

DELTA – Střední škola informatiky a ekonomie, s.r.o.

Ke Kamenci 151, PARDUBICE

MATURITNÍ PROJEKT

Webová aplikace pro lezce

Příjmení, jméno: Zamazal Štěpán

Třída: 4.A

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

Školní rok: 2021/2022

Konzultant: *Bc. Vladka Janů*

Zadání maturitního projektu z informatických předmětů

Jméno a příjmení: *Štěpán Zamazal*
Školní rok: *2021/2022*
Třída: *4.A*
Obor: *Informační technologie 18-20-M/01*

Téma práce: *Horolezecký průvodce*

Vedoucí práce: *Bc. Vlad'ka Janů*

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

Cílem projektu je vytvořit webovou aplikaci pro lezce. Aplikace by byla takovým lezeckým průvodcem. Lezec přijde na stránku, uvidí tam dané lezecké oblasti a informace k nim (polohu, obrázek skály, počet cest, názvy cest, obtížnostní hodnocení cesty, atd...). Součástí toho bude přidávání oblastí nebo cest. Projekt by měl úspěch, kdyby se dostal k co nejvíce lidem z lezecké komunity a ti by pak měli mít možnost přidávat své lokální oblasti právě do této aplikace. S tím tedy souvisí, že zde bude i správa uživatelů. Aplikace se bude dávat pouze prohlížet, ale pro upravení se bude muset uživatel přihlásit a podstatné věci budou moci měnit jen ověřeni uživatelé.

Základní technologie:

- React
- NextJS
- Typescript
- Firebase
- API Mapy.cz

Stručný časový harmonogram (s daty a konkretizovanými úkoly):

- Studování technologií, které budou v aplikaci využity. Nafocení lokálních skalních oblastí, které budou v aplikaci. Oslovení kompetentnějších lezců z daných oblastí pro zjištění informací k cestám. (září)
- Vývoj první verze webové aplikace, práce ve frameworku NextJS, zhotovení Firebase databáze s potřebnými daty. Práce s API Mapy.cz. (říjen-listopad)
- Zhotovení plně funkčního webu. Správné fungování přihlašování, možnost upravování pro adminy. (prosinec)
- Dokončování a zdokonalování. Testování webu. (leden-únor)
- Práce na dokumentaci k tomuto projektu. (leden-březen)

Prohlášení

Prohlašuji, že jsem maturitní projekt vypracoval(a) samostatně, výhradně s použitím uvedené literatury.

V Pardubicích 31.3.2022

(vlastnoruční podpis)

Poděkování

Největší poděkování bych chtěl věnovat Bc. Vladěce Janů. Děkuji za odborné vedení projektu, konzultování při vývoji aplikace a kontrolu správnosti napsané dokumentace. Chtěl bych také poděkovat mojí sestře Lucii Zamazalové, která mi vytvořila logo aplikace. Obrovské poděkování také patří Vojtu Markovi, který se mnou konzultoval, co by v aplikaci mohlo být a pomáhal mi při focení oblastí a získávání informací o nich.

Anotace

Cílem projektu je vytvořit přehlednou webovou aplikaci pro lezce. Uživatelé této aplikace budou mít možnost prohlížet jednotlivé oblasti, bloky a cesty. Další hlavní částí aplikace bude přidávání dat do databáze. Tuto možnost budou mít registrovaní a ověření uživatelé. Následně se tato data zobrazí v systému.

Projekt míří na skupinu lidí, kteří se věnují skalnímu lezení a hledají informace o cestách v různých oblastech.

Klíčová slova

Webová aplikace, Next.js, React, Firebase, Mapy.cz, Skalní lezení

Annotation

Main goal of this project is to create clearly arranged web application for climbers. Users of this application will have an option to browse individual areas, blocks and routes. Another main part of the application will be an option to add data to database. This option will be accessible for users that are registered and verified. Afterwards these data will be shown in the system.

This project aims at a group of people that are interested in rock climbing and want to find information about routes in specific areas.

Key words

Web application, Next.js, React, Firebase, Mapy.cz, Rock climbing

Obsah

1. Úvod.....	8
1.1. Cílová skupina	8
1.2. Terminologie	9
1.2.1. Oblast	9
1.2.2. Blok	9
1.2.3. Cesta	9
2. Technologie.....	9
2.1. Node.js	9
2.2. Next.js	10
2.3. React	10
2.4. Firebase.....	10
2.4.1. Cloud Firestore	11
2.4.2. Firebase Authentication.....	12
2.4.3. Cloud Storage	12
2.5. React Mapy.cz	12
2.6. TypeScript	12
2.7. Styled Components.....	12
2.8. Formik	13
2.9. Reactjs-popup	13
3. Architektura	14
3.1. Backend	15
3.1.1. Firestore databáze.....	15
3.1.2. Správa uživatelů	18
3.1.3. Uložiště fotografií	19
3.2. Node server.....	21
3.2.1. Next.js aplikace	21
3.3. Klient	24
4. Webová aplikace.....	25
4.1. Přihlášení a registrace.....	25

4.2.	Úvodní stránka.....	26
4.3.	Oblasti.....	27
4.3.1.	Oblasti	27
4.3.2.	Bloky	28
4.3.3.	Cesty.....	29
4.4.	Mapa	30
4.5.	Profil	30
4.5.1.	Uživatelské informace.....	31
4.5.2.	Změna údajů	32
4.5.3.	Ověření emailu a přidávání informací.....	32
4.5.4.	Admin.....	33
4.5.5.	Responzivita webu	34
5.	Budoucnost aplikace	35
5.1.	Co v aplikaci chybí?	35
5.2.	Co bude dál?	35
6.	Závěr	36
7.	Seznam použité literatury	37
8.	Seznam obrázků	38

1. Úvod

Každý lezec za svojí lezeckou kariéru narazí na jeden specifický problém. Tím problémem jsou průvodci skalních oblastí. Problémů s těmito průvodci je hned několik. Jedním z nich je, že tyto průvodci se občas nedají sehnat. Jsou pouze v tisknuté podobě (v podobě knížky) a jen málokdo tyto knížky vlastní. Další z problémů je aktuálnost daných průvodců. V oblastech se přidávají nové cesty, ale tyto cesty nejsou v podstatě nikde k dohledání. V neposlední řadě tu máme problém s internetovými průvodci. Ty jsou z valné většiny zastaralé a nepřilíš graficky zvládnuté.

Za tímto účelem byla vytvořena aplikace Climberries. V první řadě je tu řešení tisknuté podoby průvodců. V případě, že lezec, který vlastní papírovou podobu, bude chtít informace v ní sdílet ostatním lezcům, v této aplikaci má tu možnost. Lze zde přidat danou oblast, bloky, které se v ní nachází a k nim provázané cesty. Ostatní lezci pak budou mít možnost oblast prohlížet. Další řešený problém je aktuálnost. Vytvoří-li se nové cesty, autor má možnost je zaznamenat. Jednoduše si najde danou oblast (v případě, že aplikace ji už obsahuje) a přidá nově vytvořené cesty. Dále je tu problém se zastaralými weby. Aplikace pracuje s novými technologiemi, je rychlá a přehledná.

1.1. Cílová skupina

Jak už si jistě lze v úvodním odstavci odvodit, cílovou skupinou této aplikace jsou lidé, kteří se věnují skalnímu lezení.

Jedním typem uživatelů tak mohou být lezci nováčci, kteří nemají přesné informace o daných oblastech. Aplikace jim pomůže vyhledat si o oblasti informace a předem si určit, zda má cenu do dané oblasti jezdit.

Druhý typ uživatelů jsou zkušení lezci. Ti si v aplikaci mohou vyhledávat například nejobtížnější cesty, které se v oblasti nachází.

Poslední typ uživatelů budou využívat spíše interaktivní prostředí pro přidávání informací. Jde spíše o lezce, kteří už mají něco za sebou. Mají tak informace o oblastech, které chtějí předat dál. Díky této aplikaci tak doplní informace pro ostatní. Je to také jedna z možností, kde lze zaznamenávat kompletně nové cesty.

1.2. Terminologie

1.2.1. Oblast

Oblastí se rozumí sdružení několika bloků a cest, které jsou zahrnuty v jeden ucelený areál. Lezecká oblast pak nese svůj název (např. Mladkov, Zemská brána, atd...).

1.2.2. Blok

Bloky jsou součástí oblastí. Tento popis slouží k rozčlenění oblasti na menší kusy. Typicky se tak označuje jeden kus kamene či skály, který se v oblasti nachází. Na bloku pak najdeme cesty, které se dají lézt. Bloky se identifikují také pomocí názvů.

1.2.3. Cesta

Cesta je poslední část oblasti. Nachází se na bloku. Má své jméno, autora, obtížnostní hodnocení či způsob, jakým se leze. Cesty jsou hlavním důvodem, proč lezci navštěvují oblasti. Informace k nim jsou potřebné k porovnání obtížností s jinými cestami nebo zpětnému změnění obtížnosti. Hlavní součástí informací je i způsob přezení. Tento popis může pomoci lezci k vylezení cesty.

2. Technologie

Následující část textu se věnuje technologiím, které jsou v aplikaci využity. Webová aplikace je napsána ve frameworku Next.js, který je postaven na javascriptové knihovně React. Backendem aplikace je platforma Firebase.

2.1. Node.js

Node.js je vývojové prostředí vytvořené pro jazyk JavaScript. Slouží k vytváření rychlých a škálovatelných aplikací. Využívá architekturu zvanou „Smyčka událostí s jedním vláknem“, díky které dokáže zpracovat více klientů naráz.

Node.js má omezený počet vláken pro obsluhu požadavků. Ve chvíli, kdy přijde požadavek na server, je umístěn do fronty. Rozhraní API knihovny Node.js jsou asynchronní, navzájem se tedy neblokují. Tato knihovna je postavena na JavaScript Engine V8 a je velmi rychlá ve chvíli, kdy se kód provádí. Node.js se tak typicky využívá u realtime aplikací, kde je server schopný obsloužit více požadavků najednou. [1]

2.2. Next.js

Next.js je framework, který je postavený na vývojovém prostředí Node.js. Tento framework rozšiřuje známou JavaScriptovou knihovnu React. Společně tak dávají dohromady prostředí, které je ideální k vývoji webové aplikace. Next.js podporuje server side rendering. Tato metoda vygeneruje HTML kód na straně serveru, který po vygenerování odešle uživateli do prohlížeče. Až poté se v prohlížeči začnou „rozpohybovat“ javascriptové prvky, které uživateli zobrazí zbylý obsah stránky.

Další z funkcí, které Next.js obsahuje, je Fast Refresh nebo také Hot Module Replacement. Tato funkce slouží k okamžitému zobrazení změněného kódu po uložení.

Výhodou Next.js je také využívání kódu, který je zrovna potřeba. Tento framework rozpozná, který JavaScript nebo CSS kód je na stránce potřeba a načte pouze ten. Tím je načítání stránek rychlejší a není potřeba stahovat velké množství dat v případě otevření webu. Tuto funkci jistě ocení uživatelé s pomalým připojením.

Next.js dále obsahuje Next Router. Tento router umožňuje navigaci v adresářích na webu. Jednoduše lze díky němu přesměřovat mezi stránkami a vytvářet dynamické stránky. Na těchto stránkách tak lze zobrazit obsah, který je specifický například pro daného uživatele. Router dokáže do adresy zapsat potřebná data a zároveň je dokáže i vytáhnout. Například lze do adresy poslat ID, které se na dynamické stránce získá. ID se pak využije k zobrazení specifických dat. [2][3]

2.3. React

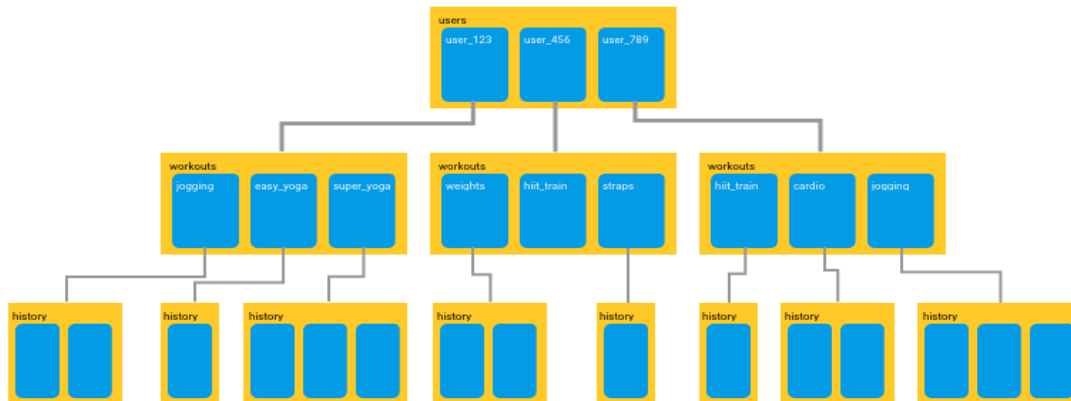
React je JavaScriptová knihovna, jejímž účelem je tvorba uživatelského rozhraní. Hlavní předností této knihovny je práce s rychle měnícími se daty. Základním stavebním kamenem React aplikací jsou komponenty. Ty nám umožňují rozdělit kód do více částí. Ten je poté přehlednější. Další obrovskou výhodou je opětovné využívání již napsaných komponent. Vývojář si tak z části ulehčuje práci, protože může využívat komponenty, které už má napsané. Tyto komponenty mají také zapouzdřenou funkcionalitu. Ve většině případů se React nepoužívá samostatně, je to pouze část receptu na funkční webovou aplikaci. V tomto projektu je například kombinován s frameworkem Next.js. [4]

2.4. Firebase

Firebase je platforma od firmy Google. Tato platforma je backend as a service. Umožňuje vývojáři využívat produkty, které Firebase obsahuje. Je to kompletní serverless řešení backendu, tudíž vývojář nemusí řešit ostatní věci spojené s backendem. Firebase obsahuje hned několik služeb jako je např.: Cloud Firestore, Firebase Authentication, Cloud Storage.

2.4.1. Cloud Firestore

Firestore je dokumentová NOSQL databáze. Je rychlá a škálovatelná. Struktura této databáze se skládá z kolekcí, ve kterých jsou dokumenty. Kolekce má vlastní ID a může obsahovat více dokumentů. Dokument má také své ID a obsahuje data ve formátu JSON. Do těchto JSON objektů lze ukládat potřebná data. Firestore nabízí ukládat data v určitých datových typech jako např. string, number, map, timestamp atd.



Obrázek 1 - Struktura databáze Firestore [11]

Data z databáze Firestore lze získat pomocí Firebase SDK, který se implementuje přímo do klientské aplikace. Výhodou této implementace je to, že není nutné programovat vlastní API, díky kterému by se data získávala.

Zabezpečení ve Cloud Firestore funguje pomocí Firebase Security Rules. Pomocí vlastních pravidel lze nastavit jednotlivé přístupy k datům ve vaší databázi. Ukázku pravidel můžeme vidět na obrázku číslo 2.

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read : if true;
      allow write: if request.auth != null;
    }
  }
}
```

Obrázek 2 - Ukázka Firebase Security Rules [zdroj: autor]

2.4.2. Firebase Authentication

Firebase Authentication zprostředkovává registraci a přihlašování uživatelů. Tyto akce probíhají přes poskytovatele, jako je například Facebook, Google nebo Twitter. Další možností je přihlášení pomocí emailu a hesla. Také umožňuje ověřování emailu, změnu hesla nebo emailu. Tento produkt opět využívá SDK, díky kterému lze tyto akce dělat přímo v aplikaci. Každý uživatel má své UID.

2.4.3. Cloud Storage

Cloud Storage slouží k rychlému a jednoduchému ukládání souborů, jako jsou například fotografie či videa. Tyto soubory lze strukturovat do složek. Každý soubor má v Cloud Storage svá metadata. V těchto datech je uvedena lokace dat ve Storage. Důležitou částí metadat je Access Token. Díky tomuto tokenu lze k souborům přistupovat. U Cloud Storage lze opět využívat Firebase Security Rules. [5]

2.5. React Mapy.cz

React Mapy.cz je balíček využívající API od Mapy.cz. Tento balíček obsahuje již předpřipravené komponenty pro React k zobrazení mapy a jejích prvků. Díky těmto prvkům lze mapu nastavit tak, aby vyhovovala potřebám aplikace. Je zde například možnost markerů, u kterých lze po kliknutí zavolat již předpřipravenou metodu onclick. Tato metoda slouží k zobrazení karty s daty, které se markeru předají. Výhodou využívání API Mapy.cz je, že poskytované mapy jsou zcela zdarma. [6]

2.6. TypeScript

TypeScript je open-source nadstavbou pro skriptovací jazyk JavaScript. Součástí TypeScriptu je transpiler. Tento kompilér převádí TypeScript kód do kódu JavaScript. Základními datovými typy tohoto jazyku jsou: boolean, number, string, any. Díky využívání TypeScriptu vývojář zamezuje možným chybám v kódu. TypeScript kód je pak mnohem spolehlivější, lépe se debuguje a předem nás upozorňuje na možné chyby. [7]

2.7. Styled Components

Styled Components je knihovna postavená pro React a React Native. Hlavním principem je to, že si lze vytvářet vlastní předstylované komponenty. Tato knihovna využívá JavaScript a CSS.

Styled komponenty generují unikátní název třídy pro styly. Díky tomu nevznikají duplikátní názvy nebo nesrovnalosti. Další výhodou je přehlednost těchto stylů. Styly jsou připsané ke každé komponentě zvlášť. Vývojář se tak jednoduše zorientuje v tom, kterou část vzhledu

přepsáním či smazáním ovlivní. S tím souvisí, že pokud je komponenta smazána, všechny styly se smažou s ní.

Styled komponenty generují na stránku ty komponenty, které jsou na stránce potřeba. Do stránky vkládají pouze styly a nic jiného. Díky tomu uživatelé načítají pouze kód, který využívají zrovna na stránce. To celou aplikaci zrychluje. Příklad napsané styled komponenty je zobrazen na obrázku číslo 3. [8]

```
const Nav = styled.nav`
  height: 125px;
  width: 100%;
  background-color: #000;
  color: #fff;
  display: flex;
  justify-content: space-between;
  align-items: center;
  @media ${device.laptop} {
    height: 200px;
    flex-direction: column;
    justify-content: space-around;
  }
  @media ${device.tablet} {
    height: 150px;
  }
`
```

Obrázek 3 - Ukázka styled komponenty [zdroj: autor]

2.8. Formik

Formik je open source knihovna pro React a React Native. Tato knihovna je zcela zdarma. Formik řeší klíčové problémy při práci s formuláři. Do těchto problémů lze zahrnout kontrolování a manipulaci se stavem formuláře, práci s ověřením správnosti zadaných uživatelských dat, zobrazování chybových zpráv a způsob, jakým se formulář zpracuje.

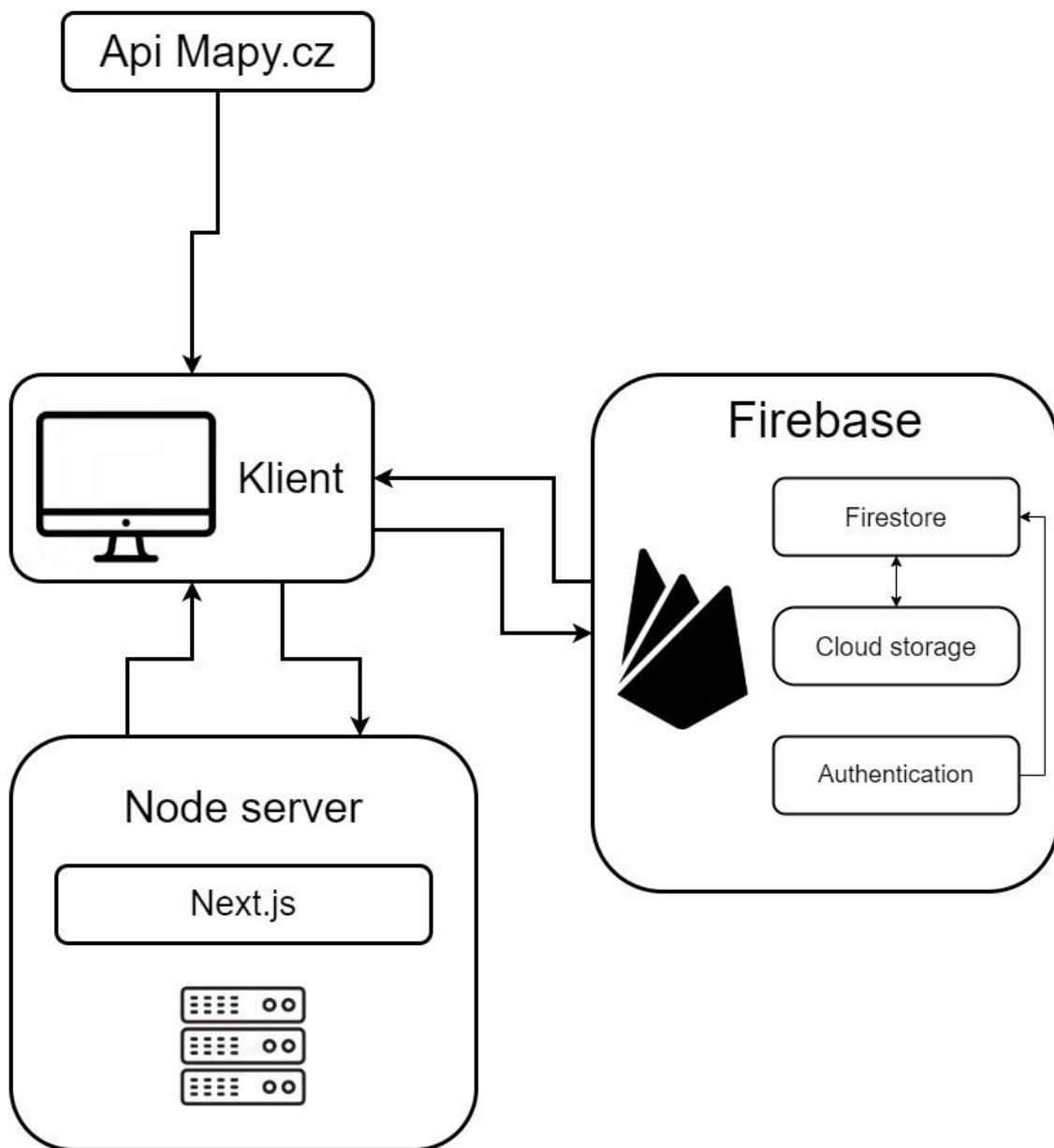
Pro validaci Formik formulářů lze využít knihovny Yup. Ta poskytuje funkce pro validaci dat, které uživatel ve formuláři zadává. Lze tak definovat akceptovatelnou délku řetězce či formát zadaného textu. [9]

2.9. Reactjs-popup

Reactjs-popup je balíček pro React aplikace. Obsahuje předpřipravené komponenty, které aplikace může využívat. Z tohoto balíčku tak lze využít například modal, tooltip nebo menu. Součástí komponenty je takzvaný trigger. To může být například tlačítko. V případě, že uživatel na tlačítko klikne, tak se vygeneruje daná komponenta. [10]

3. Architektura

Architekturu aplikace tvoří tři základní prvky. Webová aplikace (Next.js), Node server a backend (Firebase). Na obrázku číslo 4 lze vidět znázornění architektury celé aplikace. Součástí obrázku jsou všechny prvky včetně jednotlivých propojení.



Obrázek 4 - Architektura aplikace [zdroj: autor]

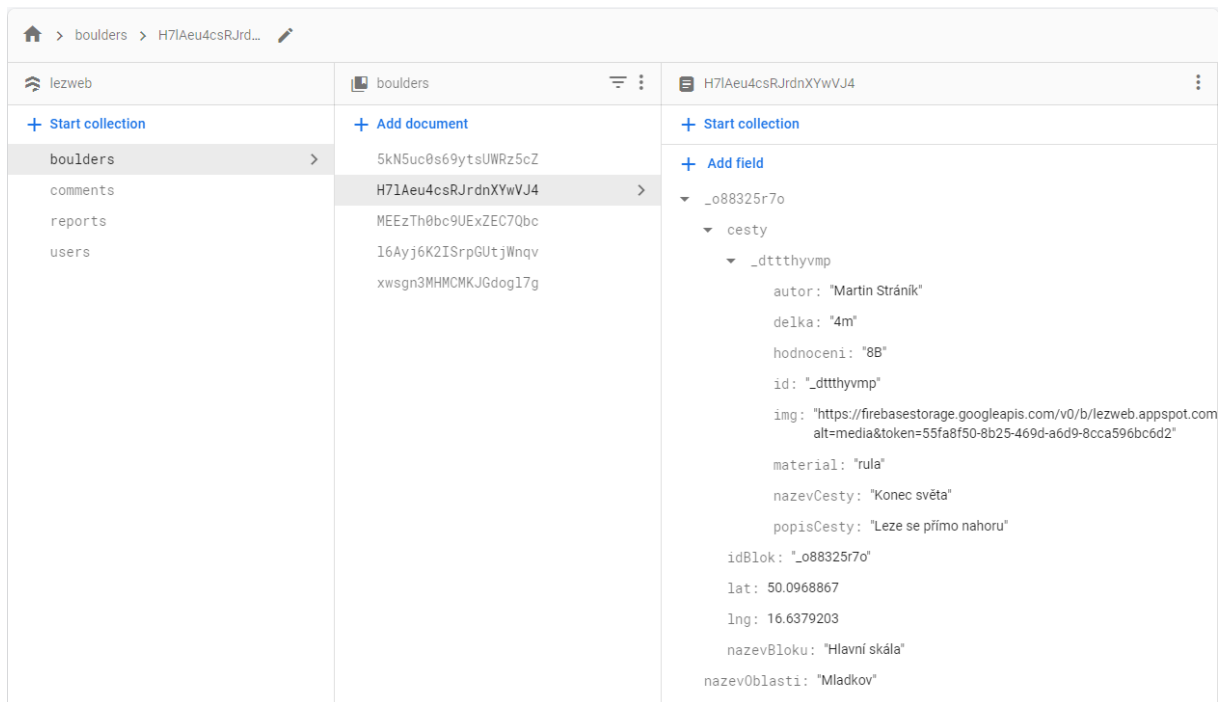
Z obrázku lze vydedukovat, že architektura aplikace je poměrně jednoduchá. Klient napřímo komunikuje s Firebase, která slouží jako backend aplikace. Komunikace probíhá přes Firebase SDK. Na Node serveru běží samotná Next.js aplikace, která se vykresluje u klienta. Posledním prvek architektury je Api Mapy.cz, které slouží k vykreslení mapy. Ta se vykresluje u klienta podle předaných atributů mapě.

3.1. Backend

Jak už je v přechozí části zmíněné, roli backendu v této aplikaci plní Firebase. Ta je napojena na klienta přes webové SDK. Komunikace tak probíhá přímo z klienta. Není zde žádný server, který by komunikaci s databází zprostředkoval. Jedná se tak o částečnou serverless aplikaci. Tím se maximalizuje rychlost a datový tok.

3.1.1. Firestore databáze

Databází této aplikace je Firestore. V ní jsou ukládány informace potřebné v aplikaci. Databáze se strukturalizuje do kolekcí, ve které jsou uloženy dokumenty. Firestore také nabízí možnost subkolekcí. Ty by v případě návaznosti oblastí, bloků a cest dávaly smysl. Pro tuto aplikaci jsou však nepoužitelné, protože Firebase nenabízí listování v subkolekcích. Lze získat pouze jednu. To by značně komplikovalo získávání dat. Tudíž struktura databáze byla zvolena na bázi objektů. To lze vidět například na obrázku číslo 5, kde je struktura databáze zobrazena.



Obrázek 5 - Ukázka struktury kolekce boulders [zdroj: autor]

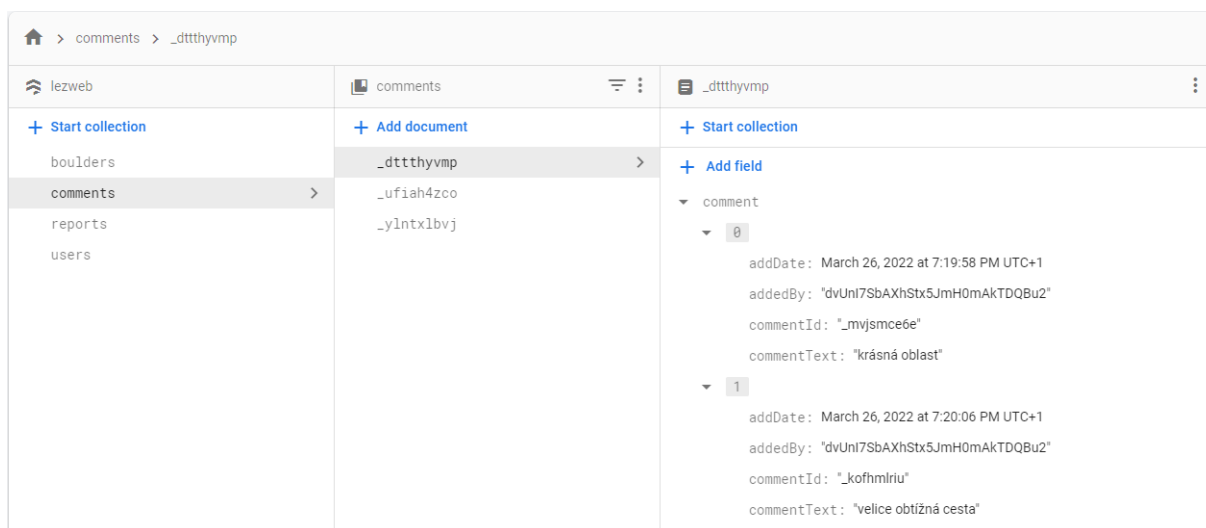
Kolekce boulders obsahuje oblasti a informace k nim. Oblasti jsou uloženy v kolekci boulders proto, že pokud stránka vyžaduje veškeré oblasti, které jsou přidány, stačí jeden dotaz na databázi a aplikace k nim má přístup. Není tak zapotřebí posílat více requestů.

Dokumenty reprezentují jednotlivé oblasti. Název dokumentu nese UID, které je vygenerováno Firebase při vytvoření dokumentu. Díky tomu má každá oblast svůj unikátní identifikátor, který je využit u dynamických stránek. Při vytváření dokumentu je uložen i název oblasti.

Bloky jsou reprezentovány objektem. Název objektu opět nese UID, které je však vygenerováno algoritmem. Firebase nenabízí generování jejich vlastního UID pro objekty. Díky tomu jedna oblast může obsahovat více bloků. Bloky obsahují informace, které jsou vyžadovány při přidávání.

Součástí bloku je také objekt nesoucí název cesty. V tom jsou uchovávány jednotlivé objekty cest. Objekt cesty má opět své UID. Jsou zde také uvedené informace, které jsou u cesty vyžadovány. Jak si lze všimnout, objekt cesty obsahuje položku `img`. V ní se nachází odkaz na fotku, která je v oblasti přidána. Tento link odkazuje na umístění v Cloud Storage, která je v projektu taky využívána.

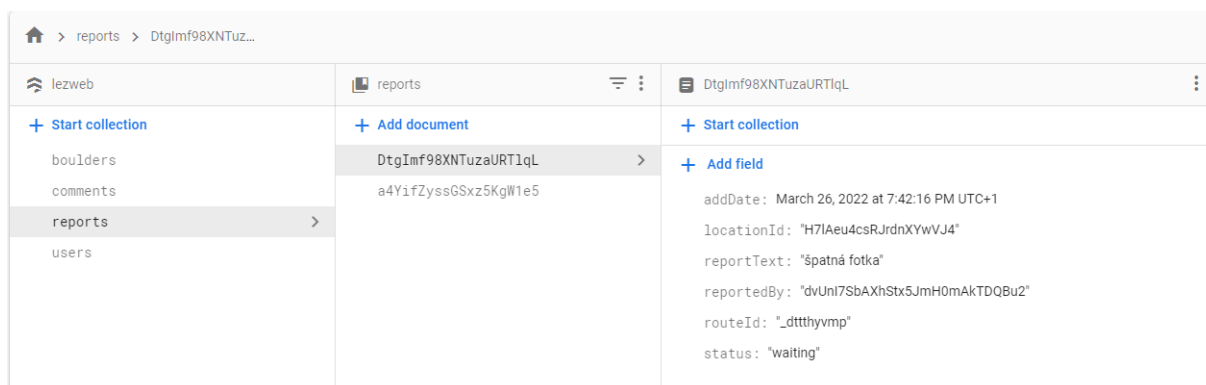
Další kolekci v databázi je kolekce s komentáři. Strukturu komentářů si lze prohlédnout na obrázku číslo 6.



Obrázek 6 - Ukázka struktury kolekce comments [zdroj: autor]

Dokumenty, které jsou uloženy v kolekci, mají název podle toho, ke které cestě byly přidány. Název je UID cesty. To si lze ověřit na obrázku číslo 5 na předchozí straně. Tyto komentáře jsou tak přiřazeny k cestě Konec světa. Dokument obsahuje pole `comment`. V tomto poli jsou na indexech od 0 přidány objekty jednotlivých komentářů. Komentáře obsahují informace, které jsou potřebné při zobrazení v aplikaci. Díky těmto datům je u komentářů zobrazen komentující, čas přidání a text komentáře.

Dále je zde kolekce reportů. Ty se dají přidávat také u cest. Kolekce obsahuje dokumenty, u kterých je vygenerováno UID. Jak lze vidět na obrázku číslo 7, každý dokument reprezentuje



Obrázek 7 - Ukázka struktury kolekce reports [zdroj: autor]

jednotlivé reporty.

Dokument obsahuje jednotlivé položky, které jsou důležité pro adminy při opravování chyby, kterou uživatel v reportu popíše. Způsob fungování reportů je podrobněji popsán v podkapitole číslo 4.5.2. s názvem Admin na straně číslo 33.

Poslední kolekci je kolekce users. Popis této kolekce lze najít v následující kapitole Správa uživatelů. Důvodem je to, že tato kolekce souvisí s registrováním uživatelů.

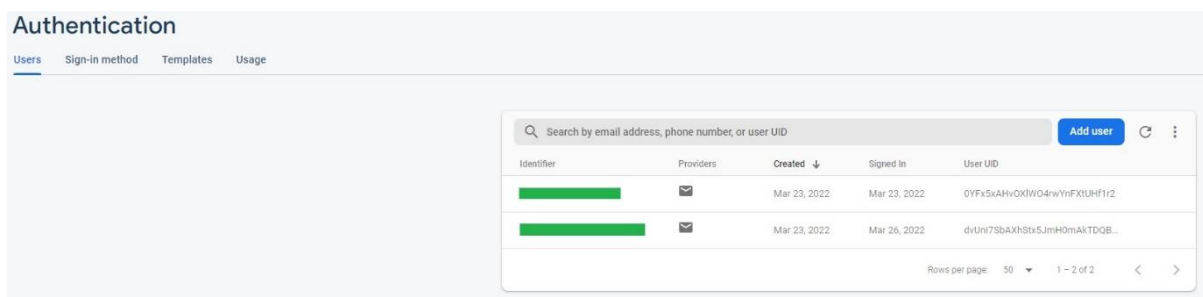
K zabezpečení firestore databáze Firebase poskytuje security rules. V nich lze například nastavit práva na čtení, zapisování či mazání. V této aplikaci jsou pravidla nastavená tak, aby každý uživatel mohl přečíst data i v případě, že není přihlášený. Jediná kolekce, která pro čtení vyžaduje přihlášení, je kolekce users. Složka users má zakázané mazání dat. Pro zapisování do jakékoliv kolekce je podle pravidel vyžadováno přihlášení a ověření emailové adresy. Nastavení pravidel lze vidět na obrázku číslo 8.



Obrázek 8 - Ukázka nastavení Firebase Security Rules [zdroj: autor]

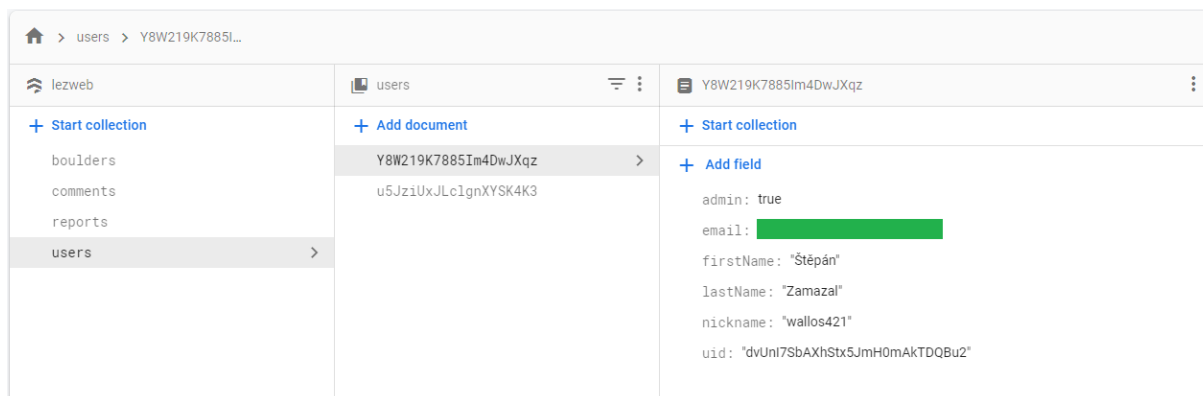
3.1.2. Správa uživatelů

Správa uživatelů je opět řešena přes Firebase. Ta poskytuje službu Firebase Authentication, která řeší veškerou registraci a přihlašování uživatelů. Pro využití této služby je zapotřebí si definovat takzvané providery, kteří zprostředkovávají způsob přihlášení. V tomto projektu je využíván provider Email/Password. To znamená, že uživatelé k registraci nevyužívají jiných účtů (jako je tomu například u providera Google), ale potřebují k registraci zadat email a heslo, které se po registraci uloží do Authentication Users. Účet je tak založený přímo ve Firebase. Zmíněná složka Users, která je zobrazena na obrázku číslo 9, tak obsahuje informace o uživateli.



Obrázek 9 - Složka Users [zdroj: autor]

Zároveň zde však není možnost ukládat i jiné informace, které by si vývojář přál. Je zde pouze účet s emailem, heslem a uživatelským ID. Proto je to v této aplikaci řešeno přes kolekci Users, která je uložena ve Firestore databázi. Tato kolekce obsahuje ostatní informace o uživateli, které zadávají při registraci. Jak lze vidět na obrázku číslo 10, tato kolekce obsahuje pouze dokumenty, které přímo reprezentují dané uživatele.



Obrázek 10 - Ukázka struktury kolekce users [zdroj: autor]

Dokument má opět vygenerované UID od Firebase, zároveň je však v kolekci uloženo UID, které bylo vygenerováno Firebase Authentication a pod kterým je uloženo ve složce Users. Toto UID je klíčem, které propojuje každý účet s jeho kolekcí. Propojení aplikace využívá ve chvílích, kdy je potřeba získávat údaje, které nejsou k získání od Firebase Authentication.

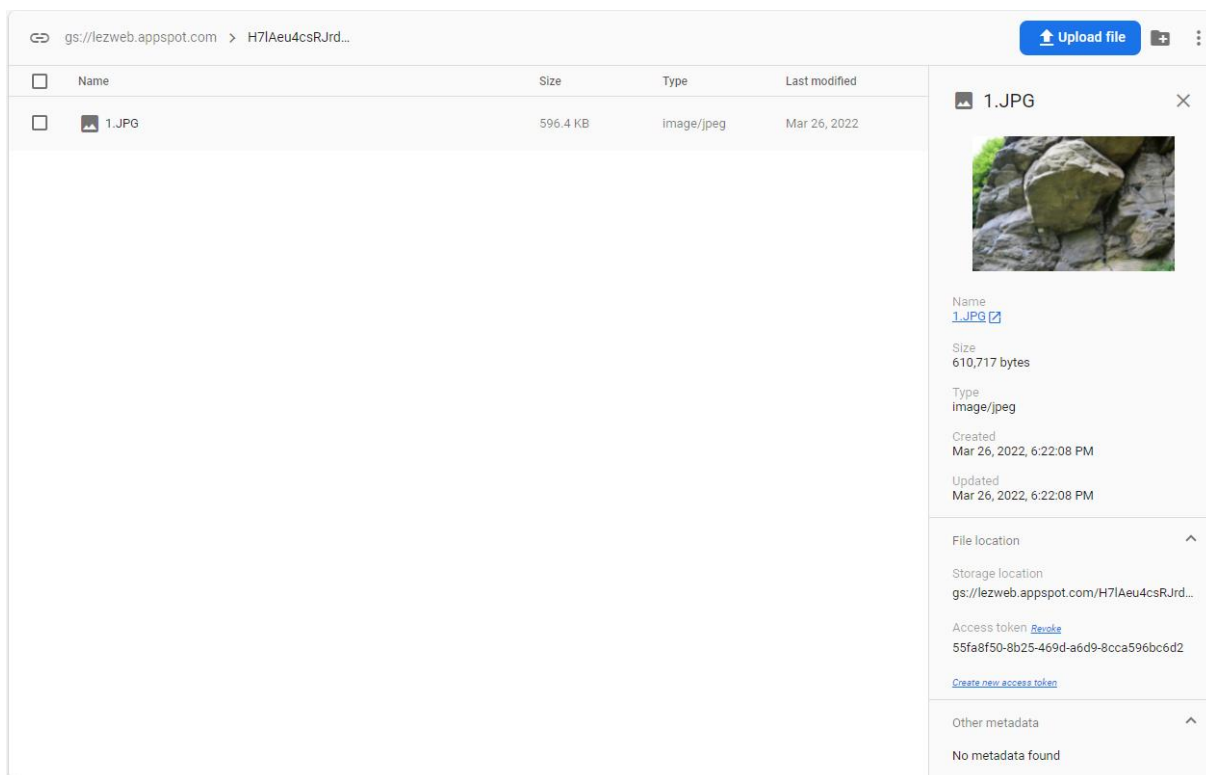
Jak si lze jistě všimnout na obrázku číslo 10, každý dokument v kolekci users obsahuje položku admin. Ta je buďto true anebo false. Je to informace o tom, zda má uživatel Adminská

oprávnění či nikoliv. Tato informace je tak využívána při kontrole, zda uživatel má možnost přeměrování na admin stránku. V případě, že je v databázi uvedeno false, uživatel je přeměrován a na stránku se nedostane.

Poslední částí správy uživatelů je ověření emailu. Toto ověření by mělo zaručit, že za počítačem sedí opravdový člověk, a že emailová adresa je pravá. Tuto možnost opět poskytuje Firebase Authentication. Jednoduše se využije funkce, která se zavolá například při kliknutí na tlačítko. Následně je do emailové schránky uživateli zaslán email pro ověření. Po kliknutí na odkaz v emailu je adresa ověřena. Účtu je tak změněn atribut verified na true. Tento atribut je součástí účtu ve Firebase Authentication. K některým akcím aplikace vyžaduje toto ověření. To je lépe popsáno v kapitole Webová aplikace.

3.1.3. Uložiště fotografií

Pro ukládání fotografií aplikace využívá Cloud Storage. Tato služba poskytuje možnost uložení souborů. Soubory lze ukládat do složek. Každý soubor obsahuje metadata. Ty obsahují údaje o souboru včetně access tokenu. Všechny údaje o souboru lze vidět na obrázku číslo 11.



Obrázek 11 - Ukázka uložené fotky [zdroj: autor]

Díky tomuto access tokenu lze ve Firestore zobrazit fotografii cesty. To je již popsáno v kapitole Firestore databáze. Fotografie se ukládají do složek podle toho, do které oblasti spadají. Název složky je tak stejný jako UID oblasti, jak lze vidět na obrázku číslo 12.

<input type="checkbox"/>	Name	Size	Type	Last modified
<input type="checkbox"/>	H7IAeu4csRJrdnXYwVJ4/	-	Folder	-
<input type="checkbox"/>	MEEzTh0bc9UEXZEC7Qbc/	-	Folder	-
<input type="checkbox"/>	unknown/	-	Folder	-

Obrázek 12 - Složky v Cloud Storage [zdroj: autor]

Je zde také složka unknown. V té je uložena fotografie pro případ, že v objektu cesty položka img obsahuje prázdný řetězec. Tato situace může nastat, protože přidání fotografie k cestě není povinné. Fotografie se tak zobrazuje na místech, kde by v normálních případech byla fotografie cesty.

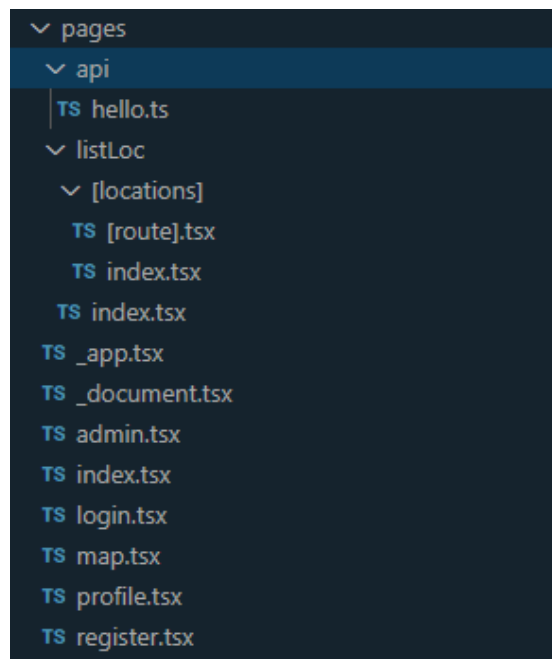
3.2. Node server

Na Node serveru běží samotná Next.js aplikace. Zde jsou také nainstalované balíčky, které jsou v aplikaci využívány. Balíčky, které má server nainstalovat, nalezne v souboru package.json. Hlavní součástí struktury samotné aplikace jsou složky pages a components.

3.2.1. Next.js aplikace

Aplikace Next.js je nadstavbou pro React. V aplikaci jsou tak využívány prvky jak Reactu, tak i Next.js. Next.js aplikace má danou svou strukturu souborů. Díky této struktuře se vývojář lépe orientuje v samotném vytváření aplikace.

Hlavní a důležitou částí je složka pages. Tato složka obsahuje všechny stránky, na které se lze v aplikaci dostat (až na stránky, které mají podtržítka před názvem). Na obrázku číslo 13 můžeme vidět strukturu stránek této aplikace.



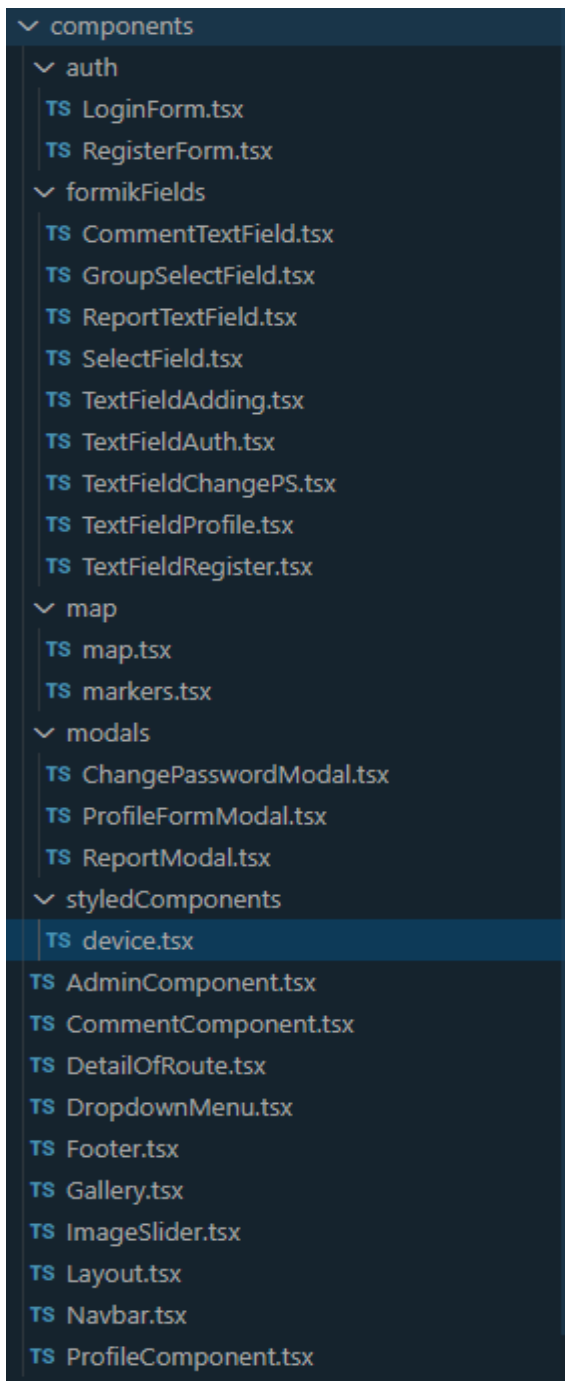
Obrázek 13 - Ukázka struktury složky pages [zdroj: autor]

Název souboru tak reprezentuje i cestu, jak se lze v adresovém řádku stránky pohybovat. Specifickým využitím v tomto projektu jsou dynamické stránky. Konkrétně se v projektu využívá dynamické stránky, která je zanořena v jiné. Do adresového řádku stránky se zadá ID oblasti. To se na dynamické stránce získá pomocí Next routeru a díky tomuto ID se tak zobrazí pouze oblasti, které jsou zapsány k dané oblasti. Podobně to pak funguje i s dynamickou stránkou cest, jen se zde do adresáře zadá ID cesty. Příklad zanoření adresáře lze vidět na obrázku číslo 14.

climberries.vercel.app/listLoc/H7IAeu4csRJrdnXYwVJ4/_dtthtyvmp

Obrázek 14 - Ukázka adresového řádku [zdroj: autor]

Další důležitou složkou v Next.js aplikaci je složka components. V té se nachází jednotlivé komponenty aplikace. Komponenty jsou alfa a omega celé React aplikace. Vývojář díky nim může kód rozdělit do více souborů. Soubory ve složce pages tak nejsou dlouhé přes mnoho řádků. Komponenty se dají také znovu využívat. Komponenty tohoto projektu můžeme vidět na obrázku číslo 15.



Obrázek 15 - Ukázka struktury složky components [zdroj: autor]

Komponenty jsou členěné logicky. Máme tu například obecné komponenty, které nejsou členěny do žádné složky. Jsou tu však komponenty, které spolu souvisí, a tak dává smysl, je uložit do jednoho adresáře.

Složka auth obsahuje veškeré komponenty, které souvisí s registrací a přihlášením uživatele. Zároveň tyto komponenty využívají komponenty ze složky formikFields. V té se nachází komponenty pro Formik formuláře. Konkrétně se jedná o jednotlivé inputy a labely k nim. Jsou zde také selecty.

Specifickou komponentou pro tento projekt je komponenta map. Ta má také svou vlastní složku a je v ní uložena jak samotná mapa, tak i markery, které se na mapě zobrazují. Mapa využívá balíček react-mapycz. To je balíček, který využívá veřejného Api Mapy.cz. Díky němu je v projektu možnost využití těchto map. Důvodem využití těchto konkrétních map je to, že jsou to české mapy. Aplikace má primárně obsahovat české lezecké oblasti. Dává tedy smysl využití těchto map vzhledem k tomu, že terén i samotné skály jsou na nich lépe zobrazeny než například u Google map. S využitím tohoto balíčku se samozřejmě pojí i určité problémy. Reactí balíček neobsahuje všechny funkce, které Api Mapy.cz nabízí. Hlavním problémem je, že po kliknutí na mapu nelze zavolat funkci, která by vrátila zeměpisnou šířku a délku místa. To je důvod, proč při přidávání polohy jsou v aplikaci pouze inputy na zeměpisnou šířku a délku, a není zde mapa pro kliknutí.

Další součástí aplikace je složka modals. Ta obsahuje komponenty z balíčku Reactjs-popup. Konkrétně jsou v aplikaci využity modaly. Je to komponenta, která se zobrazí vpředu celé stránky. Lze jí zavolat například při kliknutí na tlačítko. Modaly jsou využívány na některé formuláře Formiku.

Poslední složkou je složka styledComponents. Ta obsahuje soubor device. V souboru device jsou nadefinované break-pointy pro responzivitu aplikace. Jak je vidět na obrázku číslo 16, break-pointy jsou nazvány jako zařízení, které jsou obecným předpisem pro určité rozlišení.

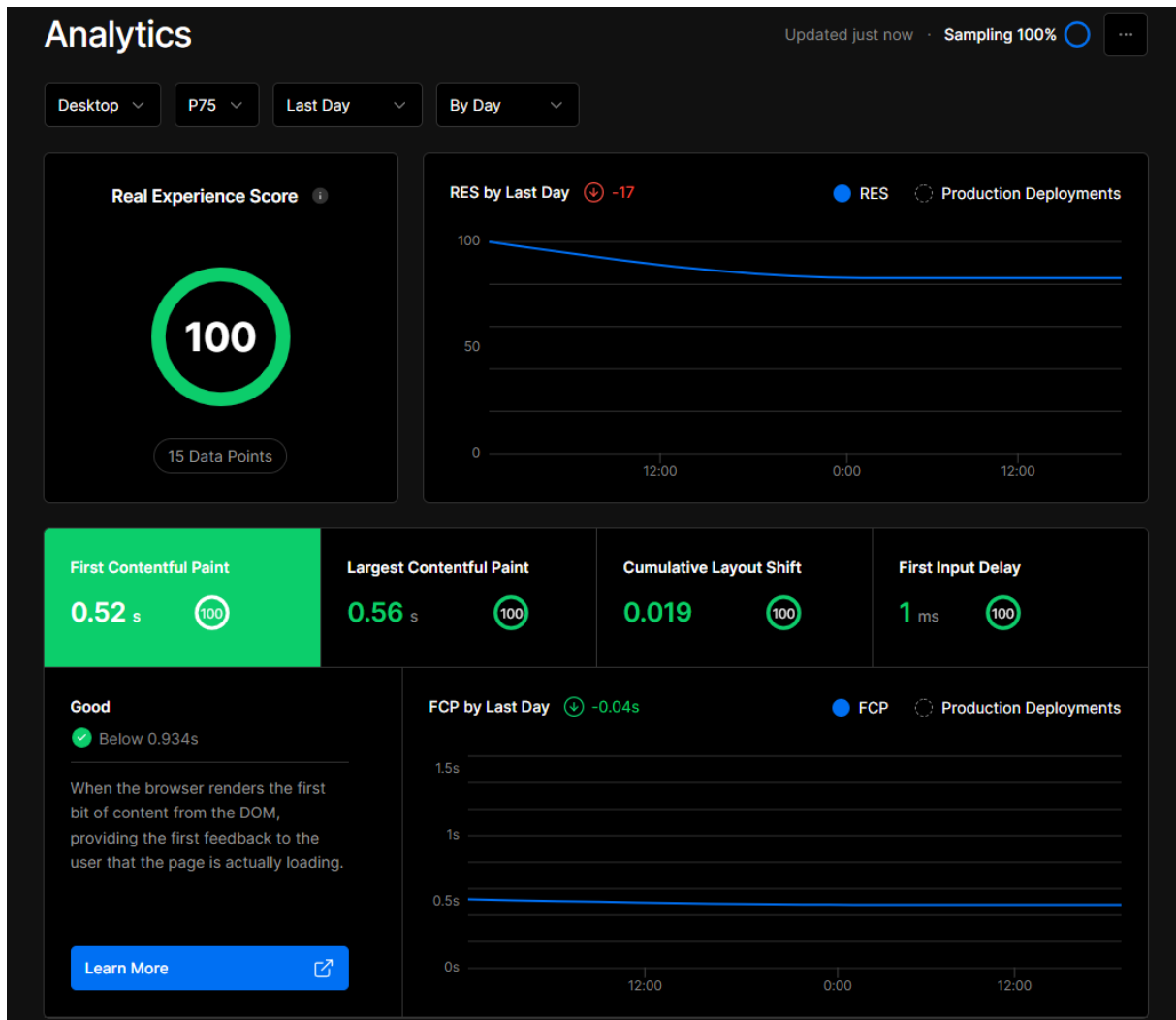
```
const size = {
  mobileS: '320px',
  mobileM: '375px',
  mobileL: '425px',
  tablet: '768px',
  laptop: '1024px',
  laptopL: '1440px',
  desktop: '2560px'
}
export const device = {
  mobileS: `(max-width: ${size.mobileS})`,
  mobileM: `(max-width: ${size.mobileM})`,
  mobileL: `(max-width: ${size.mobileL})`,
  tablet: `(max-width: ${size.tablet})`,
  laptop: `(max-width: ${size.laptop})`,
  laptopL: `(max-width: ${size.laptopL})`,
  desktop: `(max-width: ${size.desktop})`,
  desktopL: `(max-width: ${size.desktop})`
};
```

Obrázek 16 - Ukázka předdefinovaných break-pointů
[zdroj: autor]

3.3. Klient

Teď už máme přehled o tom, jak funguje backend a samotná aplikace Next.js. Tyto dvě části jsou propojeny klientem. U klienta se vykresluje samotná aplikace a následně jsou do aplikace dodána data z backendu. Data pro mapu se také dodají až ve fázi vykreslení u klienta.

Co se týče načtení dat, aplikace je na klientovi načtena velmi rychle. To lze vidět například na následujícím obrázku číslo 17.



Obrázek 17 - Vercel Analytics [zdroj: autor]

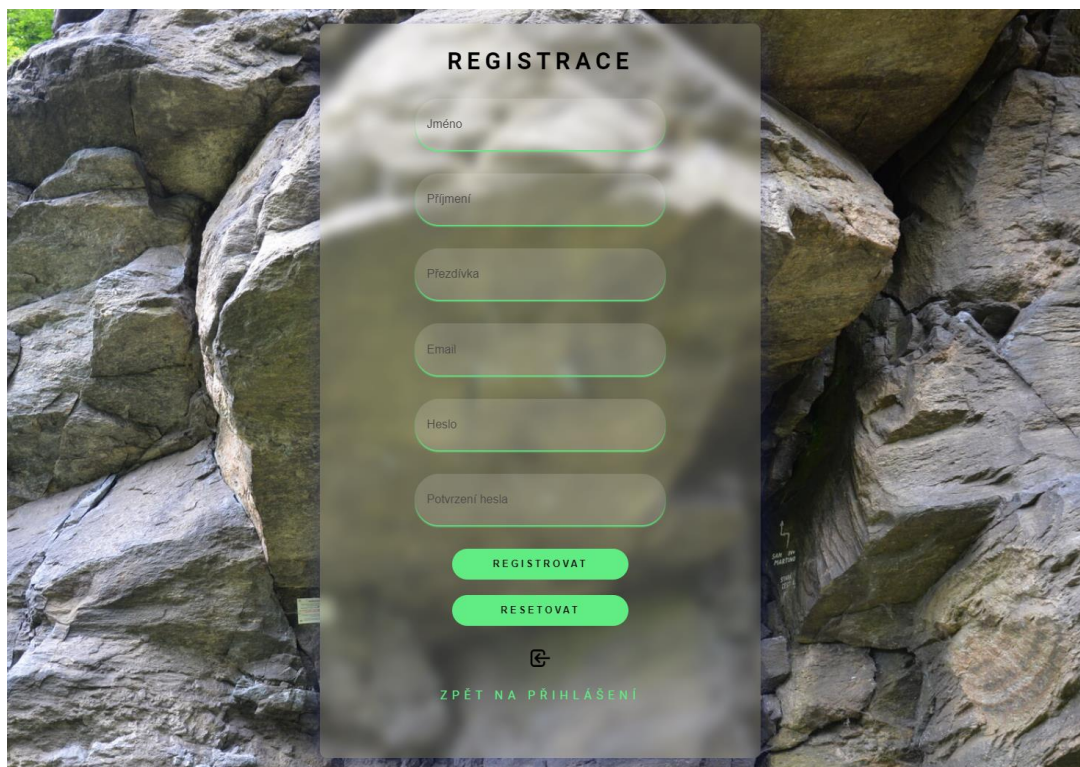
Na obrázku lze vidět analýzu, kterou poskytuje Vercel hosting. Zde je vidět, že ve veškerých aspektech je aplikace opravdu rychlá. Kvůli neplacené verzi samozřejmě není analýza úplně stoprocentní. Pro lepší pozorování by bylo zapotřebí sledovat delší časový úsek než jeden den. Rychlost načtení aplikace je samozřejmě také ovlivněna rychlostí internetu. Je však pravdou, že i pro pomalejší internetové připojení je aplikace poměrně schůdná. Záleží také na tom, jakou část aplikace chce uživatel načíst. Na některých stránkách je datový tok vyšší.

4. Webová aplikace

Následující část popisuje veškeré funkce systému, které jsou přístupné uživatelům aplikace. Jsou zde zahrnuty funkce od registrace a přihlášení, až po prohlížení dat a jejich přidávání. Dále je popsán rozdíl mezi čtyřmi typy uživatelů (neregistrovaný uživatel, registrovaný uživatel, ověřený registrovaný uživatel a admin).

4.1. Přihlášení a registrace

Přihlášení a registraci zprostředkovává Firebase authentication. Pro úspěšné přihlášení do systému je zapotřebí se registrovat. Při registraci uživatel vyplňuje své osobní údaje, které jsou zobrazené na obrázku číslo 18.

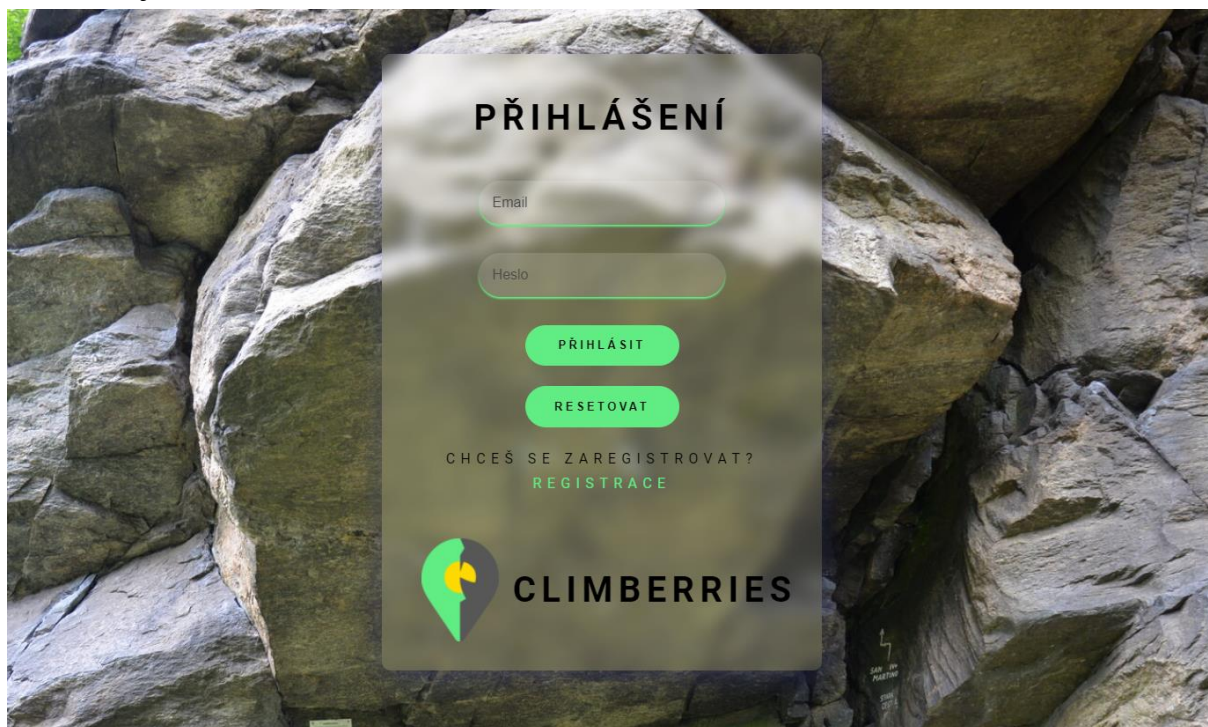


The image shows a registration form titled "REGISTRACE" overlaid on a background of large, grey, textured rocks. The form consists of several input fields with rounded corners and a light green border. The fields are labeled: "Jméno", "Příjmení", "Přezdívká", "Email", "Heslo", and "Potvrzení hesla". Below the input fields are two prominent green buttons: "REGISTROVAT" and "RESETOVAT". At the bottom of the form, there is a small back arrow icon and the text "ZPĚT NA PŘIHLÁŠENÍ" in green.

Obrázek 18 - Registrace [zdroj: autor]

Další z podmínek pro úspěšnou registraci je splnění validačních pravidel. Těmito pravidly se rozumí např. správná forma emailové adresy, shodnost hesel či požadovaná délka. Je také zapotřebí vyplnit veškeré údaje, které ve formuláři lze. Poté, po správném vyplnění, je uživatel zaregistrován. Následně je přesměrován na úvodní stránku už jako přihlášený uživatel.

V případě, že je uživatel již zaregistrován, má možnost přihlášení. Pro úspěšné přihlášení je zapotřebí vyplnit email a heslo, který byl zadán při registraci. Přihlašovací formulář je zobrazen na následujícím obrázku číslo 19.



Obrázek 19 - Přihlášení [zdroj: autor]

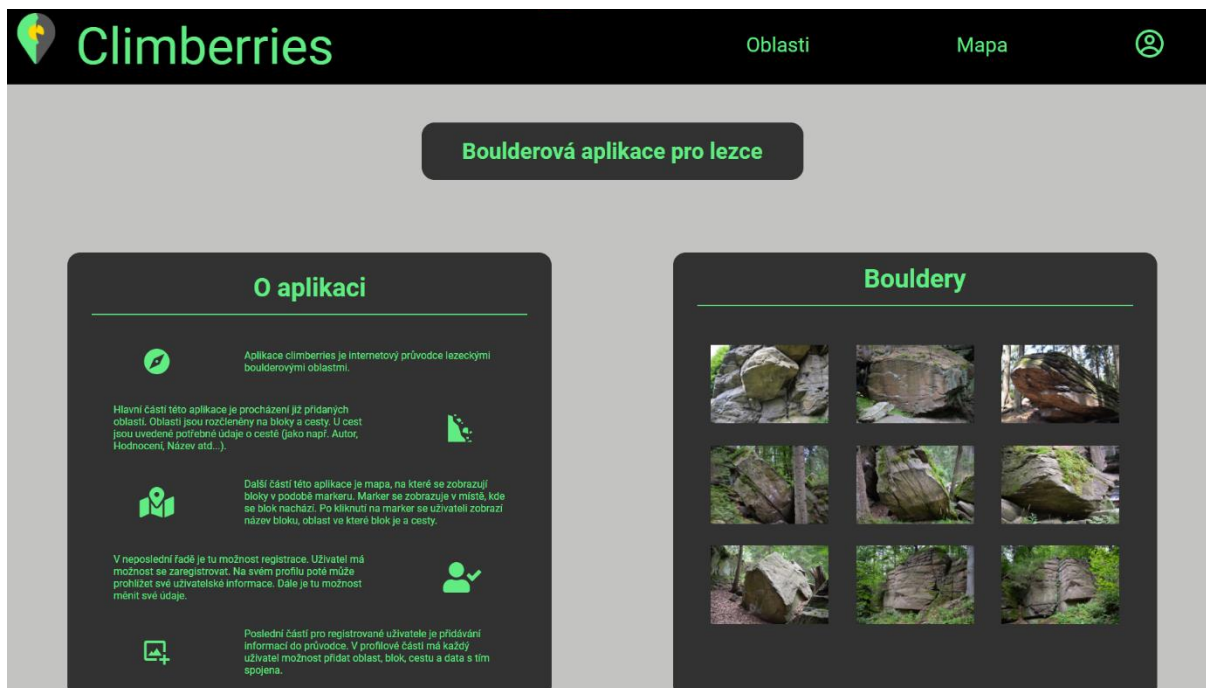
Při ověření správnosti údajů je uživatel přesměrován na úvodní stránku již jako přihlášený uživatel.

4.2. Úvodní stránka

Na úvodní stránce najde uživatel jednoduchý manuál pro lezeckého průvodce. Tento manuál obsahuje jednoduchý popis částí, ze kterých se aplikace skládá. Další částí úvodní stránky je statická galerie boulderů, kterou si můžeme prohlédnout na obrázku číslo 21. Tato galerie slouží ke zpestření vizuální stránky obsahu. Jsou zde vidět fotografie některých oblastí, které byly foceny v rámci tohoto projektu. Návštěvník aplikace si jistě povšimne i navigačního menu, které se vyskytuje v horní části stránky. Toto menu se vyskytuje téměř ve všech částech této aplikace. Jeho účelem je rychlé přesměrování na jiné části systému. Menu obsahuje logo Climberries, název, proklik na oblasti, mapu a přihlášení nebo profilovou ikonu. Tyto dvě části se zobrazují podle toho, je-li uživatel přihlášen. V případě že není, zobrazí se text „Přihlášení“. V případě, že je již uživatel přihlášen, zobrazí se profilová ikona. Rozdíl lze vidět na obrázku číslo 20.



Obrázek 20 - Navigační menu [zdroj: autor]



Obrázek 21 - Úvodní stránka aplikace [zdroj: autor]

Poslední částí stránky je patička, která se také vyskytuje ve většině částí systému.

4.3. Oblasti

Následující část aplikace Climberries se týká procházení oblastí. Do této části se lze prokliknout přes navigační menu.

4.3.1. Oblasti

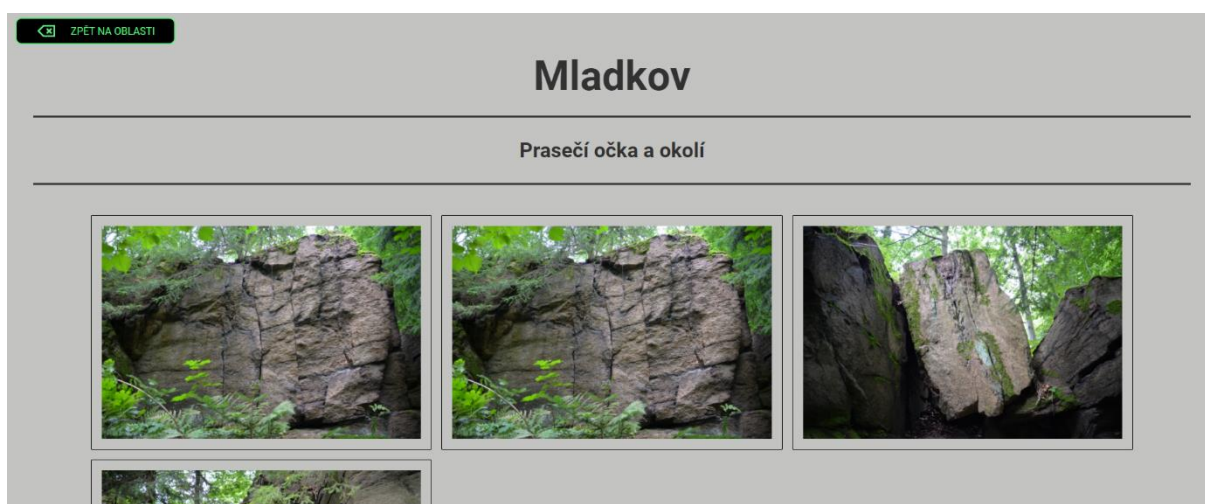
Uživatel se po prokliku zobrazí prvek s názvem oblasti a fotografií dané oblasti. Tato fotografie je náhodně vybrána z cest, které se v oblasti nachází. V případě, že v oblasti není žádná fotografie nahraná, zobrazí se uživateli alternativní obrázek. Mezi jednotlivými oblastmi se lze pohybovat pomocí šipek, které jsou vidět na obrázku číslo 22. Tyto šipky symbolizují posun vlevo či vpravo. Uživatel si tak může najít oblast, kterou chce právě procházet. Do karty konkrétní oblasti se lze přesunout pomocí kliknutí na fotografii či na název oblasti.



Obrázek 22 - Procházení oblastí [zdroj: autor]

4.3.2. Bloky

Po již zmíněném prokliku se uživateli zobrazí galerie bloků. V úvodní části galerie se zobrazuje název oblasti. Poté následuje rozčlenění cest do blokových sekcí. Tyto sekce se řadí pod název oblasti. To lze vidět na obrázku číslo 23. Každá sekce má svůj název, který je identický s názvem bloku. V sekci se zobrazují cesty, které jsou k bloku přiřazeny. Každou cestu reprezentuje fotografie, která je k cestě přiřazena. Po najetí kurzoru myši na fotografii se zobrazí název cesty, kterou fotografie představuje. Tento název zároveň slouží jako proklik na detail cesty. Na stránce se také zobrazí tlačítko pro zpětné vrácení na kartu oblastí.

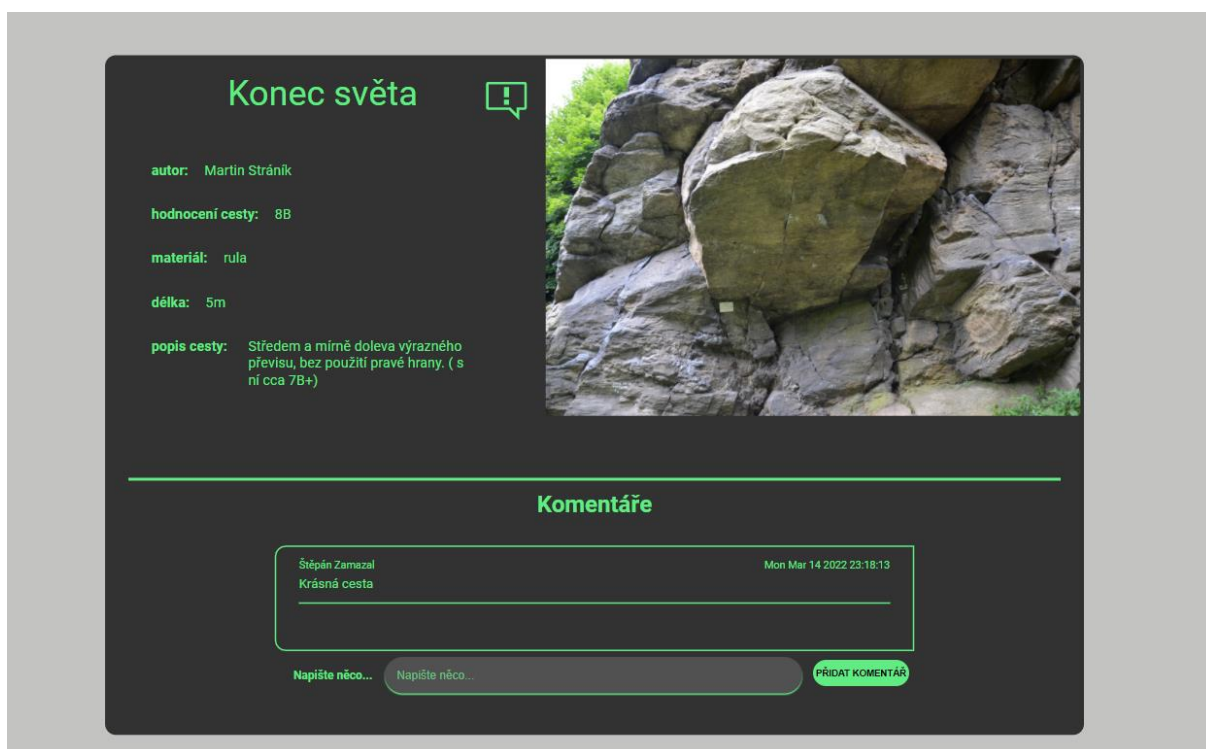


Obrázek 23 - Procházení bloků [zdroj: autor]

4.3.3. Cesty

Poslední částí na kartě Oblasti je detail samotné cesty. Na tomto detailu se uživateli zobrazí konkrétní data, která jsou k cestě přiřazena. Těmito daty se rozumí: název, autor, hodnocení cesty, materiál, délka, popis cesty, fotografie a komentáře. Uživatel si tak může přečíst všechna potřebná data k dané cestě a s pomocí fotografie identifikovat cestu při návštěvě oblasti. Vedle názvu oblasti se v případě registrovaného uživatele, který má ověřený email, zobrazí tlačítko pro reportování. Toto reportování slouží k uvědomění adminů v případě nesedících dat či nevhodné fotografie. Po kliknutí na ikonu reportu se uprostřed obrazovky zobrazí jednoduchý formulář k odeslání zprávy reportu.

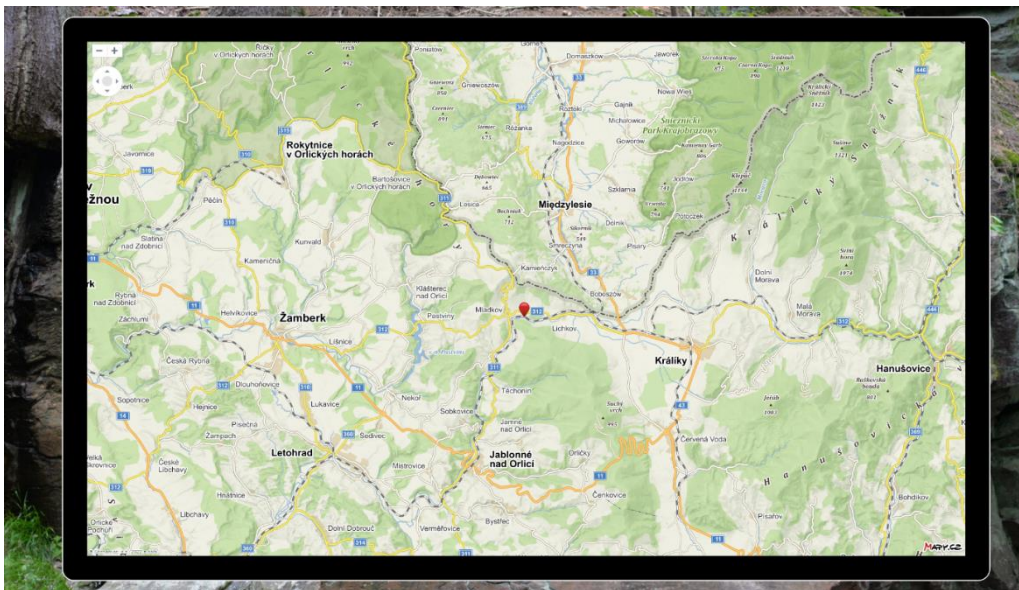
Další částí detailu je komentářová sekce, kterou můžeme vidět na obrázku číslo 24 ve spodní části. V ní se zobrazují komentáře uživatelů, kteří je přidali. Komentář obsahuje text, jméno uživatele, který komentář přidal a datum společně s časem přidání. Přidávat komentáře mohou pouze uživatelé, kteří jsou zaregistrovaní v systému a mají ověřený email. V případě, že uživatel nesplní tyto podmínky, má možnost pouze prohlížet komentáře. Možnost přidání se nezobrazí. Komentáře díky Firebase funkci onSnapshot fungují v reálném čase.



Obrázek 24 - Detail cesty [zdroj: autor]

4.4. Mapa

Mapu si uživatel může zobrazit přes proklik z navigačního menu. Uprostřed stránky se poté zobrazí mapa. Po této mapě se uživatel může pohybovat pomocí tlačítek zobrazených na mapě nebo pomocí myši. Hlavní součástí mapy jsou markery. Tyto markery slouží k určení polohy bloků. Marker má nastavené atributy, které určují jeho polohu. Tyto atributy jsou zeměpisná šířka a délka. Uživatel má také možnost kliknout na tento marker. Po kliknutí se zobrazí karta markeru. Na této kartě jsou vypsána data, která se daného bloku týkají. V hlavičce karty je uvedený název bloku. V těle karty se uživateli vypíšou všechny cesty, které se na daném bloku nachází. V patičce karty se zobrazuje název oblasti, ke které je blok přiřazen. Mapu si lze prohlédnout na obrázku číslo 25.



Obrázek 25 - Mapa [zdroj: autor]

4.5. Profil

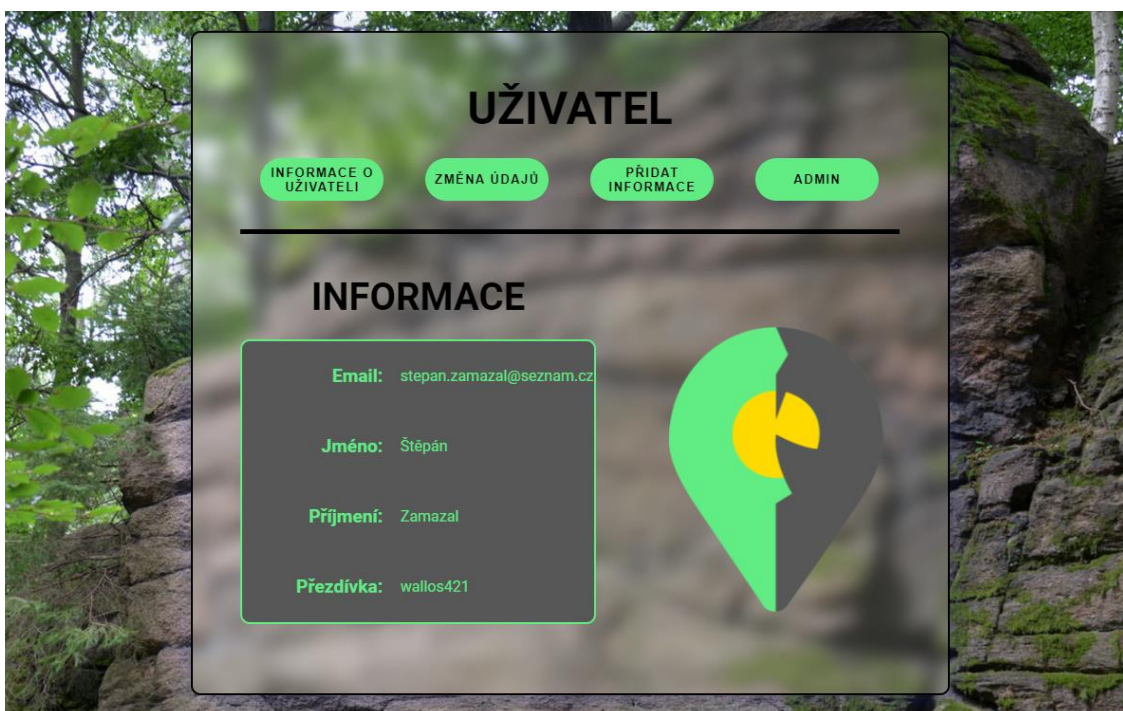
Do profilové části aplikace má uživatel přístup pouze tehdy, je-li přihlášený. V opačném případě se uživatel na stránku profil nedostane. Na uživatelský profil se lze dostat pomocí kliknutí na profilovou ikonu. Při kliknutí se uživateli zobrazí dropdown menu. V tomto menu se nacházejí dvě tlačítka, jak můžeme vidět na obrázku číslo 26. Pomocí tlačítka profil se uživatel přesměruje na stránku svého osobního profilu. Při kliknutí na tlačítko odhlásit je uživatel odhlášen ze systému a přesměrován na úvodní stránku.



Obrázek 26 - Dropdown menu [zdroj: autor]

4.5.1. Uživatelské informace

Po přesměrování na stránku osobního profilu se uživateli zobrazí některé informace, které zadal při registraci. Tyto informace jsou: email, jméno, příjmení a uživatelská přezdívka. Zadané informace jsou pak využívány například u komentářů. Součástí osobního profilu je také menu s tlačítky, kde se zobrazují ostatní možnosti profilu. Mezi možnosti můžeme počítat: informace o uživateli, změnu údajů, ověření emailu, přidání informací a možnost přesměrování na admin stránku. Každý uživatel má svá určitá práva. Podle těchto práv se zobrazují jeho možnosti. Na obrázku číslo 27 lze vidět menu uživatele, který má všechna možná práva.



Obrázek 27 - Profil [zdroj: autor]

4.5.2. Změna údajů

Jednou z možností uživatele je změna jeho osobních údajů. Tuto možnost má každý zaregistrovaný uživatel. Uživatel může měnit své jméno, příjmení, přezdívku a také heslo. Pro změnu údajů musí uživatel využít formulář, který se v této možnosti zobrazuje. Pro změnu hesla se na stránce zobrazuje tlačítko změnit heslo. Změny musí opět projít přednastavenými validačními pravidly. Změnit lze jeden či více údajů. Obrázek číslo 28 zobrazuje formulář pro změnu údajů.



The image shows a user profile interface with a dark background and light text. At the top, the word 'UŽIVATEL' is displayed in large, bold, white letters. Below it, there are four green buttons with white text: 'INFORMACE O UŽIVATELI', 'ZMĚNA ÚDAJŮ', 'PŘIDAT INFORMACE', and 'ADMIN'. A horizontal line separates this header from the main content area. The main content area is titled 'ZMĚNIT ÚDAJE' in bold, white letters. Below the title, there are three input fields, each with a label on the left and a text input field on the right. The labels are 'Jméno', 'Příjmení', and 'Přezdívka'. The input fields contain the text 'Jméno', 'Příjmení', and 'Přezdívka' respectively. At the bottom of the form, there are three green buttons with white text: 'ZMĚNIT', 'RESETOVAT', and 'ZMĚNIT HESLO'.

Obrázek 28 - Změna údajů [zdroj: autor]

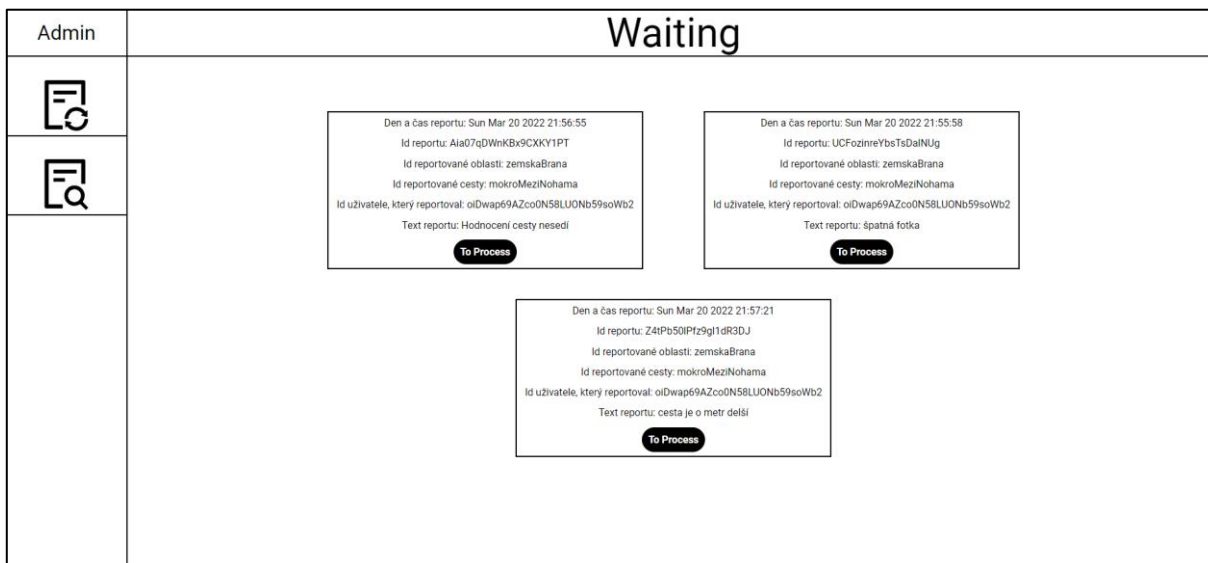
4.5.3. Ověření emailu a přidávání informací

Další z možností je přidávání dat do databáze. Pro získání této možnosti je zapotřebí, aby si uživatel ověřil svůj email. V případě, že uživatel nemá ověřený email, zobrazuje se na profilu tlačítko ověřit email. Po kliknutí na tlačítko se uživateli odešle do emailové schránky ověřovací zpráva, přes kterou si lze adresu emailu ověřit. Po znovunačtení stránky profilu už uživatel uvidí možnost přidat informace. Po kliknutí na tlačítko přidat informace se zobrazí karta pro přidávání informací. Na této kartě má uživatel možnost přidat samostatná data. Přidat lze oblast, blok nebo cestu, jak lze vidět na obrázku číslo 29. V každé z možností se nachází formulář, který je zapotřebí vyplnit pomocí validačních podmínek. Po odeslání formuláře jsou data přidána do databáze.

Obrázek 29 - Formulář pro přidávání informací [zdroj: autor]

4.5.4. Admin

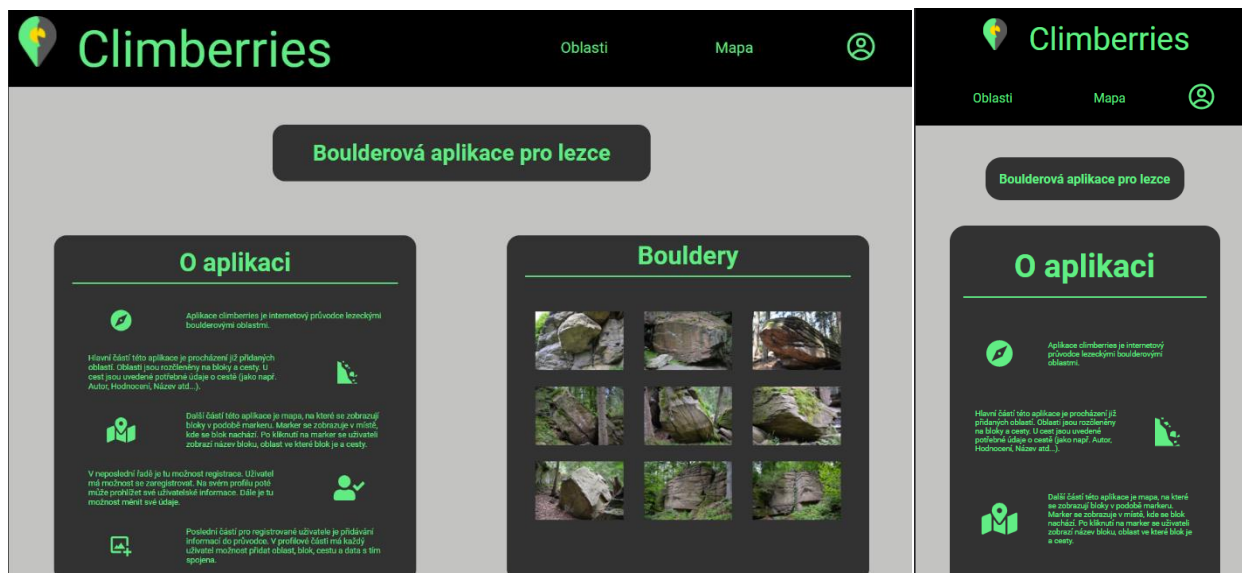
Poslední z profilových možností je stránka pro adminy. V případě, že uživatel je vedený jako admin, má na svém profilu možnost přesměrování na admin stránku přes tlačítko admin. Na této stránce se zobrazují reporty uživatelů. Admin má možnost reporty spravovat. Reporty mají dva stavy: waiting a in process. Stav reportu waiting znamená, že se admin ještě o report nezajímá. Tento report tak čeká na přesun do stavu in process. Stav in process znamená, že se admin o daný problém zajímá a aktivně se ho snaží řešit. V případě, že je problém popisovaný v reportu vyřešený, admin může tento report odstranit. Admin má reporty rozdělené do záložek. Každá záložka obsahuje reporty se stejným stavem. Mezi záložkami se lze pohybovat pomocí tlačítek, které lze vidět na obrázku číslo 30. Report obsahuje den a čas reportu, ID reportu, ID reportované oblasti a cesty, ID uživatele, který report odeslal a text reportu, ve kterém uživatel popisuje daný problém.



Obrázek 30 - Admin stránka [zdroj: autor]

4.5.5. Responzivita webu

Web je navrhnutý tak, aby si ho každý uživatel mohl zobrazit v rozlišení do 320px. Tento bod je zvolený proto, že pod tuto hranici je obtížné zobrazit veškerý obsah stránky tak, aby byl pro uživatele čitelný a přehledný. Základním principem jsou zlomové body, od kterých se rozlišení škáluje jiným způsobem. To lze vidět například na obrázku číslo 31. Ve většině případů se obsah, který lze u širokých rozlišení vidět vedle sebe, zobrazí pod sebou.



Obrázek 31 - Ukázka responzivity [zdroj: autor]

5. Budoucnost aplikace

5.1. Co v aplikaci chybí?

Je jasné, že aplikace není ve finálním stavu. Je zde hodně nedostatků. Co se týče uživatelských možností, není zde například možnost mazání vlastních komentářů či smazání celého účtu. Dále zde chybí vyhledávací pole v části oblastí. Uživatel se musí na cesty postupně proklikat. Do adminské části aplikace by bylo zapotřebí udělat určité prostředí pro adminy. Toto prostředí by sloužilo k upravování celé databáze. Takto mají možnost měnit data v databázi pouze uživatelé, kteří mají přímo přístup k databázi. Se vzhledem webu také nejsem spokojený. Opravdu nejsem grafik, to je na aplikaci poznat. I když za sebe musím říct, že je to na mé poměry dobrý výkon. Do budoucna bude vzhled aplikace ještě změněn. Návrh webu by za mě měl dělat někdo, kdo tomu rozumí a kdo má na to oko. Aplikace rozhodně není plně zabezpečena. Uživatelské vstupy sice ošetřuje Yup, ale pravidla jsou nastavená tak, že každý uživatel, který se zaregistruje a ověří si email, má možnost upravovat data. To by chtělo do budoucna promyslet. Odchytávání závadného obsahu funguje pouze na bázi uživatelů a reportování. Vhodné by bylo například využít umělé inteligence, která by předem rozpoznala závadný obsah a nepovolila by zapsání do databáze.

5.2. Co bude dál?

Aplikace je ve fázi, ve které by se dala začít uživatelsky testovat. V plánu tedy je, že se nejdříve doladí frontend celé aplikace. Poté by se aplikace nasdílela pár lezcům, kteří by aplikaci začali testovat. Testeři by zaslali poznámky, které by obsahovali jejich poznatky o tom, co by bylo dobré do aplikace přidat nebo jak aplikaci vylepšit.

Dále je v plánu jedna specifická funkce systému. Původně bylo v plánu, že by u přidávání cesty byla možnost, zaznamenat na fotografii, kudy daná cesta vede. Kvůli časové dotaci však nebyla šance toto implementovat. Nejspíše to však bude fungovat na bázi souřadnic, jejichž rozpětí bude stejné jako velikost fotky. Bude se jednat o souřadnice X a Y. Na fotku bude mít uživatel možnost přidat tečky. Pozice těchto teček se uloží do databáze a následně se tyto tečky budou vykreslovat u detailu fotky. Implementovány budou tři typy teček. Jedna barva tečky bude symbolizovat začáteční chyty cesty. Druhá barva by zobrazovala chyty, které se berou při cestě k finálnímu chytu. Poslední barvou tak dává smysl, že by byl finální chyt.

Ve chvíli, kdy budu s aplikací plně spokojen, přidám si do Firebase projektu platební kartu. Firebase mi tak bude účtovat případné přesažení kvóty. Kvóty například omezují počet přečtení za den či počet možných zápisů do databáze. Ve výsledku by to mohlo vyjít na hodně peněz, ale v plánu je určitý sponzoring od lezecké komunity. Následně bude aplikace nasdílena do

lezeckých skupin a začne si žít svým životem. Fungování aplikace budou kontrolovat admini a případné chyby se budou řešit.

6. Závěr

Cílem projektu bylo vytvořit aplikaci, která by byla lezeckým průvodcem po skalních oblastech. To bylo jednoznačně splněno. S aplikací jsem poměrně spokojen. Sice zde chybí některé věci, které jsem chtěl implementovat, ale hlavní části v aplikaci fungují. Už nyní je odezva od některých lezců velice kladná. Jediné výtky jsou snad jenom na grafické zpracování. Aplikace je však velice interaktivní a opravdu by se dala používat. Při tvorbě aplikace jsem si vyzkoušel nové technologie, které jsem doposud neznal. Načerpal jsem tak nové znalosti, které určitě budu moci v budoucnu využít. Práce na projektu mě ohromně bavila. S technologiemi jsem se sžil, a nakonec si myslím, že technické řešení je opravdu dobré. Těším se na budoucí vývoj této aplikace. Primárně hlavně chci, aby byla aplikace v budoucnu opravdu využívána a aby odezva od širší skupiny lezců byla dobrá.

7. Seznam použité literatury

1. What Is Node.js and Why You Should Use It. Kinsta [online]. 2022, 18. 1. 2022 [cit. 2022-03-24]. Dostupné z: <https://kinsta.com/knowledgebase/what-is-node-js/#nodejs-architecture-and-how-it-works>
2. Next.js for CTOs. Why do we love Next.js & what is it used for?. The Software House [online]. Gliwice: Witold Ossera, 2021, 16. 03. 2021 [cit. 2022-01-24]. Dostupné z:
3. Get Started with Next.js: The React Library Your Project Needs. FreeCodeCamp [online]. Oakland: Reed Barger, 2022, 14. 01. 2022 [cit. 2022-01-24]. Dostupné z: <https://www.freecodecamp.org/news/nextjs-tutorial/#what-features-next-js-gives-you>
4. Lekce 1 - Úvod do React. ITnetwork.cz [online]. Praha: itnetwork.cz, 2019 [cit. 2022-01-24]. Dostupné z: <https://www.itnetwork.cz/javascript/react/zaklady/uvod-do-react/>
5. Firebase Documentation [online]. Google [cit. 2022-01-30]. Dostupné z: <https://firebase.google.com/docs>
6. React-mapycz [online]. Falsy [cit. 2022-01-30]. Dostupné z: <http://falsy.cz/react-mapycz/>
7. KVAPIL, Jiří. Úvod do TypeScriptu. Itnetwork.cz [online]. [cit. 2022-03-27]. Dostupné z: <https://www.itnetwork.cz/javascript/typescript/uvod-do-typescriptu>
8. Basics: Motivation. Styled components [online]. [cit. 2022-03-23]. Dostupné z: <https://styled-components.com/docs/basics>
9. CALDWELL, Ronald. What is Formik?. Liquid Web [online]. 2021, 21. 7. 2021 [cit. 2022-03-24]. Dostupné z: <https://www.liquidweb.com/kb/formik-react/>
10. Introduction. Reactjs-popup [online]. [cit. 2022-03-27]. Dostupné z: <https://react-popup.elazizi.com/>
11. GARCÍA SIERRA, Francisco. Working with Firestore: Building a simple database model. ProAndroidDev [online]. 2018, 19. 5. 2018 [cit. 2022-03-27]. Dostupné z: <https://proandroiddev.com/working-with-firestore-building-a-simple-database-model-79a5ce2692cb>

8. Seznam obrázků

Obrázek 1 - Struktura databáze Firestore [11]	11
Obrázek 2 - Ukázka Firebase Security Rules [zdroj: autor].....	11
Obrázek 3 - Ukázka styled komponenty [zdroj: autor]	13
Obrázek 4 - Architektura aplikace [zdroj: autor]	14
Obrázek 5 - Ukázka struktury kolekce boulders [zdroj: autor]	15
Obrázek 6 - Ukázka struktury kolekce comments [zdroj: autor]	16
Obrázek 7 - Ukázka struktury kolekce reports [zdroj: autor].....	17
Obrázek 8 - Ukázka nastavení Firebase Security Rules [zdroj: autor]	17
Obrázek 9 - Složka Users [zdroj: autor].....	18
Obrázek 10 - Ukázka struktury kolekce users [zdroj: autor]	18
Obrázek 11 - Ukázka uložené fotky [zdroj: autor].....	19
Obrázek 12 - Složky v Cloud Storage [zdroj: autor].....	20
Obrázek 13 - Ukázka struktury složky pages [zdroj: autor].....	21
Obrázek 14 - Ukázka adresového řádku [zdroj: autor]	21
Obrázek 15 - Ukázka struktury složky components [zdroj: autor]	22
Obrázek 16 - Ukázka předdefinovaných break-pointů [zdroj: autor]	23
Obrázek 17 - Vercel Analytics [zdroj: autor].....	24
Obrázek 18 - Registrace [zdroj: autor]	25
Obrázek 19 - Přihlášení [zdroj: autor].....	26
Obrázek 20 - Navigační menu [zdroj: autor]	27
Obrázek 21 - Úvodní stránka aplikace [zdroj: autor].....	27
Obrázek 22 - Procházení oblastí [zdroj: autor]	28
Obrázek 23 - Procházení bloků [zdroj: autor].....	28
Obrázek 24 - Detail cesty [zdroj: autor].....	29
Obrázek 25 - Mapa [zdroj: autor].....	30
Obrázek 26 - Dropdown menu [zdroj: autor].....	30
Obrázek 27 - Profil [zdroj: autor].....	31
Obrázek 28 - Změna údajů [zdroj: autor].....	32
Obrázek 29 - Formulář pro přidávání informací [zdroj: autor].....	33
Obrázek 30 - Admin stránka [zdroj: autor]	34
Obrázek 31 - Ukázka responzivity [zdroj: autor].....	34