokud chybí váš jazyk mezi podporovanými tvůrcem, tak je zde používán balíček, který poskytuje spolehlivé překlady do většiny jazyků. Takže nakonec zde mám z části češtinu, protože nejde nastavit u všech atributů.

Komponenta `ReactWeather` přijímá data z dříve zmíněné funkce. Je stylovatelná i přes knihovnu `styled-components`. Za zmínku stojí parametr `showForecast`, který pouze jeho napsáním přidá týdenní předpověď k dnešnímu počasí.

### 3.1.3. React Popup

Formulář na přidávání fotografií je zobrazen knihovnou `React Popup`, která poskytuje upravitelná vyskakovací okna. Je jednoduchá na implementaci i následné stylování.

## 3.2. Backend

Backend aplikace je postaven na takzvané serverless architektuře. Veškerý backend funguje na cloudových technologiích využívající servery od Googlu, díky čemuž není potřeba vytvářet vlastní server. V tomto cloudu je obsažena celá logika backendu a CRUD neboli komunikace aplikace s databází.

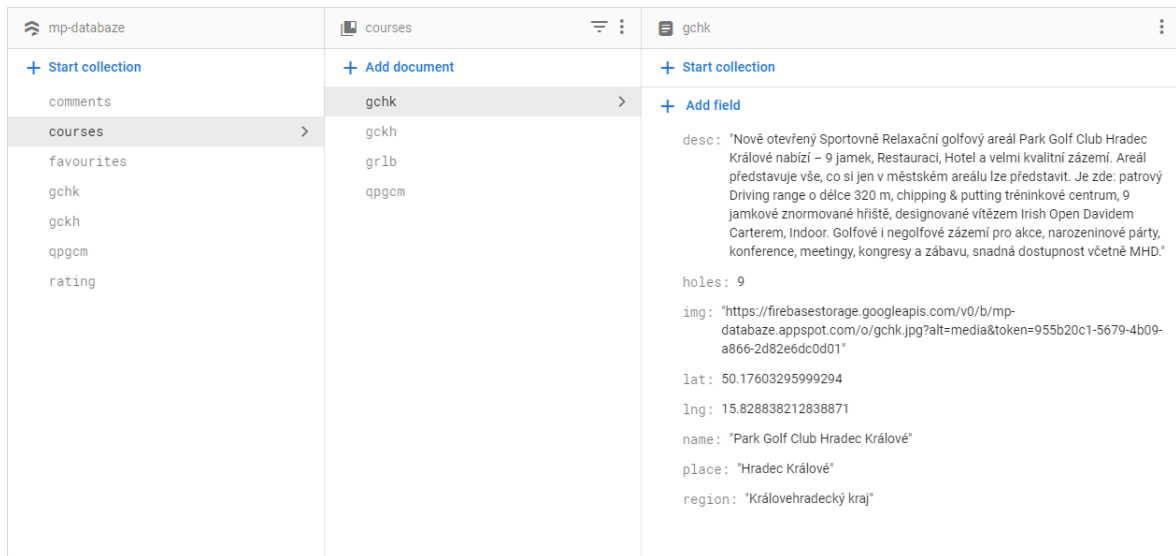
### 3.2.1. Databáze

V mé aplikaci využívám databázi od Firebase jménem `Cloud Firestore`, kterou vyvíjí společnost Google. Poskytuje knihovnu pro React, která prací se samotnou databází značně usnadňuje a urychluje práci s ní a ostatní funkce.

`Firestore` je NoSQL databáze a jako většina takových databází využívá systém kolekcí a dokumentů. Kolekce je nejvyšší prvek, který obsahuje dokumenty. Dokumenty již obsahují data psaná typem „klíč: hodnota“.

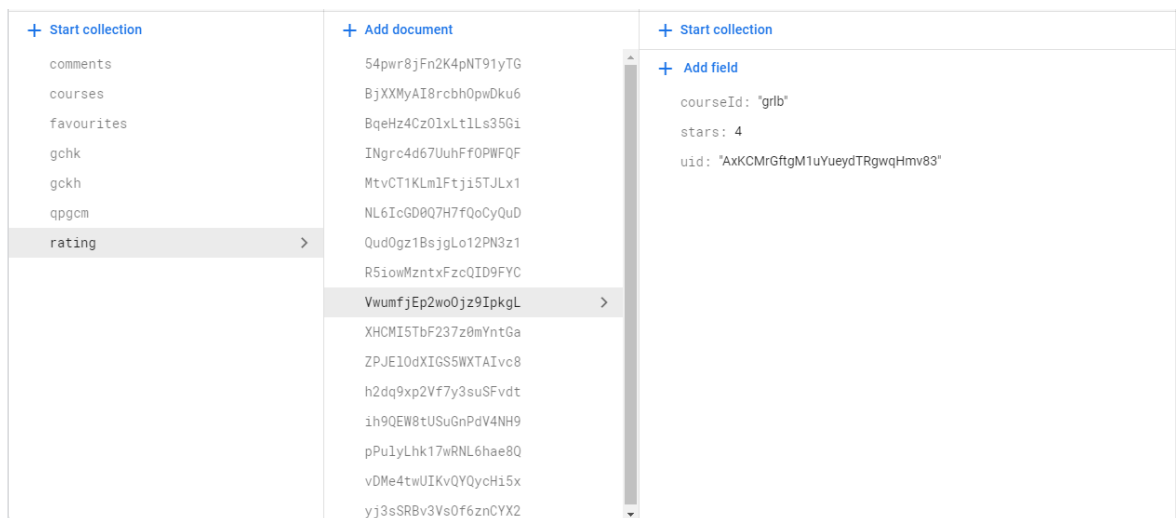
Má databáze momentálně obsahuje 1 kolekci, která se nijak nemění. 2 kolekce ve kterých se dynamicky přidávají a upravují dokumenty. Do jedné z kolekcí se data přidávají v dokumentech s uživatelským `UID`. Poté jsou tu kolekce, které nesou název podle zkratk hřišť a jsou v nich uložené údaje o fotografiích.

Nejdříve budu mluvit o neměnné kolekci `courses`. Jsou v ní dokumenty s názvy podle zkratk pro hřiště, které také slouží jako unikátní identifikátory dokumentů. Na obrázku číslo 23 můžete vidět architekturu této kolekce.



Obrázek 23 Collection courses

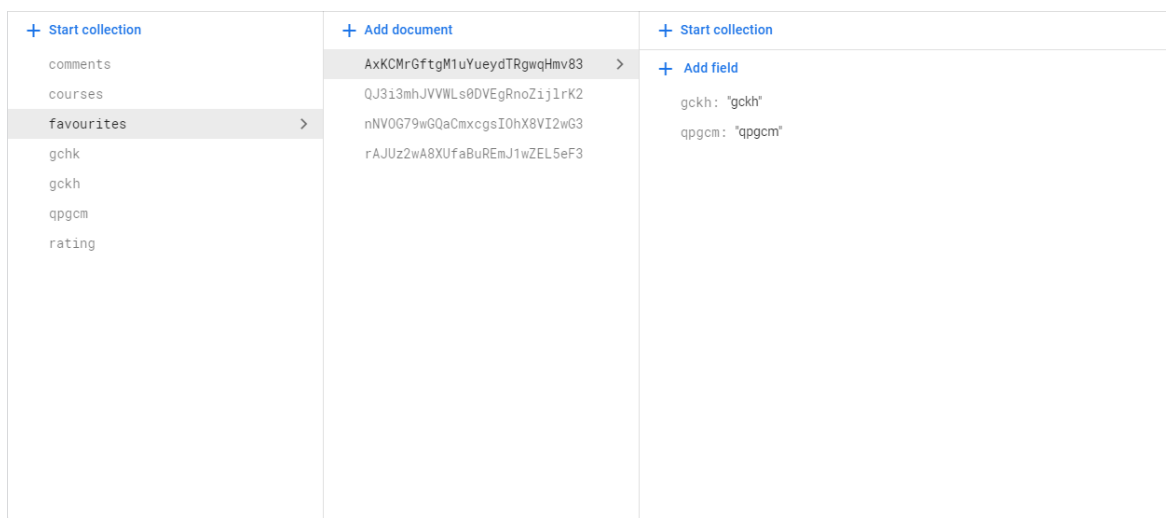
Nyní bych rád mluvil o zmíněných 2 kolekcích comments, rating. Tyto kolekce jsou strukturované stejným způsobem. Všechna data do těchto kolekcí jsou přidávána uživateli z aplikace díky funkci addDoc(), kterou poskytuje právě zmíněná knihovna firebase pro React. Zde mají již dokumenty ID přidělené automaticky, protože na něm nezáleží. Dokumenty také vždy obsahují ID hřiště a uživatele pro další využití v kódu. Využití v kódu u komentářů je například aby mohl uživatel mazat své komentáře funkcí deleteDoc(). A u hodnocení je to kvůli tomu, aby se nemohl hodnocení opakovat, ale pouze se měnily funkcí updateDoc() Jak vypadají dokumenty přidávané do databáze s automatickým ID demonstruji na obrázku 24 kolekce rating.



Obrázek 24 Collection rating

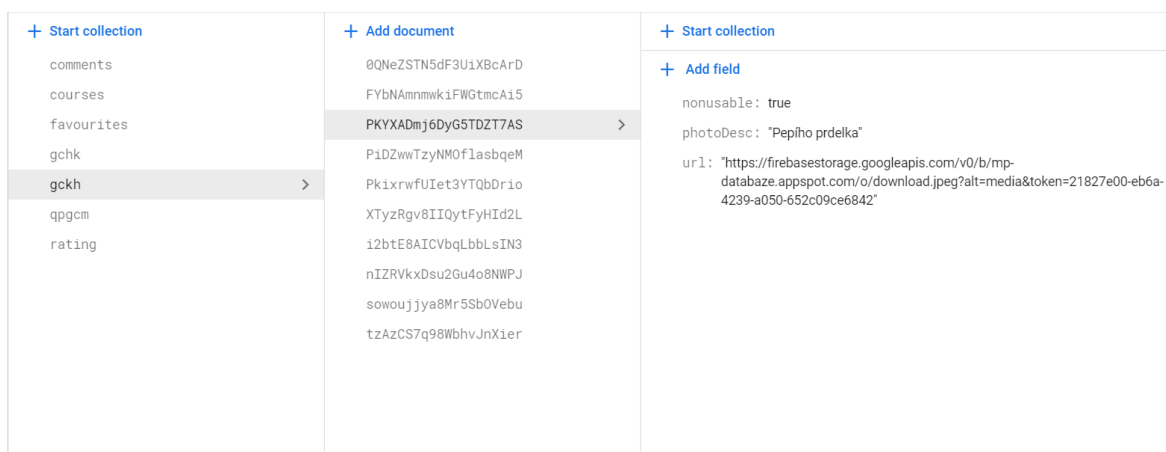
Zmíněnou kolekcí, kam se přidávají dokumenty s uživatelským ID je favourites. To znamená, že ID dokumentu je stejné jako uživatelo, pro rychlejší a přehlednější tahání dat z databáze. Další zajímavostí o této kolekci je, že do ní přichází data, která

mají klíč i hodnotu totožnou. Je to z toho důvodu, že poslaná data musí vždy být nenulová, to znamená „klíč: hodnota“ i přesto, že zde by stačilo prázdné pole neboli klíč bez hodnoty. Tuto strukturu můžete vidět na obrázku číslo 25.



Obrázek 25 Collection favourites

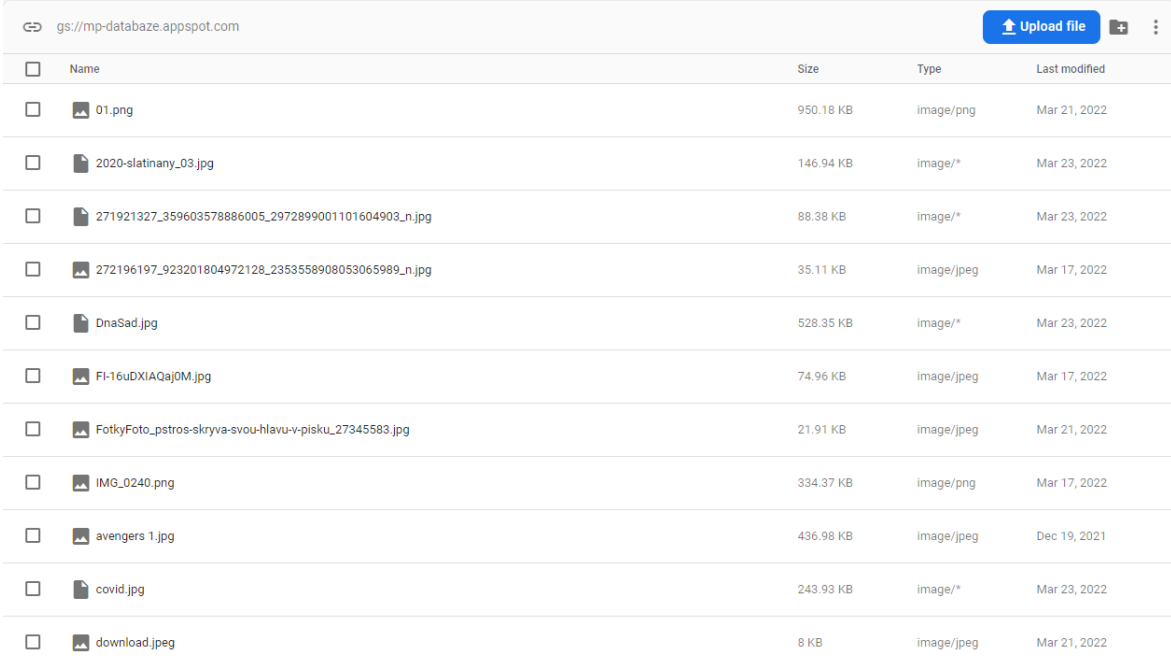
Posledními kolekce jsou kolekce, které se jmenují podle ID dokumentu hřiště neboli jeho zkratky (např. gckh, ppgcm...). Tyto kolekce opět mají automaticky generované ID dokumentů. Pole, která se přidají po nahrání fotografie skrz formulář do Cloud Storage a samotné databáze, jsou photoDesc a url. Když admin na své stránce povolí danou fotografii k zobrazení přidá se jí pole validated s hodnotou true. Pokud ji z nějakého důvodu zamítne nastaví se pole nonusable na true a nezobrazí se na stránce hřiště. Fotografie se nemažou z důvodu, že by mohlo dojít k pochybení a administrátor by se mohl ukliknout, tak aby se to dalo zpětně změnit. Pokud bude fotografie smazaná naschvál, tak do budoucna bude přidána funkce, která bude každý měsíc mazat fotografie s jejich příslušnými dokumenty. Možnost kdy administrátor neschválil z nějakého důvodu fotografii můžete vidět na obrázku číslo 26.














Obrázek 26 Course ID Collections

## 3.2.2. Cloud storage

Cloud storage je úložiště pro různé typy souborů, které si vývojář zvolí, ale většinou je používané pro fotografie nebo videa. Toto úložiště je do nějakých limitů zcela zdarma, ale ty pro tento projekt postačí. Já osobně využívám toto úložiště k ukládání fotografií. Pokud uživatel přidá jakýkoliv jiný soubor, tak by to nemělo udělat nic, protože jak je vidět na obrázku číslo 26, tak používám pouze odkaz na dané soubory. Fotografie neukládám do zvláštních složek, protože mi to v rámci tohoto projektu nepřišlo potřeba. Jak vypadá samotné úložiště můžete vidět na obrázku číslo 27.













| <input type="checkbox"/> | Name   | Size      | Type       | Last modified |
|--------------------------|--|-----------|------------|---------------|
| <input type="checkbox"/> |  01.png   | 950.18 KB | image/png  | Mar 21, 2022  |
| <input type="checkbox"/> |  2020-slatinany_03.jpg                                    | 146.94 KB | image/*    | Mar 23, 2022  |
| <input type="checkbox"/> |  271921327_359603578886005_2972899001101604903_n.jpg      | 88.38 KB  | image/*    | Mar 23, 2022  |
| <input type="checkbox"/> |  272196197_923201804972128_2353558908053065989_n.jpg      | 35.11 KB  | image/jpeg | Mar 17, 2022  |
| <input type="checkbox"/> |  DnaSad.jpg   | 528.35 KB | image/*    | Mar 23, 2022  |
| <input type="checkbox"/> |  FI-16uDxIAQaj0M.jpg                                      | 74.96 KB  | image/jpeg | Mar 17, 2022  |
| <input type="checkbox"/> |  FotkyFoto_pstros-skryva-svou-hlavu-v-pisku_27345583.jpg | 21.91 KB  | image/jpeg | Mar 21, 2022  |
| <input type="checkbox"/> |  IMG_0240.png   | 334.37 KB | image/png  | Mar 17, 2022  |
| <input type="checkbox"/> |  avengers 1.jpg   | 436.98 KB | image/jpeg | Dec 19, 2021  |
| <input type="checkbox"/> |  covid.jpg  | 243.93 KB | image/*    | Mar 23, 2022  |
| <input type="checkbox"/> |  download.jpeg  | 8 KB      | image/jpeg | Mar 21, 2022  |

Obrázek 27 Architektura storage

## 3.2.3. Firebase Authentication

Využívám přihlášení přes Google, protože s ním Firebase dobře pracuje. Nemusím vytvářet nové objekty v databázi, protože všechna data o uživateli si drží u sebe přímo v sekci Authentication. Tyto data o uživateli následně získávám funkcí `onAuthStateChanged()`, která data drží díky React hooku `useContext()` ve funkci `useAuth()`. Tuto funkci můžeme destruovat pro získání objektu `user` a z něj už můžeme získávat data jako je ID nebo jméno. Jak vypadá seznam uživatelů ve vaší aplikaci můžete vidět na obrázku číslo 28.

| Identifier                 | Providers   | Created ↓    | Signed In    | User UID                      |
|----------------------------|---|--------------|--------------|-------------------------------|
|                            |  | Mar 21, 2022 | Mar 23, 2022 | nNV0G79wGQaCmxcgsI0hX8VI2...  |
|                            |  | Mar 21, 2022 | Mar 21, 2022 | 2bbacSDMZYSipFyzF9dafvn7ITH2  |
|                            |  | Mar 21, 2022 | Mar 21, 2022 | 05E5E09nSvPBXenyJFcZ4jqWfga2  |
|                            |  | Mar 21, 2022 | Mar 21, 2022 | uKjGEE5MDEcZBT3GXFrBwiQ1w...  |
|                            |  | Mar 21, 2022 | Mar 21, 2022 | rAJUz2wA8XUfaBuREmJ1wZEL5e... |
|                            |  | Mar 21, 2022 | Mar 21, 2022 | HLhB57vSBod70QHvPt492qvlEj1   |
|                            |  | Mar 20, 2022 | Mar 20, 2022 | xGKximk2uiP3GG66jijyOZ9AyTCr1 |
|                            |  | Mar 16, 2022 | Mar 16, 2022 | QJ3i3mhJVVWLS0DVEgRnoZijlrK2  |
|                            |  | Jan 18, 2022 | Mar 23, 2022 | L1VDYvAdjvWvgWsiQ5DqhCreLEy1  |
| pavlicek.marecek11@gmai... |  | Oct 17, 2021 | Mar 22, 2022 | AxKCMrGftgM1uYueydTRgwqHmv... |

Rows per page: 50 1 - 10 of 10

Obrázek 28 Auth info

## 4. Technologie

V této kapitole bych se chtěl věnovat využitým technologiím v mém projektu. Navážu tím na minulou kapitolu. Využívám zde pojmy jako Javascript (dále jen JS) nebo uživatelské rozhraní (dále jen UI).

### 4.1. React

React je JS knihovna, kterou v roce 2011 vytvořil Jorda Walke a usnadňuje vytváření interaktivních UI ve webových aplikacích. Knihovnu React spravuje společnost Meta (dříve Facebook). Z hlediska MVC architektury je React pouze V neboli view (vrstva pohledu), která zobrazuje uživateli data. Využívá deklarativního způsobu práce s daty, což napomáhá k lepšímu debugování. Tento způsob využívá takzvané komponenty. [1]

#### 4.1.1. Komponenty

Komponenty jsou základním stavebním kamenem Reactu. Jsou to izolované kousky kódu. Fungují podobně jako JS funkce, nezávisle na sobě a vrací HTML. Kladou důraz na znovupoužitelnost kódu, to znamená, že naše komponenta se bude moci vyskytovat na více místech zároveň. [2][3]

### 4.2. Node.js

Node.js je asynchronní JS runtime, který je designovaný pro vytváření škálovatelných síťových aplikací. Všechn kód je spouštěn asynchronně, což povoluje provádění operací nezávisle na jejich pořadí a zpracovává tím pádem více operací zároveň. Právě



díky této funkci vyniká oproti ostatním. Implementuje rozhraní pro webové servery jako jsou souborové systémy a http. Díky tomu umožňuje uživatelům práci se soubory a reakce na HTTP požadavky.[4]

### **4.3. Next.js**

Next.js je framework Reactu pro vytváření takzvaných single-page webových aplikací. Toto napomáhá jak uživatelskému pohledu, tak programátorům. Dokáže sloužit i jako generátor statických stránek. Vytvořil jej společnost Vercel (dříve ZEIT), která poskytuje server-less hosting optimalizovaný pro tento framework. [5]

#### **4.3.1. Automatické rozdělování kódu**

Next.js automaticky rozděluje kód, takže se načte pouze JS a CSS potřebný na dané stránce. Tato akce zrychluje načítání stránek. [5]

#### **4.3.2. SSR – server-side rendering**

Next.js podporuje takzvaný server-side rendering (vykreslování na straně serveru). Komponenty Reactu, o kterých se mluvilo dříve, se načtou na straně serveru. Jakmile uživatel klikne na naši stránku, tak už se nemusí nic dalšího dít a uživateli přijde načítání mnohem rychlejší. Tato funkce je velmi důležitá pro SEO. [5]

#### **4.3.3. HMR – Hot module replacement**

HMR je funkce výhodná pro vývojáře, protože díky ní hned vidí všechny změny, které udělal. Na rozdíl od klasického „live reload“, kde se načítá celá stránka znovu, načítá tato funkce pouze komponenty, které byly změněny. Tím pádem je celé načítání rychlejší a lépe se s stránkou pracuje. [5]

### **4.4. Typescript**

Typescript je open source jazyk vyvinutý společností Microsoft. Je nadstavbou JS, to znamená, že k již funkčnímu jazyku přidává funkce navíc. Základní funkcí se systém typů. Můžete definovat typ proměnné nebo objektu dříve, než se do něj vepíší data. Pomocí typů je zachycení chyb možné již v kódu, zatímco je JS dokáže zachytit pouze po spuštění kódu. [6]

### **4.5. Firebase**

Firebase je platforma pro vytváření webových a mobilních aplikací od společnosti Google. Je to cloudová platforma poskytující takzvaný backend-as-a-service. Dají se v ní nastavit takzvané security rules, díky kterým můžete spravovat autorizaci dle uživatelských účtů. [7]

#### **4.5.1. Firebase Authentication**

Firebase Authentication se soustředí na děláni přihlašovacích systémů jednoduše a příjemně pro obě strany. Podporuje mnoho způsobů přihlášení jako je Google, Twitter, Facebook a další. Vytvořil ji stejný tým, který vytvořil přihlašování přes Google nebo také Chrome Password Manager. Přichází také s knihovnou FirebaseUI, kterou můžete použít pro zkombinování více přihlašovacích způsobů. Funguje jak na webech, tak na mobilních zařízeních. Normálně trvá udělat přihlašování dlouhou dobu, ale s připravenými funkcemi Firebase je máte hotovou během chvíle. [8]

## 4.5.2. Cloud Firestore

Cloud Firestore je flexibilní, škálovatelná NoSQL databáze pro mobilní, webový a serverový vývoj. Collections a v nich documents jsou hlavními částmi struktury dat v této databázi. Nabízí offline podporu pro mobilní a webové aplikace, takže můžete vytvářet responsivní aplikace bez ohledu na latenci sítě. Určitým uživatelům můžete jednoduše zamezit přístup k datům na základě některých informací o nich jako je jejich ID a další. Tato práce s uživateli je nejjednodušší, pokud využíváte Firebase Authentication. Nabízí také integraci ostatních produktů Firebase. [9]

## 4.5.3. Cloud Storage

Cloud Storage je výkonná, jednoduchá a k ceně efektivní služba pro ukládání uživatelských souborů jako jsou fotografie nebo videa. Firebase SDK for Cloud storage společně s Firebase Authentication přidávají jednoduchou a intuitivní kontrolu přístupu. Na základě uživatelské identity přidávají Google zabezpečení k nahrávání a stahování souborů ve vašich aplikacích. Upload nebo download dat z Cloud Storage je automaticky pozastavován, když vaše zařízení ztratí přístup k internetu. [10]

## 4.6. styled-components

Knihovna styled-components je výsledkem přemýšlení o tom, jak by mohlo být CSS u React komponent posunuto na další úroveň. Tato knihovna je velmi příjemná pro vývojáře a skvěle spolupracuje i s Next.js. Tato knihovna generuje unikátní jména tříd pro vaše Komponenty. Je jednoduché upravovat styly, protože je máte připojené na svou komponentu a nemusíte složitě hledat v CSS souborech. Všechno připojení využitých kaskádových stylů za vás udělá automaticky tato knihovna. Na obrázku číslo 29 můžete vidět, jak je jednoduchá samotná implementace stylované komponenty. [11]

```
const StyledCourse = styled.div`
  margin: 5px;
  border: 1px solid #000555;
  border-radius: 10px;
  float: left;
  width: 320px;
  cursor: pointer;
  padding: 0;
  &:hover {
    border: 1px solid #000666;
  }
`;
```

Obrázek 29 Ukázka styled-component

## 4.7. OpenWeatherMap

OpenWeatherMap poskytuje data o momentálním počasí, předpovědi počasí (4, 16, 30 dní), nebo dokonce data o počasí stará až 1 rok. Jsou zde i poskytována i data o rychlosti větru, teplotě nebo dokonce vlhkosti. Data můžete získávat z jakékoliv lokace na světě. Do určitých limitů je celá api zadarmo, poté už si musíte vybrat plán který vám bude sedět. [12]

## 5. Závěr

Byla vytvořena webová aplikace, který splňuje vše, co jsem si stanovil. Pro implementaci byly zvoleny technologie Next.js a Google Cloud Platform.

Web je připraven pro spuštění pro celkem úzkou skupinu lidí, kterou jsou hráči golfu.

Při zpracovávání mého webu jsem narazil na mnoho problémů. Ať už to byly zacyklené dotazy do databáze nebo pouze spolužáci, kteří se to snažili co nejvíce rozbít abych viděl kde mám díry v systému. Většina problémů je nyní již s vynaložením úsilí vyřešená. Takže je aplikace připravena na produkční nasazení.

Pokud by se web uchytil přidaly by se na stránku reklamy, aby se dokázal sám uživit. Také bych chtěl navázat spolupráci s Českou Golfovou Federací, aby mi poskytla potřebná data pro rozšíření. Kdyby byl opravdu velký zájem o tuto aplikaci je v plánu mobilní aplikace.

# Seznam použité literatury

- [1] React - A JavaScript library for building user interfaces. *React* [online]. [cit. 2022-01-22]. Dostupné z: <https://reactjs.org/>
- [2] React Components. *W3schools* [online]. [cit. 2022-01-22]. Dostupné z: [https://www.w3schools.com/react/react\\_components.asp](https://www.w3schools.com/react/react_components.asp)
- [3] Lekce 1 - Úvod do React. *ITNetwork* [online]. [cit. 2022-01-22]. Dostupné z: <https://www.itnetwork.cz/javascript/react/zaklady/uvod-do-react/>
- [4] About Node.js. *Node.js* [online]. [cit. 2022-03-22]. Dostupné z: <https://nodejs.org/en/about/>
- [5] Next.js; what is it and why do we use it? *Clock*. [online]. [2022-01-22]. Dostupné z: <https://www.clock.co.uk/insight/next-js-what-is-it-and-why-do-we-use-it>
- [6] Přehled TypeScriptu. *Microsoft* [online]. [cit. 2022-01-22]. Dostupné z: <https://docs.microsoft.com/cs-cz/learn/modules/typescript-get-started/2-typescript-overview>
- [7] Firebase products. *Firebase* [online]. [cit. 2022-01-22]. Dostupné z: <https://firebase.google.com/products-build>
- [8] Firebase Authentication: Simple, no-cost multi-platform sign-in. *Firebase* [online]. [cit. 2022-03-20]. Dostupné z: <https://firebase.google.com/products/auth>
- [9] Cloud Firestore: Store and sync app data at global scale. *Firebase* [online]. [cit. 2022-03-20]. Dostupné z: <https://firebase.google.com/products/firestore>
- [10] Cloud Storage for Firebase: Store and serve content with ease. *Firebase* [online]. [cit. 2022-03-20]. Dostupné z: <https://firebase.google.com/products/storage>
- [11] Basics - Motivation. *Styled-components* [online]. [cit. 2022-03-22]. Dostupné z: <https://styled-components.com/docs/basics#motivation>
- [12] Weather API. *OpenWeatherMap* [online]. [cit. 2022-03-26]. Dostupné z: <https://openweathermap.org/api>

## Seznam obrázků

|  |    |
|--|----|
| Obrázek 1 Ukázka celé Landing page .....                 | 8  |
| Obrázek 2 Mapa s markery .....                           | 9  |
| Obrázek 3 Seznam hřišť .....                             | 9  |
| Obrázek 4 Seznam po využití textového vyhledávání .....  | 10 |
| Obrázek 5 Seznam po využití filtru oblíbených.....       | 10 |
| Obrázek 6 Ukázka celé detail page .....                  | 11 |
| Obrázek 7 Popis hřiště.....                              | 11 |
| Obrázek 8 Hodnocení na detail page.....                  | 12 |
| Obrázek 9 Průměrné hodnocení na Landing page .....       | 12 |
| Obrázek 10 Počasí.....                                   | 12 |
| Obrázek 11 Pole pro přidání komentáře .....              | 13 |
| Obrázek 12 Komentář.....                                 | 13 |
| Obrázek 13 Přidání fotografie.....                       | 14 |
| Obrázek 14 Fotografie s popiskem v posuvníku .....       | 15 |
| Obrázek 15 Ukázka celé admin page.....                   | 15 |
| Obrázek 16 Navbar před přihlášením.....                  | 16 |
| Obrázek 17 Navbar po přihlášení .....                    | 16 |
| Obrázek 18 Seznam s filtry v mobilní responzi.....       | 17 |
| Obrázek 19 Mapa a navigační menu v mobilní responzi..... | 17 |
| Obrázek 20 Posuvník a komentáře v mobilní responzi ..... | 18 |
| Obrázek 21 Detail s hodnocením v mobilní responzi .....  | 18 |
| Obrázek 22 Architektura aplikace .....                   | 19 |
| Obrázek 23 Collection courses .....                      | 21 |
| Obrázek 24 Collection rating .....                       | 21 |
| Obrázek 25 Collection favourites .....                   | 22 |
| Obrázek 26 Course ID Collections .....                   | 22 |
| Obrázek 27 Architektura storage .....                    | 23 |
| Obrázek 28 Auth info.....                                | 24 |
| Obrázek 29 Ukázka styled-component.....                  | 26 |