

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor č. 18: Informatika

Histories.cc

platforma pro sdílení historických fotografií

Krátký Kryštof
Pardubický kraj

Pardubice, 2022

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor č. 18: Informatika

Histories.cc

platforma pro sdílení historických fotografií

Jméno: Krátký Kryštof

Škola: SŠIE Delta

Kraj: Pardubický

Konzultant: RNDr. Koupil Jan, Ph.D.

Prohlášení

Prohlašuji, že jsem svou práci SOČ vypracoval/a samostatně a použil/a jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů.

Prohlašuji, že tištěná verze a elektronická verze soutěžní práce SOČ jsou shodné.

Nemám závažný důvod proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Pardubicích dne 24. března 2022

Krátký Kryštof

Poděkování

Rád bych poděkoval svému konzultantovi RNDr. Janu Koupilovi, Ph.D. za odborné vedení a konzultace při tvorbě tohoto projektu.

Anotace

Cílem této práce SOČ je vytvořit platformu postavenou na moderních technologiích, která umožní jednoduché sdílení historických fotografií. Tato platforma dále umožňuje například prohlížení fotografií na mapě, řazení fotografií podle data kdy byly pořízeny, nebo komunikaci s ostatními uživateli díky možnosti komentovat příspěvky.

Klíčová slova

Webová aplikace, Historické fotografie, Sdílení fotografií, Next.js, Neo4j

Annotation

The goal of this work is to create a platform based on modern technologies that will allow easy sharing of historical photographs. This platform also allows, for example, viewing photos on a map, sorting photos by the date they were taken or communicating with other users by commenting on posts.

Keywords

Web app, Historical photos, Photos sharing, Next.js, Neo4j

OBSAH

1	Pojmy	2
2	Úvod	3
2.1	Existující řešení	3
3	Použité technologie	4
3.1	Technologie použité na frontendu i na backendu	4
3.2	Technologie pro frontend	4
3.3	technologie pro backend	5
4	Úložiště dat	7
4.1	Úložiště souborů	7
4.2	Databáze	8
5	Obrázky	13
5.1	Zpracování při nahrání	13
5.2	Optimalizace a použití komponenty Image	13
5.3	Blurhash	13
6	Struktura projektu	15
6.1	Oprávnění uživatelů	15
6.2	Historické datum příspěvku	15
6.3	Uživatelské vstupy	16
7	Frontend	18
7.1	Struktura React komponent	18
7.2	Lokalizace	19
7.3	Stránka s mapou	19
7.4	Stránka domů	21
7.5	Komentáře	22
7.6	Profilová stránka uživatele	22
7.7	Stránka detail příspěvku	25
8	Vývoj	26
8.1	Nasazení na server	26
8.2	Verzování	26

8.3 Testování	26
9 Závěr	27
A Zadání maturitního projektu	33

1 POJMY

1.0.1 NSFW

Zkratka vycházející z anglického „Not Safe For Work“ (nevhodné pro práci), zpravidla se takto označují příspěvky obsahující nahotu, nebo násilí, a tudíž by se jim mělo vyhnout v profesionálním prostředí. [1]

1.0.2 Garbage collection

Garbage collector se stará se o uvolnění úseků v paměti, které nejsou žádným procesem využívány [2]

1.0.3 Runtime environment

Jedná se o skupinu software umožňující spuštění programu v reálném čase, obvykle implementuje základních funkce jako je například výpis textu nebo připojení k internetu [3]

2 ÚVOD

Všichni ví, jak vypadá těch pár míst, se kterými se každodenně setkávají. Málo kdo ale tuší, jakou zajímavou mají daná místa historii, jak vypadala před 20, nebo 100 lety. Existují online komunity, které si přesně toto uvědomují. Bohužel jsou tyto komunity nucené využívat standardní platformy, které nemají fungovat jako archiv, ale naopak počítají pouze s prohlížením nově nahraného obsahu. To činí prohledávání, nebo řazení fotografií téměř nemožným. Cílem tohoto projektu bylo vytvořit platformu, která bude na míru šitá potřebám zmíněných komunit a umožní komukoliv snadné prohlížení historických fotografií. Mezi hlavní funkce patří zobrazování míst na mapě, komunikace s ostatními uživateli pomocí komentářů a možnost filtrování a řazení fotografií podle data jejich vzniku.

2.1 EXISTUJÍCÍ ŘEŠENÍ

2.1.1 Historypin

Historypin je open-source platforma fungující jako archiv historických fotografií. Umožňuje vyhledávání míst na mapě, filtrování příspěvků podle data pořízení fotografie, nebo např. vytváření kolekcí. Obsahuje velké množství příspěvků, což kvůli špatně optimalizaci stránky vede k pomalému načítání dat a nepřehlednosti mapy. Celá platforma je v angličtině a česká komunita na ní téměř neexistuje.

2.1.2 Facebook

Facebook je jedna z nejznámějších sociálních sítí zaměřující se na komunikaci uživatelů. Uživatelům umožňuje vytváření skupin (komunit), nebo například přidávání příspěvků. Primárním účelem Facebooku zůstává sdílení co nejaktuálnějšího obsahu. Neposkytuje téměř žádné možnosti filtrování příspěvků, či propojení s mapou. Ačkoliv zde jsou komunity jako je například „Pardubice v běhu času“, nebo „Historické Karlovy Vary ve fotografii“.

3 POUŽITÉ TECHNOLOGIE

V projektu byly použity moderní a v současné době jedny z nejpoužívanějších technologií pro zajištění co nejděší podpory aplikace.

3.1 TECHNOLOGIE POUŽITÉ NA FRONTENDU I NA BACKENDU

3.1.1 Typescript

Typescript je nadstavba na JavaScript, umožňující statické typování [4]. Veškerý kód napsaný v JavaScriptu je validním také v Typescriptu. [5][6]

3.1.2 NPM

NPM je největším registrem JavaScriptových balíčků (knihoven) tvořených komunitou. Umožňuje jednoduchou instalaci balíčků pomocí CLI (command line interface). Konfigurační soubor obsahuje seznam všech balíčků použitých v projektu včetně jejich verzí, což zamezuje chybám v kódu z důvodu změn v nových verzích balíčků. [7][8]

3.1.3 GraphQL

GraphQL je dotazovací jazyk pro API [9]. Nejpoužívanější alternativou pro GraphQL je REST. Oproti REST u GraphQL klient v requestu specifikuje požadovaná data, která jsou následně vrácena serverem. Tím je eliminován přenos nepotřebných, nebo duplicitních dat, čímž je celá operace urychlena. GraphQL je silně typované [10], což znamená že musí být předem stanovený typ vrácených dat. [11][12]

3.2 TECHNOLOGIE PRO FRONTEND

Část webové aplikace, se kterou uživatel přímo interaguje (klientská strana) je označována za frontend aplikace. Obsahuje všechny prvky, se kterými se uživatel dostává do kontaktu přímo (např. texty, obrázky, tlačítka atd.). Frontend uživateli umožňuje ovládat aplikaci bez znalosti programování, nebo hlubší znalosti technologií, na kterých je postavena. Za frontend lze označit také mobilní, nebo desktopové aplikace. [13]

3.2.1 React

React je open-source JavaScriptová knihovna (občas bývá označován za framework) využívaná na frontendu a pro tvorbu uživatelských rozhraní složených z komponent viz 7.1. Komponenty jsou části kódu, které lze znovu používat čímž se předchází duplicitnímu kódu. Mezi hlavní funkce Reactu patří také state management a následné vykreslování DOMu (document object model). DOM je objektově orientovaná reprezentace webové stránky usnadňující její modifikaci. [14][15]

3.2.2 Next.js

Next.js je React framework zaměřený na zlepšení uživatelské i vývojářské zkušenosti. Hlavní výhodou Next.js je SSR (server side rendering) [16], který zajišťuje načtení dat na serveru a následné poslání načtené stránky klientovi, na rozdíl od Reactu, kde je klientovi poslána prázdná stránka a až poté se načítají data. SSR zároveň zlepšuje skóre SEO [17]. Next.js disponuje vlastními React komponentami, např. komponentou *Link*, která zajišťuje automatický prefetch odkazů, nebo *Image*, která optimalizuje načítání obrázků viz kapitola 5.2. Prefetch dovoluje prohlížeči na pozadí ukládat do cache data, která by uživatel mohl v blízké budoucnosti potřebovat. Jakmile uživatel klikne na link, obsah je načten instantně. Next.js obsahuje také file system rounting podrobněji popsany v kapitole 7. [18]

3.2.3 Tailwind

Tailwind je open-source CSS framework, který umožňuje rychlejší a přehlednější psaní kaskádových stylů pomocí definovaných HTML tříd. Tailwind neobsahuje žádné předvytvořené styly pro komponenty na rozdíl od alternativ jako je např. Bootstrapu. [19][20]

3.2.4 GraphQL code generator

GraphQL code generator je knihovna, umožňující generování typů pro Typescript z GraphQL schématu. Díky tomu odpadá nutnost psát typy ručně, čímž je odstraněn jeden možný bod selhání. [21]

3.3 TECHNOLOGIE PRO BACKEND

Za backend je označována část aplikace běžící na serveru, se kterou lze komunikovat pouze pomocí API. Tato část může starat například o ověřování vstupů, rozesílání emailů, veškeré zápisy do databáze a manipulaci s daty. Uživatel nemá přímý přístup k této části aplikace. [13]

3.3.1 Node.js

Node.js je runtime environment pro spouštění JavaScriptového kódu v reálném čase mimo prohlížeč. Hlavním důvodem výběru bylo použití stejného programovacího jazyka pro backend

i frontend, což umožňuje opakované využívání některých funkcí (např. validace) a usnadňuje práci a orientaci v projektu. [22]

4 ÚLOŽIŠTĚ DAT

4.1 ÚLOŽIŠTĚ SOUBORŮ

4.1.1 Protokol IPFS

Inter Planetary File System (meziplanetární souborový systém) je peer-to-peer trustless protokol pro ukládání a sdílení souborů v distribuovaném souborovém systému. [23]

Decentralizace

Decentralizované protokoly nemají vlastníka a jsou utvářeny všemi uživateli společně. Tento koncept často bývá spojen s konceptem trustless, který umožňuje každému uživateli kdykoliv ověřit pravdivost obsahu (nemusí jen slepě důvěřovat ostatním uživatelům), v tomto případě k ověření dochází vygenerováním hashe přijatého souboru a následným porovnáním s očekávaným hashem, díky čemuž není možné obsah cenzurovat, nebo jakkoliv modifikovat [24]. Decentralizace umožňuje rychlejší načtení souboru, pokud se v blízkosti nachází node poskytující soubor, nebo načtení souboru bez nutnosti přístupu k celosvětové síti, pokud se node poskytující obsah nachází v lokální síti. [25][26]

Zpracování obsahu

Nahrání souboru do IPFS je doprovázeno vygenerováním hashe obsahu nahraného souboru, který dále slouží jako unikátní identifikátor souboru (CID - content identifier), pomocí kterého je následně soubor přístupný. Vygenerovaný hash má vždy stejnou délku nezávisle na velikosti nahraného souboru a při nahrání souboru se shodným obsahem z jakéhokoliv zařízení bude vždy hash totožný, čímž je zamezeno ukládání duplicitního obsahu. Nahraný soubor si může jakýkoliv node připnout, což zabrání garbage collectoru před jeho smazáním. Po připnutí je soubor uložen do lokálního úložiště nodu a dále poskytován síti. Existují tzv. pinning services, které umožňují pronajmutí nodu, spravovaného poskytovatelem s přístupem k vysokorychlostnímu internetu. V projektu byl použit pinning service Infura. [27][28]

Přístup k obsahu je možný skrz veřejnou gateway (např. *ipfs.io*, nebo *cf-ipfs.com*), nebo skrz gateway nodu, kde jsou soubory připnuty. Přístup přes gateway určitého nodu umožňuje vynechat krok vyhledávání nejbližšího nodu poskytujícího soubor, čímž zrychluje proces získání dat. [29]

4.1.2 Porovnání IPFS s S3

S3 je centralizované škálovatelné objektové úložiště dostupné přes webovou službu. Soubory v IPFS jsou adresovány na základě obsahu na rozdíl od S3, kde jsou soubory adresovány na základě lokace (což umožňuje nahrávat soubory do složek). IPFS na rozdíl od S3 neumožňuje nastavení oprávnění k přístupu k souboru, tzn. pokud zná uživatel CID souboru, který je v síti dostupný, nemůže mu být zabráněno v přístupu k tomuto souboru. [30]

4.2 DATABÁZE

Databáze je organizovaná kolekce strukturovaných dat ukládaná elektronicky v počítači, nebo na serveru (obvykle v cloudu). Databáze je obvykle ovládána pomocí DBMS (systém správy databází). Data lze jednoduše přidávat, upravovat, mazat a vyhodnocovat. Většina databází používá pro přístup k datům dotazovací jazyk (u relačních databází je to nejčastěji jazyk SQL). [31]

ACID model je důležitým konceptem v teorii databází. Skládá se z následujících čtyř podmínek která musí DBMS splňovat:

- **Atomicity** (atomicita) stanovuje že při zápisu dat do databáze jsou zapsána buď všechna data, nebo žádná. Pokud selže jakákoliv část transakce, data jsou vrácena do stavu ve kterém byla před začátkem transakce. Toto chování musí být zachováno i při selhání DBMS, operačního systému, nebo hardwaru. [32]
- **Consistency** (konzistence) zaručuje zapsání pouze platných dat. Pokud je provedena transakce, která porušuje pravidla konzistence databáze, je celá transakce vrácena zpět a databáze je obnovena do předchozího stavu, který je v souladu s těmito pravidly. Pokud je transakce provedena úspěšně, přivede databázi z jednoho konzistentního stavu do nového konzistentního stavu. [32]
- **Isolation** (izolace) vyžaduje, aby pokud probíhá více transakcí najednou, nebyly zároveň probíhající transakce vzájemně ovlivněny. Probíhající transakce nemůže přistupovat k mezidatům jiných probíhajících transakcí. [32]
- **Durability** (trvanlivost / odolnost) zaručuje, že data z úspěšně provedené transakce nebudou ztracena. Trvanlivost je zajištěna pomocí zálohování databáze a transakčních protokolů, které usnadňují obnovení odevzdaných transakcí navzdory jakýmkoli následným selháním softwaru nebo hardwaru. [32]

Eventuální konzistence je alternativa k ACID modelu. Využívá se primárně u distribuovaných nebo decentralizovaných databází za účelem dosažení vysoké dostupnosti. Transakce je u tohoto systému považována za úspěšnou při aktualizaci dat na většině nodů. Mezi takové databáze patří např. databáze Cassandra, nebo DynamoDB.

4.2.1 Definice grafové databáze

Hlavním rozdílem grafových (síťových) databází oproti relačním databázím je způsob ukládání dat. Data v grafové databázi jsou uložena jako uzly a hrany [33]. Uzly představují entity a hrany znázorňují relace mezi entitami. Hrana vždy spojuje právě dva uzly. Uzly i hrany mohou mít vlastnosti. Grafová databáze byla zvolena, protože aplikace generuje mnoho vysoce propojených dat, která lze následně vyhodnocovat a provádět na základě nich např. doporučování příspěvků v reálném čase. Grafové databáze jsou často využity právě pro analýzu a práci s propojenými daty. [34][35]

4.2.2 Neo4j

V projektu je použita databáze Neo4j. Databáze byla zvolena zejména kvůli její aktivní komunitě a detailní dokumentaci. Neo4j je používána také některými světovými organizacemi např. NASA, Airbnb, Lyft a Ebay. Neo4j splňuje všechny zásady ACID, čímž je garantovaná stálost dat v databázi. [36]

4.2.3 Constraints

Constraints je sada omezení při práci s daty. Lze definovat omezení na základě vlastností vrcholu (např. nelze vytvořit entitu „User“ bez vlastnosti „username“), na základě relací (např. nemůže existovat entita „Post“ bez relace „CREATED“ s entitou „User“, nemůže existovat relace „FOLLOW“ mezi entitou „User“ a entitou „post“), na základě datových typů (např. vlastnost „email“ entity „User“ musí vždy být typu string).

4.2.4 Dotazovací jazyk Cypher Query Language

Cypher je výchozí dotazovací jazyk pro Neo4j. Syntaxí je podobný jazyku SQL. Cypher Query Language je jedinečný, díky použití syntaxe typu ASCII-art (dotaz je vizuálně podobný vzoru, kterého se týká), pro uzly používá kulaté závorky a šipky pro relace. Například dotaz odpovídající všem uživatelům, kteří vytvořili alespoň jeden příspěvek by tedy vypadal takto: „MATCH (:User)-[:CREATED]->(:Post)“ [37]

4.2.5 Datová struktura - seznam entit a jejich možných relací

Podle konvencí Neo4j jsou všechny entity pojmenovány s velkým písmenem na začátku a relace se všemi písmeny velkými, pokud se jedná o víceslovný název, jednotlivá slova jsou oddělena podtržítkem. [38] Všechny entity mají vlastnosti „createdAt“ obsahující čas jejich vytvoření a „id“ (unikátní identifikátor).

User (uživatel)

Má relaci „CREATED“ s každou entitou, kterou vytvořil (např. „Comment“, „Post“, „Collection“). Může mít také relaci „FOLLOW“ s ostatními uživateli a relaci „REPORT“ s příspěvky, které nahlásil.

Vlastnosti:

- **username** (uživatelské jméno) - každé uživatelské jméno musí být unikátní bez ohledu na malá a velká písmena; může obsahovat pouze písmena bez diakritiky, čísla a podtržítka
- **email** - obsahuje unikátní email uživatele
- **firstName** - křestní jméno uživatele
- **lastName** - příjmení uživatele (tato vlastnost je nepovinná)
- **bio** - vlastní popis uživatele
- **googleID** - je vyplněno pouze pokud použije uživatel registraci pomocí Google účtu, jedná se uživatelské id poskytnuté Google API
- **verified** - pokud má hodnotu *true*, jedná se o ověřeného uživatele; uživatel je automaticky ověřený, pokud je zaregistrován pomocí Google účtu, jinak je nutné potvrdit ověřovací email; pokud uživatel neověří svůj účet do 3 dnů, bude smazán
- **passwordResetToken** (token pro reset hesla) - tato vlastnost je pouze u uživatelů, kteří zapomněli své heslo; token je z bezpečnostních důvodů platný pouze 24 hodin od vytvoření a je zaslán pomocí emailu
- **authorizationToken** (autorizační token) - je odeslán na uživatelem zadaný email a následně použit při ověření uživatele
- **isAdmin** - pokud se jedná o administrátorský účet, má hodnotu *true*
- **password** - ukládá hash hesla vygenerovaný pomocí algoritmu bcrypt

Post (příspěvek)

Může mít relaci „BELONGS_TO“ s entitou „Photo“ a relaci „IS_LOCATED“ s entitou „Place“. Entita „User“ s ní může mít relaci „LIKE“.

Vlastnosti:

- **description** - uživatelem zadaný text příspěvku
- **historické datum** - datum kdy byla historická fotografie pořízena podle formátu popsaného v kapitole 6.2

Photo (fotografie)

Každý použitý obrázek je uložen jako entita „Photo“.

Vlastnosti:

- **index** - pokud příspěvek obsahuje více obrázků, budou seřazeny vzestupně podle této vlastnosti
- **blurhash** - obsahuje vygenerovaný řetězec znaků viz kapitola 5.3
- **hash** obsahuje unikátní identifikátor souboru viz 4.1.1
- **width** - šířka obrázku v pixelech, spolu s výškou („height“) je použita pro lepší vykreslení blurhashe na frontendu
- **height** - výška obrázku v pixelech
- **nsfw** - pokud se jedná od NSFW fotografii její hodnota je *true*; popis vyhodnocování fotografií je popsán v kapitole 5.1.1

Place (místo)

Každé místo na mapě je uloženo jako entita „Place“. Může mít relaci „HAS_PREVIEW“ s entitou „Photo“, všechny příspěvky týkající se tohoto místa s ním mají relaci „IS_LOCATED“.

Vlastnosti:

- **location** - obsahuje souřadnice uložené jako typ point [39]
- **name** - název místa, společně s vlastností „description“ může být změněno pouze uživateli s administrátorským oprávněním
- **description** - popis místa
- **icon** - významná místa mohou mít vlastní ikonu, která se poté zobrazuje na mapě; tato vlastnost obsahuje IPFS hash souboru s ikonou

Comment (komentář)

Má relaci „BELONGS_TO“ s entitou „Post“.

Vlastnosti:

- **content** - samotný text komentáře
- **edited** - výchozí hodnota je *false*, pokud uživatel svůj komentář změní, je nastavena na *true*, pro informaci ostatních uživatelů (komentáře reagující na změněný komentář by se po změně mohly zdát nerelevantní)

Collection (kolekce)

Má relaci „CONTAINS“ se všemi příspěvky, které jsou v ní uloženy.

Vlastnosti:

- **name** - název kolekce
- **description** - uživatelem vytvořený popis
- **isPrivate** - pokud má tato vlastnost hodnotu *true*, kolekci bude mít možnost vidět pouze uživatel, který ji vytvořil, nebo uživatelé s administrátorským oprávněním, v opačném případě bude přístupná komukoliv

5 OBRÁZKY

5.1 ZPRACOVÁNÍ PŘI NAHRÁNÍ

5.1.1 Filtrování nevhodného obsahu

Pro kontrolu NSFW obsahu bylo použito API třetí strany (Rapid API). Toto API bylo zvoleno z důvodu vysoké kvality vytrénovaného modelu a tudíž velké úspěšnosti rozpoznání nevhodného obsahu a malé míry falešně pozitivních výsledků (tzv. false positive). Příspěvky obsahující fotografii klasifikovanou jako NSFW jsou vytvořeny, ale do databáze jsou uloženy s vlastností „nsfw“ nastavenou na *true*. Pokud se takový příspěvek má zobrazit uživateli, nejdříve je zobrazena cenzurovaná verze s varováním, po jehož přijmutí lze zobrazit fotografii bez cenzury.

5.1.2 Nahrání a validace souboru

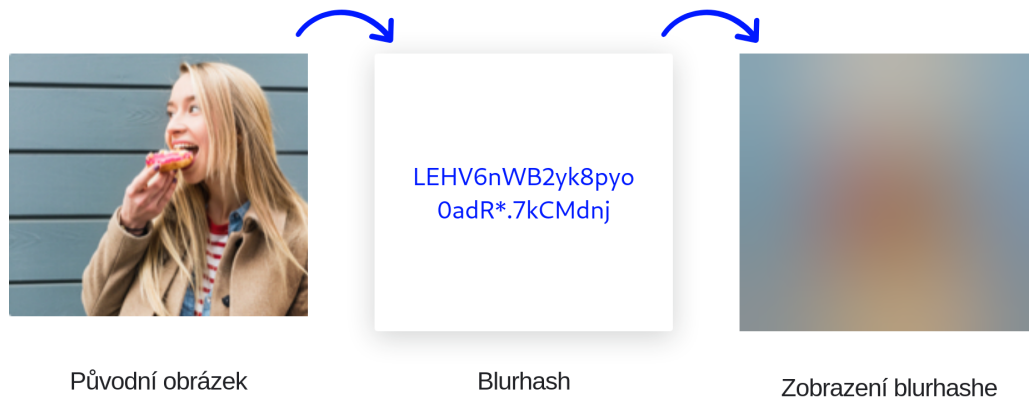
Maximální velikost nahraného souboru je 20MB, při maximálním počtu najednou nahrávaných souborů na 5. Po nahrání je na backendu ověřeno, zda se jedná o formát obrázku (např. formáty PNG, JPG, GIF), poté je převeden do formátu JPG. Pokud nejdelší strana obrázku přesahuje 2560px, obrázek je zmenšen při zachování poměru stran tak, aby jeho nejdelší strana měla 2560px a nahrán na IPFS. Následně je vygenerován blurhash, který je uložen do databáze.

5.2 OPTIMALIZACE A POUŽITÍ KOMPONENTY IMAGE

Next.js poskytuje React komponentu Image, která umožňuje využívat optimalizaci obrázků (tzn. zmenšení jejich rozlišení a kvality výměnou za rychlejší načtení). V argumentech komponenty lze nastavit kvalitu obrázku a tím zmenšit datový objem nutný doručit klientovi. Zároveň obrázek není přenášen v plném rozlišení, ale rozlišení je přizpůsobeno velikosti, ve které je obrázek vykreslen na obrazovce klienta. Optimalizované obrázky jsou cachovány na serveru, díky čemuž jsou následně poskytovány instantně. [40]

5.3 BLURHASH

Blurhash je formát vyvinut a využíván společností Wolt. Umožňuje vygenerování krátkého řetězce znaků z obrázku. Řetězec znaků lze použít pro vykreslení rozmazané verze fotografie, která je následně vykreslena na místě fotografie před jejím načtením. To zajišťuje lepší uživatelskou zkušenost a pocit rychlejšího načítání obrázků. [41][42]



Obrázek 5.1: Generování a zobrazení blurhashe z původního obrázku [43]

6 STRUKTURA PROJEKTU

6.1 OPRÁVNĚNÍ UŽIVATELŮ

Oprávnění uživatelů se dělí na následující čtyři kategorie:

Nepřihlášení uživatelé nejsou nijak omezeni v zobrazování obsahu (kromě soukromých kolekcí). Nemají však žádnou možnost přidávání, úpravy ani odebrání, či nahlašování obsahu.

Neověření přihlášení uživatelé bez administrátorských oprávnění mohou sledovat ostatní uživatele a označovat příspěvky jako oblíbené, nemají možnost přidávat žádný obsah ani nahlašovat příspěvky ostatních uživatelů. Za ověřeného uživatele je považován uživatel zaregistrovaný s Google účtem, nebo uživatel, který potvrdil vlastnictví zadané emailové adresy skrz ověřovací email. Pokud uživatel nepotvrdí svou emailovou adresu přes potvrzovací email, po třech dnech bude jeho účet smazán.

Ověření přihlášení uživatelé bez administrátorských oprávnění mohou zobrazovat veškerý obsah (kromě soukromých kolekcí ostatních uživatelů), vytvářet, upravovat a odstraňovat vlastní příspěvky, komentáře a kolekce. Mohou také označovat příspěvky jako oblíbené, přidávat a odebírat příspěvky z vlastních kolekcí a sledovat ostatní uživatele. Na stránce nastavení, mohou měnit údaje týkající se jejich účtu jako jsou například uživatelské jméno, křestní jméno, příjmení atd. Mohou také nahlašovat cizí příspěvky, jedná-li se o nevhodný obsah.

Přihlášení uživatelé s administrátorským oprávněním mají všechny možnosti jako standardní uživatelé rozšířené o možnost označit fotografii v příspěvku jakéhokoliv uživatele jako NSFW, měnit název, popis, umístění a náhled místa a mazání příspěvků, nebo komentářů ostatních uživatelů, pokud usoudí, že se jedná o nevhodný obsah. Administrátorský účet musí vždy být označen jako ověřený.

6.2 HISTORICKÉ DATUM PŘÍSPĚVKU

Každý příspěvek má datum, kdy byl vytvořen, a historické datum. Historické datum je datum kdy byl pořízen originál fotografie nahrané v příspěvku (zpravidla se jedná o starší data). U starších dat je nutné počítat s tím že, se může jednat pouze o odhad, proto není možné použít datum ve formátu podle standardu ISO 8601, jak je zvykem. V navržené struktuře se počítá s tím, že je znám alespoň rok, jako dodatečné informace lze přidat i měsíc a případně den. Druhá možnost je zadání časového období (například červen až červenec roku 1984). Proto jsou

uložené vlastnosti příspěvku rozděleny na počáteční a konečné datum. V případě že se jedná o jedno datum, jsou počáteční a konečné datum stejné. V praxi jde o šest vlastností, které má každý příspěvek: „startDay“, „startMonth“, „startYear“, „endDay“, „endMonth“, „endYear“.

6.3 UŽIVATELSKÉ VSTUPY

6.3.1 Rozšířené možnosti formátování v textových vstupech

Následující možnosti jsou možné ve všech textových vstupech určených pro delší text (např. popis uživatelského účtu - „bio“, popis příspěvku, nebo komentáře)

Označit uživatele je možné zadáním znaku zavináče následovaným uživatelským jménem označeného uživatele. Uživatelské jméno bude poté v textu zvýrazněno modrou barvou (pokud se jedná o uživatelské jméno přihlášeného uživatele, bude označeno žlutou barvou) a po kliknutí na něj bude uživatel přeměrován na profil označeného uživatele. Pokud se změní uživatelské jméno označeného uživatele, všechna označení starého uživatelského jména budou nahrazena novým.

Zvýraznit text lze buď ztučněním, nebo podtržením. Pokud je část textu ohraničená dvěma hvězdičkami, bude vykreslen jako tučný text (např. „**Pardubice** jsou město“ bude vykresleno jako „**Pardubice** jsou město“), pokud bude část textu označena dvěma podtržítka, bude vykreslena kurzívou (např. „Pardubické závodiště (__hipodrom__)“ bude vykresleno jako „Pardubické závodiště (*hipodrom*)“).

6.3.2 Validace vstupů

Všechny textové vstupy zadané uživatelem je nutné validovat (tj. ověřovat jestli neobsahují nechtěné znaky nebo řetězce znaků). Pokud by vstupy nebyly validovány mohl by uživatel např. získat přístup k databázi, nebo přepisovat obsah stránky (pokud by nebyl použit framework, který tuto chybu ošetřuje) vložením škodlivého textu.

GraphQL

GraphQL provádí základní validaci vstupu ověřením jeho typu (uživatel na místo čísla nemůže zadat text).

Validace textových vstupů

Uživatelské jméno - musí být unikátní, může obsahovat malá a velká písmena bez diakritiky, čísla a podtržítka.

Popisky příspěvků a komentáře - mohou obsahovat jakékoliv znaky, jsou omezeny pouze délkou. Všechny speciální znaky jsou escapovány pro zabránění injectu do databáze.

Souřadnice

Při vytváření příspěvku jsou jeho souřadnice validovány podle standardu ISO 6709.

Soubory

Middleware omezuje maximální počet najednou nahrávaných souborů na 5 při maximální velikosti jednoho souboru 20MB.

Historické datum

- **Validace přesného data** - pokud je uživatelem zadáno přesné datum, je pro ověření jeho správnosti použita funkce JavaScriptu pro rozpoznávání dat
- **Validace data s chybějícím dnem nebo měsícem** - pokud je vyplněn pouze den a rok, den bude nastaven jako null z logických důvodů
- **Validace roku** - minimální povolený hodnota je 1400, v případě potřeby ji lze změnit v konfiguračním souboru. Maximální hodnota je vždy aktuální rok
- **Validace měsíce** - měsíc je zadán jako číslo od 1 do 12
- **Časové rozmezí** - pokud datum je konce rozmezí starší než datum začátku jsou tyto dvě hodnoty prohozeny; pokud je datum začátku a konce shodné, jedná se o přesné datum (nejedná se o rozmezí), tento formát je validní

7 FRONTEND

Next.js poskytuje file-system routing (přesměrování na základě souborového systému), v praxi to znamená, že soubory nacházející se ve složce *pages*, nebo v jejích podsložkách jsou automaticky dostupné na odpovídající URL adrese.

7.1 STRUKTURA REACT KOMPONENT

Struktura React komponent byla inspirována „atomickým designem“. „Atomický design“ funguje na bázi rozdělení všech komponent na pět základních kategorií: „atomy“, „molekuly“, „organismy“, „šablony“ a „stránky“. Tato struktura zaručuje přehlednost a jednotnost i u rozsáhlých projektů. [44]

- „**atomy**“ (atoms) - nejzákladnější stavební prvky stránky (např. tlačítka, textové vstupy a nadpisy); nemohou být dále rozděleny při zachování stejné funkčnosti [44]
- „**molekuly**“ (molecules) - sestávají z několika atomů uspořádaných do jednoduchého UI, tím může být např. textový vstup s tlačítkem potvrdit, které dohromady tvoří vyhledávací formulář [44]
- „**organismy**“ (organisms) - jsou relativně složitější UI komponenty skládající se ze skupin „molekul“ a „atomů“, např. vrchní lišta s navigací a vyhledáváním [44]
- „**šablony**“ (templates) - umisťují komponenty („organismy“, atd.) do rozvržení stránky a určují základní strukturu návrhu [44]
- „**stránky**“ (pages) - naplňují „šablony“ reálnými daty; jedná se o soubory uložené v již zmíněné složce *pages* a jejích podsložkách [44]



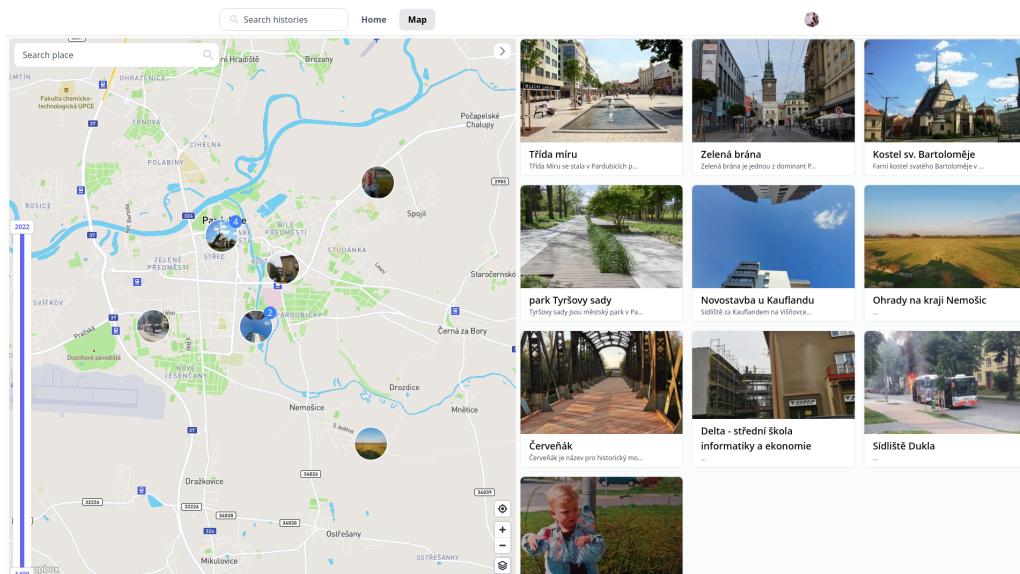
Obrázek 7.1: Diagram „atomického designu“ [45]

7.2 LOKALIZACE

Lokalizace je přizpůsobení textů, formátů dat, měny atd. na stránce (nebo v aplikaci) tak aby splňovaly jazykové a kulturní požadavky cílené lokality. V praxi to znamená že v kódu jsou použity pouze klíče (zkratky) jednotlivých slov, slovních spojení, nebo vět, ke kterým je následně podle souboru s překlady přiřazena odpovídající hodnota (text). Může být použit jeden soubor s překlady pro jeden jazyk, nebo být dále rozdělen na několik souborů (např. soubor s chybovými hláškami, s vysvětlivkami a ostatní). V projektu byl vytvořen soubor pro překlad do angličtiny a češtiny. Tyto soubory se nacházejí ve složce *translation/languages*. Jazyk zobrazení je vybrán podle jazyku prohlížeče, pokud se jedná o nepřihlášeného uživatele, v opačném případě je nastaven podle uživatelova nastavení. [46]

7.3 STRÁNKA S MAPOU

Stránka s mapou je výchozí stránka celé webové aplikace (viz obrázek 7.2). Pro zobrazení mapy je použit balíček *react-map-gl*, který umožňuje zobrazování např. markerů (značek), polygonů a 3D terénu. Pro zobrazení konkrétních stylů mapy je použit tiling service. Mapa je rozdělena do virtuální mřížky (jedno pole je jeden tzv. tile), pro každý tile je poté ze serveru poslán obrázek korespondující s konkrétní částí mapy (některé tiling services používají místo rastrové grafiky grafiku vektorovou, což snižuje nároky na síť), server posílá několik tilů zároveň, ty jsou následně cachovány u klienta. Díky tomu není nutné stahovat data pro jednu část mapy vícekrát. V projektu byl jako tiling service použit Mapbox, jedná se o placené řešení s možností použití již vytvořených stylů, nebo vytvořením stylu zcela vlastního. Alternativou je např. Leaflet, který používá open-source projekt Open street maps. Jeho nevýhodou je malé množství stylů bez možnosti tvorby vlastního.



Obrázek 7.2: Stránka s mapou

7.3.1 Zobrazení webové stránky

Základní zobrazení stránky je rozděleno na pravou a levou část.

Levá část zobrazuje mapu s ovládacími prvky. V levém horním rohu je textový vstup pro vyhledání místa na mapě podle textu. Levý spodní roh obsahuje posuvník znázorňující časovou osu, který umožňuje omezení zobrazených míst a příspěvků podle historického data fotografií, které obsahují. V pravém spodním rohu jsou tlačítka pro přiblížení a oddálení mapy, přesunutí na aktuální polohu zařazení a menu pro změnu vzhledu mapy. Možné vzhledy mapy jsou světlý, tmavý a satelitní. V pravém horním rohu se nachází tlačítko umožňující skrýt pravou část a zobrazit mapu přes celou obrazovku.

Pravá část zobrazuje náhledy všech míst, která se nacházejí v oblasti mapy a která obsahují příspěvky s historickým datem odpovídajícím filtrování podle historického data. Toto zobrazení lze přepnout na zobrazení všech příspěvků, které splňují zmíněná kritéria. Při výběru místa se namísto seznamu míst otevře detail vybraného místa obsahující jeho náhled, název, popis a seznam příspěvků obsahující fotografie daného místa.

7.3.2 Geolokace klienta

Pokud parametry URL adresy neobsahují souřadnice na kterých by měla být mapa zobrazena, stránka si vyžádá přístup ke geolokaci. Pokud je přístup udělen, automaticky je lokace mapy nastavena na souřadnice poskytnuté prohlížečem.

7.3.3 Ukládání pozice na mapě do parametrů URL adresy

Při každém pohybu po mapě jsou souřadnice aktuální pozice mapy uloženy do parametrů URL adresy. Díky tomu při obnovení stránky pozice mapy zůstává nezměněna. Tato funkcionality byla inspirována např. Mapy.cz a Google maps. Toto také umožňuje uživateli sdílet aktuální zobrazení mapy zkopírováním URL adresy.

7.3.4 Načítání míst na mapě

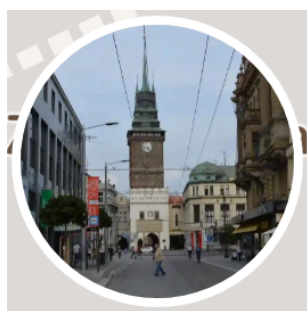
Vzhledem k tomu že množství míst na mapě není teoreticky nijak omezeno, musí být načítána postupně. Pokud by ale byla načítána nová místa při každém pohybu, nebyl by pohyb po mapě plynulý a z toho důvodu téměř nepoužitelná (kvůli množství dotazů na API). Tento problém byl vyřešen tím, že dotaz na API je odeslán po dokončení pohybu. Pro zmenšení objemu přenesených dat každý takový požadavek obsahuje seznam míst, která byla již dříve načtena a nacházejí se v zobrazené oblasti, tato místa jsou vynechána z odpovědi serveru.

7.3.5 Zobrazení míst na mapě

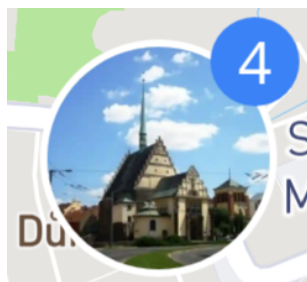
Místo na mapě je zobrazeno jako fotografie v kruhu, nacházející se na souřadnicích daného místa viz obrázek 7.3. Pro optimalizaci mapy jsou vykreslovány pouze místa nacházející se v zobrazené oblasti.

7.3.6 Seskupování míst na mapě

Při oddálení mapy by se v případě většího počtu zobrazených míst mohla daná místa překrývat, snižovat přehlednost mapy a zpomalovat pohyb po mapě. Z toho důvodu je použit tzv. clustering (shlukování více bodů), místo těchto bodů je následně vykreslen pouze jeden, který se nachází na souřadnicích průměru souřadnic všech bodů, které cluster obsahuje viz obrázek 7.4. Pro rozhodnutí, který obrázek bude na mapě zvolen jako náhled clusteru jsou porovnány vzdálenosti všech míst, která cluster obsahuje od středu clusteru. jako náhled je použit náhled místa, které je nejbližší středu clusteru. V pravém vrchním rohu náhledu je poté zobrazeno číslo vyjadřující počet míst seskupených v clusteru. Při kliknutí na náhled clusteru je mapa přiblížena tak aby bylo zobrazeno co nejvíce míst jako samostatná místa a ne clustery.



Obrázek 7.3: Zobrazení jednoho místa na mapě

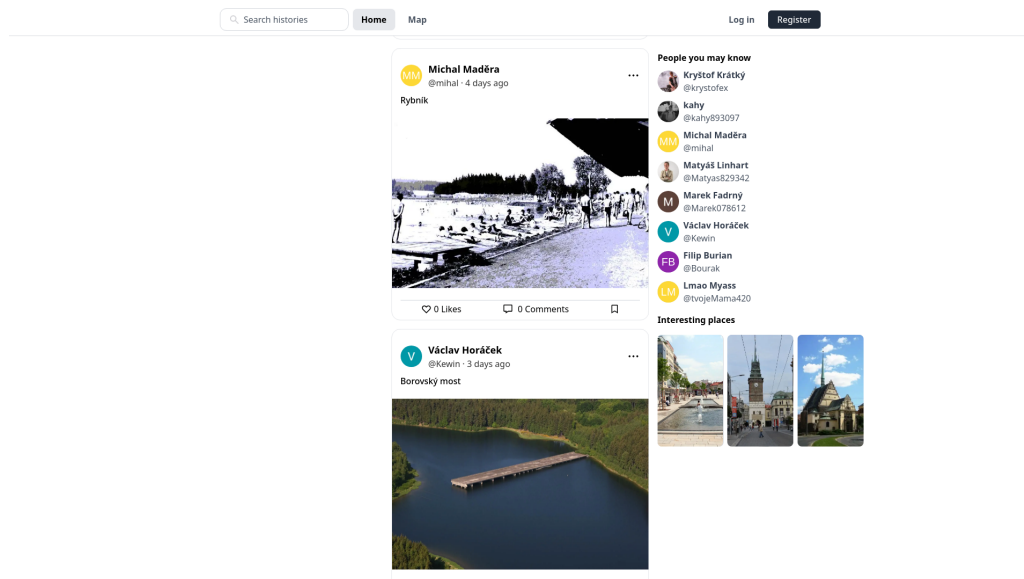


Obrázek 7.4: Zobrazení seskupení několika míst na mapě

7.4 STRÁNKA DOMŮ

Pro lepší představu je tato stránka ukázána na obrázku 7.5. Stránka domů slouží k prohlížení nových příspěvků od uživatelů, které uživatel sleduje. Zároveň obsahuje sekci doporučení uživatelé a zajímavá místa, které uživateli umožňují objevovat nové uživatele a místa. Pokud uživatel není přihlášen, jsou zde zobazeni nejpopulárnější uživatelé a nejoblíbenější příspěvky.

Pro nejlepší UX (user experience) je použit infinite scroll (nekonečné rolování), které zajišťuje dostatek obsahu a průběžné načítání dalších příspěvků, které mají být zobrazeny.



Obrázek 7.5: Domovská stránka

7.5 KOMENTÁŘE

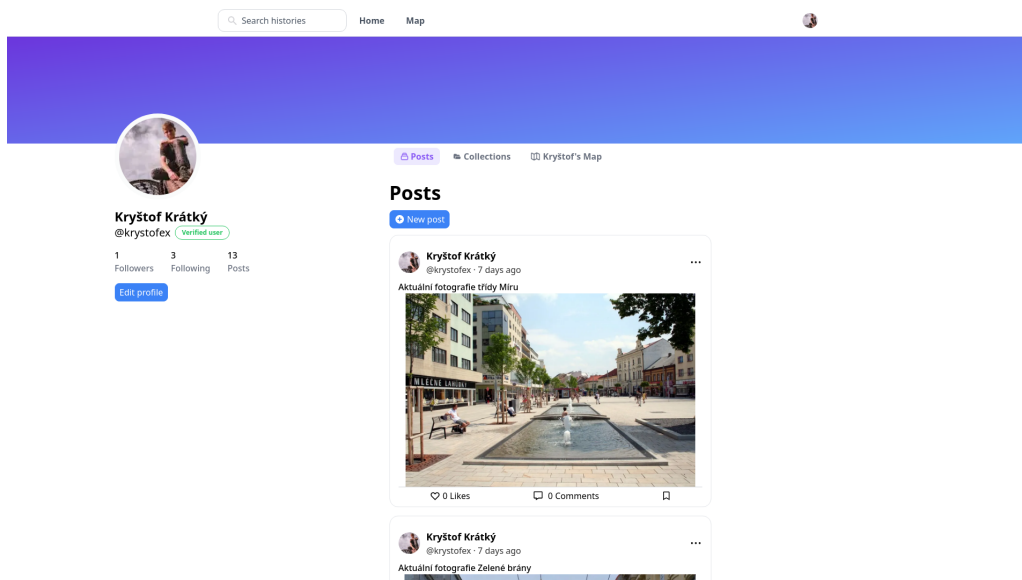
Komentáře umožňují uživateli reagovat na příspěvky a komentáře ostatních uživatelů, nebo s nimi vést diskuzi. Hloubkou komentářového vlákna je označováno maximální možné množství na sebe do sebe zanořených komentářů. V projektu je omezena na jednu úroveň (lze komentovat příspěvky a komentáře příspěvků) kvůli větší přehlednosti a intuitivnějšímu ovládní. Komentáře jsou velice důležité pro vyjadřování názorů a zapojení nových uživatelů do komunity. Zároveň jsou komentáře užitečné např. při vyhodnocování popularity příspěvků, nebo analýze komunity pomocí sociálního grafu. [47]

7.6 PROFILOVÁ STRÁNKA UŽIVATELE

Profilová stránka zobrazuje všechny informace o uživateli (viz obrázek 7.6). Tato stránka se nachází na URL adrese závislé na uživatelském jménu, např. profilová stránka uživatele s uživatelským jménem „Joe“ by se nacházela na adrese `/user/joe` (nerozlišují se velká a malá písmena v uživatelském jméně). Hlavní rozložení stránky je rozděleno na levý panel zabírající přibližně čtvrtinu obrazovky a hlavní obsah zabírající zbytek obrazovky. Ve vrchní části hlavního obsahu je navigační lišta, ve které je možné přepínat mezi třemi záložkami, podle kterých se mění obsah zobrazený v hlavní části.

7.6.1 Levý panel

Levý panel zobrazuje veškeré informace o daném uživateli. Ve vrchní části je zobrazena uživatelská profilová fotografie. Pod profilovou fotografií je uživatelské jméno a příjmení. Dále je zde uživatelské jméno, vedle kterého se zobrazují tzv. bannery. Mezi možné bannery patří např. „ověřený uživatel“ (viz 6.1), „nový uživatel“ (je zobrazen po dobu tří dnů od založení účtu), nebo „admin“ (viz 6.1)



Obrázek 7.6: Profilová stránka uživatele

7.6.2 Záložka příspěvky

Jednu se o nejdůležitější záložku. Na této záložce jsou zobrazeny všechny příspěvky, které uživatel vytvořil seřazený od nejnovějšího po nejstarší. Každý příspěvek je zobrazen v kartě. Karta se skládá ze tří hlavních částí: vrchní panel, fotografie a spodní panel.

Vrchní panel

V levé vrchní části vrchního panelu jsou informace o autorovi příspěvku, jako je jeho jméno, uživatelské jméno a profilová fotografie. Vedle uživatelského jména je zobrazeno, před jak dlouhou dobou byl příspěvek vytvořen. Pod těmito informacemi se nachází uživatelský popis. V pravé části vrchního panelu je tlačítko pro zobrazení více možností. Po kliknutí na tlačítko více možností se otevře menu obsahující různé možnosti podle typu uživatele.

- **Nepřihlášení uživatelé** vidí možnosti „zobrazit detail příspěvku“ a „zkopírovat odkaz na příspěvek“
- **Přihlášení uživatelé**, kteří zároveň nejsou autorem příspěvku vidí kromě stejných možností, jako uživatelé nepřihlášení také možnost příspěvek nahlásit.

- **Autor příspěvku** vidí kromě stejných možností jako nepřihlášený uživatel také možnosti „smazat příspěvek“ a „upravit příspěvek“
- **Administrátor** administrátor vidí stejné možnosti jako autor příspěvku rozšířené o možnost „označit příspěvek jako NSFW“, případně „odznačit příspěvek jako NSFW“

Fotografie

Pod vrchním panelem se nachází část s fotografií. Pokud příspěvek obsahuje více fotografií, je možné se mezi nimi přesouvat pomocí šipek na levé a pravé straně. Při kliknutí na fotografii je otevřeno dialogové okno, které umožňuje si fotografii detailněji prohlédnout.

Spodní panel

V levé části spodního panelu je tlačítko *to se mi líbí*. Pokud je uživatel přihlášen, může označit příspěvek jako to se mi líbí. U takto označeného příspěvku je tlačítko vyplněno červenou barvou, v opačném případě je tlačítko zobrazeno pouze jako obrys. Vedle tohoto tlačítka je zobrazen celkový počet uživatelů, kteří tento příspěvek označili jako to se mi líbí. V prostřední části spodního panelu je zobrazena ikona komentáře společně s celkovým počtem komentářů k příspěvku. Po kliknutí na ikonu komentáře je uživatel přesměrován na detail příspěvku, kde je možné zobrazit jednotlivé komentáře, nebo napsat vlastní komentář. V pravé části spodního panelu se nachází ikona záložky. Po kliknutí na ikonu záložky je otevřeno dialogové okno, ve kterém je možné přidat příspěvek do jakékoliv z kolekcí přihlášeného uživatele.

7.6.3 Záložka kolekce

Na této stránce jsou zobrazeny všechny uživatelské kolekce, ke kterým má přístup aktuální uživatel (pokud má přihlášený uživatel administrátorská oprávnění, jsou zobrazeny všechny kolekce, jinak jsou zobrazeny pouze veřejné kolekce). Kolekce je znázorněna jako karta, na které je název kolekce, její popis a počet příspěvků, které kolekce obsahuje. Po kliknutí na kolekce je zobrazen detail kolekce, ve kterém jsou zobrazeny všechny příspěvky zarovnané do mřížky. Při kliknutí na jakýkoliv příspěvek je otevřen jeho detail, ve kterém je možné zobrazit předchozí, nebo následující příspěvek v kolekci. Ve vrchní části kolekce je také možnost příspěvky seřadit. Možnosti řazení jsou: od nejoblíbenějšího, od nejstaršího podle historického data, od nejnovějšího podle historického data, od nejstaršího podle času vytvoření, a od nejnovějšího, nebo nejnovějšího podle času vytvoření. Pokud je přihlášený uživatel autor kolekce, má možnost upravit jméno a popis kolekce, nebo kolekci smazat.

7.6.4 Záložka uživatelská mapa

Na této mapě jsou označena všechna místa, ke kterým uživatel přidal příspěvek. Tato mapa slouží čistě pro shrnutí a nemá nahrazovat hlavní mapu. Proto zde nejsou zobrazeny žádné fotografie ani možnost jakékoliv jiné interakce než prohlížení mapy.

7.7 STRÁNKA DETAIL PŘÍSPĚVKU

Rozložení stránky je rozděleno na pravou a levou část. Hlavním účelem levé části je zobrazení fotografií příspěvku. Nad fotografií je zobrazena ikona kalendáře společně s historickým datem. Po kliknutí na ikonu kalendáře bude uživatel přesměrován na mapu, kde je automaticky nastaveno filtrování. Pokud má příspěvek nastaveno přesné datum, omezí filtr na zobrazení všech příspěvků v rozsahu mezi 20 lety před datem příspěvku a 20 lety po datu příspěvku. Pokud je datum příspěvku stanoveno v rozmezí několika let, je filtrování nastaveno podle tohoto rozmezí. Hlavní část zabírá fotografie místa, pod kterou je tlačítko líbí se mi a pokud má příspěvek více fotografií možnost mezi nimi posouvat. V pravé části se nachází komentářová sekce.

8 VÝVOJ

8.1 NASAZENÍ NA SERVER

8.1.1 Docker

Docker je soubor platforem a služeb poskytujících virtualizaci na úrovni operačního systému pro spouštění softwaru v balíčcích nazývané kontejnery. Docker zajišťuje že programy a aplikace vždy poběží ve stejném prostředí nezávisle na zařízení nebo operačním systému. Jednoduchost spuštění kontejneru usnadňuje nasazení aplikace na server. Celý projekt včetně databáze a úložiště je pak možné spustit jedním skriptem bez nutnosti nastavování.

8.1.2 Hosting

Jako hosting je označována služba, na kterou lze nahrát svou webovou aplikaci, která následně bude přístupná na internetu. Pro hostování aplikace byla zvolena platforma DigitalOcean, z důvodu jednoduchosti administrátorského rozhraní (na rozdíl od např. AWS) a ceně. Zároveň nabízí funkce jako například automatické naklonování aplikace z Git repozitáře a následné spuštění na serveru při změně kódu. Umožňuje také pronajmutí virtuálního počítače, na kterém lze následně aplikaci spustit například v Docker kontejneru.

8.2 VERZOVÁNÍ

Git je software pro sledování změn v souborech, které jsou součástí repozitáře. Git poskytuje možnost se kdykoliv vrátit do jakékoliv verze projektu. Git lze také použít při kolaboraci více programátorů, nebo pro synchronizaci kódu.

8.3 TESTOVÁNÍ

8.3.1 Při vývoji

Unit testing je automatizované testování jednotlivých funkcí kódu zadáním vstupních parametrů a následným porovnáním výsledků s očekávanými výsledky.

Lint je označení pro skupinu nástrojů, analyzující statický kód bez nutnosti jeho spuštění. Zároveň zajišťuje jednostnost kódu kontrolou pravidel pro formátování.

9 ZÁVĚR

V rámci projektu se podařilo vyvinout webovou aplikaci umožňující uživatelům sdílet a prohlížet historické fotografie. V současné době má uživatel možnost vytvářet vlastní příspěvky, prohlížet příspěvky ostatních uživatelů, ukládat příspěvky do kolekcí a interagovat s ostatními uživateli pomocí komentářů. Veškerý nahraný obsah prochází kontrolou proti nevhodnému obsahu. Aplikace je zcela připravena pro použití reálnou komunitou, na základě jejíž zpětné vazby bude moci být dále vyvíjena a upravována. Funkční verze aplikace je dostupná na adrese <https://www.histories.cc>.

SEZNAM OBRÁZKŮ

5.1	Generování a zobrazení blurhashe z původního obrázku [43]	14
7.1	Diagram „atomického designu“ [45]	18
7.2	Stránka s mapou	19
7.3	Zobrazení jednoho místa na mapě	21
7.4	Zobrazení seskupení několika míst na mapě	21
7.5	Domovská stránka	22
7.6	Profilová stránka uživatele	23

LITERATURA

- [1] NSFW Meaning, what does NSFW mean in text *myenglishteacher.eu* [online]. [cit. 2022-03-19]. Dostupné z: <https://www.myenglishteacher.eu/blog/nsfw-meaning/>
- [2] The Very Basics of Garbage Collection *basenoru.se* [online]. [cit. 2022-03-19]. Dostupné z: <http://basen.oru.se/kurser/koi/2008-2009-p1/texter/gc/index.html/>
- [3] What Does Runtime Environment (RTE) Mean *Techopedia* [online]. [cit. 2022-03-19]. Dostupné z: <https://www.techopedia.com/definition/5466/runtime-environment-rte>
- [4] Dynamic typing vs. static typing *Oracle docs* [online]. [cit. 2022-03-19]. Dostupné z: https://docs.oracle.com/cd/E57471_01/bigData.100/extensions_bdd/src/cext_transform_typing.html
- [5] What is typescript *typescripittutorial.net* [online]. [cit. 2022-03-19]. Dostupné z: <https://www.typescripittutorial.net/typescript-tutorial/what-is-typescript/>
- [6] TypeScript for the New Programmer *Typescript docs* [online]. [cit. 2022-03-19]. Dostupné z: <https://www.typescriptlang.org/docs/handbook/typescript-from-scratch.html/>
- [7] What is npm *W3 schools* [online]. [cit. 2022-03-19]. Dostupné z: https://www.w3schools.com/whatis/whatis_npm.asp
- [8] About npm *NPM docs* [online]. [cit. 2022-03-19]. Dostupné z: <https://docs.npmjs.com/about-npm>
- [9] Introduction to GraphQL *GraphQL* [online]. [cit. 2022-03-19]. Dostupné z: <https://graphql.org/learn/>
- [10] Schemas and Types *GraphQL* [online]. [cit. 2022-03-19]. Dostupné z: <https://graphql.org/learn/schema/#type-system>
- [11] What is REST API? *Red Hat* [online]. [cit. 2022-03-19]. Dostupné z: <https://www.redhat.com/en/topics/api/what-is-a-rest-api/>
- [12] What is an API? *Amazon AWS* [online]. [cit. 2022-03-19]. Dostupné z: <https://aws.amazon.com/what-is/api/>

- [13] Frontend vs. backend: what's the difference? *Pluralsight* [online]. [cit. 2022-03-19]. Dostupné z: <https://www.pluralsight.com/blog/software-development/front-end-vs-back-end/T>
- [14] Components and Props *React docs* [online]. [cit. 2022-03-19]. Dostupné z: <https://reactjs.org/docs/components-and-props.html>
- [15] Getting started *React docs* [online]. [cit. 2022-03-19]. Dostupné z: <https://reactjs.org/docs/getting-started.html>
- [16] Client-side vs. Server-side vs. Pre-rendering for Web Apps *Toptal* [online]. [cit. 2022-03-19]. Dostupné z: <https://www.toptal.com/front-end/client-side-vs-server-side-pre-rendering>
- [17] What Is SEO *Next.js docs* [online]. [cit. 2022-03-19]. Dostupné z: <https://moz.com/learn/seo/what-is-seo>
- [18] Gettings started *Next.js docs* [online]. [cit. 2022-03-19]. Dostupné z: <https://nextjs.org/docs/getting-started>
- [19] Build fast, responsive sites with Bootstrap *Bootstrap* [online]. [cit. 2022-03-19]. Dostupné z: <https://getbootstrap.com/>
- [20] Get started with Tailwind CSS *Tailwind CSS* [online]. [cit. 2022-03-19]. Dostupné z: <https://tailwindcss.com/docs/installation/>
- [21] Introduction to GraphQL Code Generator *GraphQL code generator docs* [online]. [cit. 2022-03-19]. Dostupné z: <https://www.graphql-code-generator.com/docs/getting-started>
- [22] About Node.js® *Node.js®* [online]. [cit. 2022-03-19]. Dostupné z: <https://nodejs.org/en/about/>
- [23] IPFS is a distributed system for storing and accessing files, websites, applications, and data *IPFS* [online]. [cit. 2022-03-19]. Dostupné z: <https://docs.ipfs.io/>
- [24] Decentralized vs Trustless Networks *Bluzelle* [online]. [cit. 2022-03-19]. Dostupné z: <https://bluzelle.com/blog/decentralized-vs-trustless-networks>
- [25] Centralized, Decentralized, & Distributed Networks *Cryptopedia* [online]. [cit. 2022-03-19]. Dostupné z: <https://www.gemini.com/cryptopedia/blockchain-network-decentralized-distributed-centralized>
- [26] Decentralized and Trustless Networks *Hackernoon* [online]. [cit. 2022-03-19]. Dostupné z: <https://hackernoon.com/decentralized-and-trustless-networks-f881671fae4e>
- [27] Content addressing and CIDs *IPFS* [online]. [cit. 2022-03-19]. Dostupné z: <https://docs.ipfs.io/concepts/content-addressing/>

- [28] Persistence | IPFS Docs *IPFS* [online]. [cit. 2022-03-19]. Dostupné z: <https://docs.ipfs.io/concepts/persistence/>
- [29] IPFS Gateway *IPFS* [online]. [cit. 2022-03-19]. Dostupné z: <https://docs.ipfs.io/concepts/ipfs-gateway/>
- [30] What is Amazon S3? *Amazon AWS* [online]. [cit. 2022-03-19]. Dostupné z: <https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html>
- [31] What Is a Database? *Oracle* [online]. [cit. 2022-03-19]. Dostupné z: <https://www.oracle.com/database/what-is-database/>
- [32] The ACID Database Model *Lifewire* [online]. [cit. 2022-03-19]. Dostupné z: <https://www.lifewire.com/the-acid-model-1019731/>
- [33] Graphentheorie *Mathopedia* [online]. [cit. 2022-03-19]. Dostupné z: <https://mathepedia.de/Graphentheorie.html>
- [34] Graph Databases for Beginners: Why Graph Technology Is the Future *Neo4j* [online]. [cit. 2022-03-19]. Dostupné z: <https://neo4j.com/blog/why-graph-databases-are-the-future/>
- [35] Graph Algorithms in Neo4j: Neo4j Graph Analytics *Neo4j* [online]. [cit. 2022-03-19]. Dostupné z: <https://neo4j.com/blog/graph-algorithms-in-neo4j-neo4j-graph-analytics/>
- [36] The Fastest Path to Graph *Neo4j* [online]. [cit. 2022-03-19]. Dostupné z: <https://neo4j.com>
- [37] Cypher Query Language *Neo4j* [online]. [cit. 2022-03-19]. Dostupné z: <https://neo4j.com/developer/cypher/>
- [38] Naming rules and recommendations - Neo4j Cypher Manual *Neo4j* [online]. [cit. 2022-03-19]. Dostupné z: <https://neo4j.com/docs/cypher-manual/4.4/syntax/naming/>
- [39] Spatial functions - Neo4j Cypher Manual *Neo4j* [online]. [cit. 2022-03-19]. Dostupné z: <https://neo4j.com/docs/cypher-manual/4.4/functions/spatial/>
- [40] API documentation for the Image Component and Image Optimization *Next.js docs* [online]. [cit. 2022-03-19]. Dostupné z: <https://nextjs.org/docs/api-reference/next/image>
- [41] BlurHash is a compact representation of a placeholder for an image *Blurhash* [online]. [cit. 2022-03-19]. Dostupné z: <https://blurha.sh/>
- [42] How we came to create a new image placeholder algorithm, BlurHash *Wolt blog* [online]. [cit. 2022-03-19]. Dostupné z: <https://blog.wolt.com/hq/2019/07/01/how-we-came-to-create-a-new-image-placeholder-algorithm-blurhash/>

- [43] Obrázek převzat z <https://blurha.sh> *Blurhash* [online]. [cit. 2022-03-19]. Dostupné z: <https://blurha.sh/>
- [44] Atomic Design Methodology *bradfrost.com* [online]. [cit. 2022-03-19]. Dostupné z: <https://atomicdesign.bradfrost.com/chapter-2/>
- [45] Diagram atomického designu *Github* [online]. [cit. 2022-03-19]. Dostupné z: <https://user-images.githubusercontent.com/4838076/33235048-d083dca6-d217-11e7-9aea-9a5ef5ae6fe7.png>
- [46] Internationalization vs. localization (i18n vs l10n): What's the difference? *Lokalise* [online]. [cit. 2022-03-19]. Dostupné z: <https://lokalise.com/blog/internationalization-vs-localization/>
- [47] Reasons Why Comments Are Important For Instagram *Time square chronicles* [online]. [cit. 2022-03-19]. Dostupné z: <https://t2conline.com/reasons-why-comments-are-important-for-instagram/>

PŘÍLOHA A ZADÁNÍ MATURITNÍHO PROJEKTU

Zadání maturitního projektu z infromatických předmětů

Jméno a příjmení: *Kryštof Krátký*
Školní rok: *2021/2022*
Třída: *3.B*
Obor: *Informační technologie 18-20-M/01*

Téma práce: aplikace pro sdílení historických fotografií
Vedoucí práce: RNDr. Jan Koupil, Ph.D.

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

Cílem projektu je webová aplikace, pro sdílení historických fotek, se zobrazením míst na mapě a možností časové osy pro zobrazení fotek z určitého období.

Věci co budu muset udělat:

- Zjištění, jestli už něco podobného existuje
- Rozhodnutí technologií, které budou použity, návrh datového modelu
- Vymyslet funkcionalitu frontendu, UI, návrh architektury systému
- Vytvoření algoritmu pro návrhy doporučených fotek

Stručný časový harmonogram (s daty a konkretizovanými úkoly):

- Září: průzkum - rozhodnutí, které technologie použiju, zjištění, jestli existuje něco podobného, vymyšlení názvu
- Říjen: Návrh datového modelu
- Listopad: Uživatelské rozhraní
- Prosinec-leden: Funkcionalita
- Únor: Jestli bude všechno hotovo algoritmus na doporučování fotek, dodělávání toho co chybí
- Březen: Testování, dokumentace