

Univerzita Pardubice  
Dopravní fakulta Jana Pernera

Program pro generování galerií fotografií ve formě webových stránek  
Martin Svoboda

Bakalářská práce  
2020

Univerzita Pardubice  
Dopravní fakulta Jana Pernera  
Akademický rok: 2019/2020

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Martin Svoboda**  
Osobní číslo: **D16066**  
Studijní program: **B3709 Dopravní technologie a spoje**  
Studijní obor: **Aplikovaná informatika v dopravě**  
Téma práce: **Program pro generování galerií fotografií ve formě webových stránek**  
Zadávací katedra: **Katedra informatiky v dopravě**

### Zásady pro vypracování

Vytvořte program pro generování galerií fotografií ve formě webových stránek. Fotografiím a jejich náhledům transformujte velikost podle parametrů zadávaných do programu a počet náhledů na stránce. V programu se bude zadávat i nadpis pro galerii. Stránky musí být bezchybné, kontrolovatelné validátorem <https://validator.w3.org>.

Rozsah pracovní zprávy: **30 normostran**  
Rozsah grafických prací:  
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. BRADY, Gavin. How to Convert an Image to JPG Format. *How-To Geek* [online]. LifeSavvy Media, ?2019, 14. 9. 2018 [cit. 2019-11-05]. Dostupné z: <https://www.howtogeek.com/362730/how-to-convert-an-image-to-jpg-format/>
2. JPG File Extension: What is a JPG file? *FileInfo: The File Extensions Database* [online]. Sharpened Productions, ?2019 [cit. 2019-10-07]. Dostupné z: <https://fileinfo.com/extension/jpg>
3. Image File Formats – All About Images: Native File Formats. *Library – University of Michigan* [online]. Ann Arbor, Michigan: University of Michigan, ?2018 [cit. 2019-10-07]. Dostupné z: <https://guides.lib.umich.edu/c.php?g=282942&p=1885348>
4. LUNDQUIST, Samuel. Image file formats: when to use each type of file. *99designs* [online]. [cit. 2019-10-07]. Dostupné z: <https://en.99designs.de/blog/tips/image-file-types>
5. Jimp – npm. *Npm* [online]. [cit. 2019-10-07]. Dostupné z: <https://www.npmjs.com/package/jimp>
6. Using Express middleware. *Express* [online]. StrongLoop, IBM, ?2017 [cit. 2019-10-06]. Dostupné z: <https://expressjs.com/en/guide/using-middleware.html>
7. Express. *Npm* [online]. [cit. 2019-10-06]. Dostupné z: <https://www.npmjs.com/package/express>
8. KATARIA, Saransh. How to setup Typescript with Babel and Webpack. *Wisdom Geek* [online]. Wisdom Geek, ?2019, 12. 2. 2019 [cit. 2019-10-06]. Dostupné z: <https://www.wisdomgeek.com/development/web-development/how-to-setup-typescript-with-babel-and-webpack>
9. The package.json guide. *Flavio Copes* [online]. Flavio Copes, ?2019, 3. 8. 2018 [cit. 2019-10-06]. Dostupné z: <https://flaviocopes.com/package-json/>
10. GRAY, Tanya. What the heck is npm? *Medium* [online]. Medium Corporation, ?2019, 2. 7. 2018 [cit. 2019-10-06]. Dostupné z: <https://medium.com/@tanya/what-the-heck-is-npm-b8168f61e3b5>
11. What is npm? *Node.js* [online]. 26.8.2011 [cit. 2019-10-06]. Dostupné z: <https://nodejs.org/en/knowledge/getting-started/npm/what-is-npm/>
12. Node.js – Quick Guide: Node.js – Introduction. *Tutorialspoint* [online]. [cit. 2019-10-06]. Dostupné z: [https://www.tutorialspoint.com/nodejs/nodejs\\_quick\\_guide.htm](https://www.tutorialspoint.com/nodejs/nodejs_quick_guide.htm)

Vedoucí bakalářské práce: **Ing. Josef Šroll, Ph.D.**  
Katedra informatiky v dopravě

Datum zadání bakalářské práce: **28. listopadu 2019**  
Termín odevzdání bakalářské práce: **31. ledna 2020**

L.S.

---

**doc. Ing. Libor Švadlenka, Ph.D.**  
děkan

---

**doc. Ing. Karel Greiner, Ph.D.**  
vedoucí katedry

V Pardubicích dne 28. listopadu 2019

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 31. 5. 2020

Martin Svoboda

## **PODĚKOVÁNÍ**

Rád bych poděkoval vedoucímu své bakalářské práce panu Ing. Josefu Šrollovi, Ph.D. za ochotu vést moji bakalářskou práci a bezproblémový průběh. Rovněž musím poděkovat svému zaměstnavateli HolodeckVR GmbH za ohleduplnost k mým studijním povinnostem.

## **ANOTACE**

Bakalářská práce má za cíl vytvořit aplikaci pro generování galerií fotografií ve formě webových stránek, popsat její NPM moduly použité při tvorbě aplikace a popsat platformu pro tvorbu serverových aplikací Node.js. Práce má rovněž za cíl popsat typy souborů, které jsou používány pro ukládání obrázků.

## **KLÍČOVÁ SLOVA**

HTML, Node.js, NPM, JavaScript, TypeScript, obrázky

## **TITLE**

Application for generating galleries as web pages

## **ANNOTATION**

The purpose of this thesis is to develop an application for generating image galleries as web pages, describe server-side application platform Node.js and its modules used in development of the application. Last but not least, describe file extensions used for saving images.

## **KEYWORDS**

HTML, Node.js, NPM, JavaScript, TypeScript, images

# OBSAH

ÚVOD .....	12
1    NODE.JS .....	13
1.1    Vlastnosti .....	13
1.2    Node.js – NPM.....	13
1.2.1    Použití NPM.....	14
1.2.2    Instalace balíčků.....	14
1.2.3    Package.JSON.....	14
2    TYPESCRIPT .....	16
2.1    Instalace .....	16
2.2    Tsconfig.json.....	16
3    ELECTRON JS .....	17
3.1    Vlastnosti .....	17
4    REACT.JS.....	18
4.1    Vlastnosti a části .....	18
4.1.1    JSX.....	18
4.1.2    React Native .....	19
4.1.3    Virtual Document Object Model.....	19
5    APLIKACE PRO GENEROVÁNÍ GALERIÍ VE FORMĚ WEBOVÝCH STRÁNEK.....	20
5.1    Frontend .....	21
5.1.1    Individuální komponenty .....	21
5.1.2    Hlavní stránka .....	28
5.1.3    Stránka s galerií.....	31
5.2    Backend.....	34
5.2.1    Hlavní okno a obsluha serveru.....	34
5.2.2    Webový server .....	34
5.3    Požadavky na systém .....	35
5.4    Instalace a spuštění.....	35
5.5    NPM moduly v projektu.....	36
5.5.1    Express .....	36
5.5.2    File System.....	38
5.5.3    Formidable .....	41

5.5.4	Jimp.....	44
5.5.5	React paginate.....	46
5.5.6	React Carousel Image Gallery.....	47
5.5.7	Styled components.....	48
5.5.8	React Flexbox Grid.....	49
5.5.9	React select.....	50
5.5.10	Font Awesome.....	51
5.5.11	Tailwind CSS.....	53
5.6	Nastavení pro vývoj aplikace.....	55
5.6.1	Frontend.....	55
5.6.2	Backend.....	56
6	OBRÁZKOVÉ TYPY SOUBORŮ..... <b>CHYBA! ZÁLOŽKA NENÍ DEFINOVÁNA.</b>	
6.1	Ztrátové formáty.....	57
6.1.1	JPG/JPEG.....	57
6.2	Bezztrátové formáty.....	58
6.2.1	GIF.....	58
6.2.2	PNG.....	59
6.2.3	TIFF.....	60
6.2.4	RAW.....	62
7	ZÁVĚR.....	63
7.1	Zhodnocení.....	63
7.2	Výstup z HTML validátoru.....	63
7.2.1	Validace úvodní stránky.....	63
7.2.2	Validace stránky s galerií.....	64
7.3	Vyjádření k poznámkám z předchozí obhajoby.....	64
8	POUŽITÁ LITERATURA.....	68



## SEZNAM OBRÁZKŮ

Obrázek 1: Komponenta TopContainer .....	22
Obrázek 2: Komponenta GalleryModal.....	23
Obrázek 3: Komponenta Footer.....	24
Obrázek 4: Hlavní stránka .....	28
Obrázek 5: Gallery Modal .....	30
Obrázek 6: Stránka s agalerií .....	31
Obrázek 7: Modální okno .....	65
Obrázek 8: Zobrazení metadat.....	66

## **SEZNAM ZKRATEK**

NPM	Noda Package Manager
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
URL	Uniform Resource Locator
HTTP	Hypertext Transfer Protocol
URI	Uniform Resource Identifier
DOM	Document Object Model
JPEG	Joint Photographic Experts Group
PNG	Portable Network Graphic
GIF	Graphics Interchange Format
TIFF	Tagged Image Format File
PX	Pixel

## SLOVNÍK POJMŮ

callback	funkce, která vstupuje jako parametr do jiné funkce
backend	proces, program běžící např. na vzdáleném serveru
frontend	část programu, kterou vidí uživatel; uživatelské rozhraní
engine	optimalizovaný software s konkrétní funkcí
render	vykreslení, zobrazení něčeho
upload	nahrání (např. souboru na server)
parse	převedení jednoho datového typu na jiný
framework	struktura usnadňující programování
open–source	software s otevřeným zdrojovým kódem
handler	funkce, která reaguje na nějakou událost v programu (např. kliknutí)
buffer	vyrovnávací paměť
tag	značka, štítek, element
desktop	aplikace, která je schopna běžet na samostatném počítači
full screen	zobrazení přes celou obrazovku
request	požadavek
response	odpověď

# ÚVOD

Aplikace, jež je výsledkem praktické části této bakalářské práce, usnadňuje prohlížení fotografií uložených na lokálním zařízení a nabízí jednoduchou možnost jejich přeformátování. Obsluha celé aplikace je prováděna přes webové rozhraní, tudíž je velmi snadná.

Aplikace využívá technologii Node.js, jež umožňuje tvorbu serverových aplikací pomocí jazyka JavaScript. Pro správnou funkcionalitu aplikace využívá open source moduly z repositáře NPM.

V práci je popsáno základní fungování aplikace, včetně ukázek kódů. Dále jsou zde zmíněny technologie použité při tvorbě aplikace a popsány typy souborů používané pro ukládání obrázků.

# 1 NODE.JS

Node.js je platforma pro tvorbu serverových aplikací založená na jazyce JavaScript. Node.js byl vyvinut Ryanem Dahlem v roce 2009 a jeho nejnovější verze je v12.16.3. Definice Node.js, jak je uvedeno v oficiální dokumentaci, je následující – „Node.js je platforma postavená na běhovém prostředí Chrome v jazyce JavaScript (Chrome’s JavaScript runtime) pro snadnou tvorbu rychlých a škálovatelných síťových aplikací. Node.js využívá neblokující I/O model.“ [30]

Node.js je open–source, multiplatformní prostředí, aplikace Node.js mohou být spuštěny na platformách OS X, Microsoft Windows a Linux. [30]

## 1.1 Vlastnosti

Node.js se vyznačuje dvěma klíčovými vlastnostmi. Mezi ně patří asynchronnost a řízení událostmi, jednoduchost a škálovatelnost.

### **Asynchronnost a řízení událostmi**

Všechny knihovny Node.js jsou asynchronní a neblokující. V praxi to znamená, že Node.js nikdy nečeká na návrat dat z volání funkce, ale data jsou přístupná v callback funkci. [30]

### **Jednoduchost a škálovatelnost**

Node.js používá jednovláknový model se smyčkou událostí. Vzhledem k tomu, že server na události reaguje neblokujícím způsobem, je celý model vysoce škálovatelný. To v praxi znamená, že server je schopen poskytovat služby většímu počtu požadavků. [30]

## 1.2 Node.js – NPM

NPM (zkratka pro Node Package Manager) je on-line repositář pro publikování Node.js open–source projektů. Stejně tak lze tomuto pojmu rozumět jako příkazovému řádku, který slouží pro interakci s repositářem a pomáhá při instalaci či správě jednotlivých NPM balíčků. Pod slovem balíček se může skrývat celá knihovna nebo pouze konkrétní funkce napsaná v jazyce JavaScript, TypeScript, či v některém z frameworků založeném na těchto jazycích (např. React nebo Angular). Na NPM je publikováno nepřehledné množství Node.js knihoven a aplikací. Balíčky je možno procházet na adrese <https://npmjs.com>. [50]

### 1.2.1 Použití NPM

K použití balíčku NPM je nutno mít v projektu soubor s názvem package.json. Tento soubor obsahuje seznam všech NPM balíčků, které jsou v projektu použity. Soubor package.json je možno vytvořit přes příkazový řádek, a to příkazem „npm init“. Nainstalované balíčky je poté v projektu možno importovat funkcí require, např. require('fs').

### 1.2.2 Instalace balíčků

K instalaci balíčku slouží příkaz „npm install *název balíčku* [--save]“. Balíčky mohou být instalovány buď lokálně nebo globálně. V prvním případě se balíčky instalují do složky, ve které je uložen projekt, ve druhém případě se instalují na jedno konkrétní místo, které je společné pro celý operační systém v systému. [23]

### 1.2.3 Package.JSON

Soubor package.json je manifestem pro každý Node.js projekt. Obsahuje metadata o projektu jako například jméno projektu, autor, skripty nebo popis. [47]

Některé z atributů souboru package.json:

- name
  - Určuje název projektu.
  - Menší než 214 znaků.
  - Nesmí obsahovat mezery.
- author
  - Obsahuje informace o autorovi či autorech.
  - Může obsahovat další atributy jako jméno, email či url.
- contributors
  - Obsahuje pole s informacemi o spolupracovnících na projektu.
- bugs
- homepage
- version
  - Číslo verze ve formátu X.X.X, kde X je celé, nezáporné číslo.
- licenc,
- keywords
  - Obsahuje pole klíčových slov souvisejících s balíčkem.
- description

- repository
  - Informace o umístění balíčku.
- main
  - Určuje vstupní bod pro daný balíček; název skriptu, který zpřístupňuje funkce balíčku.
- dependencies
  - Obsahuje seznam jiných balíčků, které jsou balíčkem/aplikací používány.
- engines
  - Říká, jaká verze enginů je potřebná ke spuštění aplikace. [47]

## 2 TYPESCRIPT

TypeScript je open–source programovací jazyk vyvinutý v roce 2012 společností Microsoft. TypeScript je založen na jazyce JavaScript, na rozdíl od něj umožňuje explicitně deklarovat datové typy proměnných. Kód napsaný v TypeScript se kompiluje na JavaScript. Program je tedy možno bez omezení spustit v prohlížeči, nebo v případě použití tohoto jazyka pro backend vývoj, v běhovém prostředí Node.js např. jako webový server nebo desktopovou či mobilní aplikaci. [27]

### 2.1 Instalace

K instalaci slouží příkaz „`npm install typescript`“. V případě, že je TypeScript nainstalován globálně, je možno jej použít pro kterýkoli další projekt bez nutnosti další instalace. Pro konfiguraci kompilace slouží soubor `tsconfig.json`. [27]

### 2.2 Tsconfig.json

Do Node.js projektu napsaného v jazyce TypeScript je nutno do kořenové složky umístit konfigurační soubor `tsconfig.json` pro TypeScript kompilér. V tomto souboru je možno specifikovat konfiguraci kompilace. Například které soubory s příponou `*.ts` mají být kompilovány a které mají být vyňaty. [27]



## 3 ELECTRON JS

Electron JS je runtime framework, který umožňuje vytvářet multiplatformní desktopové aplikace pomocí HTML5, CSS a JavaScriptu. Jedná se o open-source platformu, kterou vytvořil Cheng Zhao v roce 2013 ve spolupráci se společností GitHub. [14] [49]

Electron je spojením dvou populárních technologií: Node.js a Chromium. Takže jakákoli webová aplikace, může být spuštěna přes Electron JS. [14] [9]

### 3.1 Vlastnosti

Vlastnosti frameworku Electron JS jsou:

- výkon
  - Aplikace vytvořené pomocí Electronu mají srovnatelný výkon s nativními desktopovými aplikacemi. [9]
- správa kódu
  - Aplikace mají stejný kód pro různé platformy. Odpadá tím starost o údržbu několika různých projektů pro každou platformu zvlášť. [9]
- znovuvyužití komponent
  - Kód pro Electron aplikaci je psán v jazyce JavaScript nebo v některém z jeho frameworků. To znamená, že individuální komponenty je poté možné využít ve standardní Node.js aplikaci (např. webový server nebo konzolová aplikace). [9]
- snadný vývoj
  - Pro aplikace je možno využívat moduly ze stále se rozrůstajícího repozitáře npm. [9]
- uživatelské rozhraní
  - Uživatelské rozhraní je možno vytvářet pomocí HTML, CSS a JavaScriptu, stejně tak například pomocí frameworků, jako jsou React.js nebo Angular. To dává možnost vytvářet moderní, uživatelsky přívětivé rozhraní s celou řadou funkcí, efektů či animací. [9]

## 4 REACT.JS

React.js je open source javascriptová knihovna, která se používá k vytváření uživatelských rozhraní, převážně pro jednostránkové aplikace. React také umožňuje vytvářet opakovaně použitelné komponenty uživatelského rozhraní. React poprvé vytvořil Jordan Walke, softwarový vývojář ze společnosti Facebook. Vůbec poprvé byl v produkci React použit v roce 2011 ve webovém rozhraní Facebooku.

Silnou stránkou Reactu je schopnost reagovat na rychle se měnící data a renderování těchto změn bez nutnosti znovunačtení celé stránky. Tento framework může rovněž být použit s kombinací jiných knihoven nebo frameworků JavaScriptu, jako je Electron JS. Možné je rovněž použití Reactu pro tvorbu multiplatformních mobilních aplikací, známé jako React Native. [32]

### 4.1 Vlastnosti a části

Zde jsou popsány esenciální vlastnosti a části frameworku React.js.

#### 4.1.1 JSX

V Reactu se místo standardního JavaScriptu používá JSX, který umožňuje použít HTML tagy k renderování dílčích komponent. V souborech s příponou JSX je ale stále možné používat standardní JavaScript.

##### Příklad JSX

Příklad vytvoření `div` elementu, který má šířku 50 % a je v něm vnořen `p` element, který zobrazuje text obsažený v proměnné `text`:

##### Kód v JavaScriptu (verze ES6 a vyšší)

```
const parentEl = document.createElement('div');
const childEl = document.createElement('p');
const text = "Some text";

childEl.innerText = text;
parentEl.style.width = '50%';

parentEl.appendChild(childEl);
```

### **Kód v JSX za použití HTML tagů**

```
const text = 'Some Text';  
  <div style={{ width: '50%' }}>  
    <p>{text}</p>  
  </div>
```

#### **4.1.2 React Native**

React Native je rámec pro vytváření mobilních aplikací používající pouze Javascript. React Native byl vytvořen Facebookem v roce 2015 a používá stejný design jako React.js, což umožňuje vytvářet znovupoužitelné komponenty uživatelského rozhraní pro mobilní aplikace. Využívá stejné základní stavební bloky jako běžné aplikace pro iOS a Android. [32]

#### **4.1.3 Virtual Document Object Model**

React pro porovnání změn v HTML kódu vytváří Virtual Document Object Model (virtuální DOM), neboli jakousi virtuální podobu HTML dokumentu. Virtual DOM je datová struktura typu k-cestný strom. Vždy, když v dokumentu dojde ke změně, vytvoří React nový Virtual DOM a prohlídkou do šířky jej porovná s tím stávajícím. Nalezne-li porovnávací algoritmus změnu, vyrenderuje element, nebo pouze část elementu (např. atribut), jehož se daná změna týká. V praxi to znamená, že je React schopen velmi efektivně a rychle renderovat změny v dokumentu právě proto, že je schopen detekovat změny každého jednotlivého elementu či atributu. [32]

## 5 APLIKACE PRO GENEROVÁNÍ GALERIÍ VE FORMĚ WEBOVÝCH STRÁNEK

V této části je popsána samotná aplikace, respektive její dvě hlavní komponenty: frontend a backend.

Zdroje modulů při tvorbě aplikace:

- Font Awesome: [1] [19]
- App root path: [2]
- Axios: [3]
- Bluebird: [6]
- Body parser: [7]
- Dotenv: [11]
- Electron Compile: [13]
- Express: [17]
- File system: [18]
- Formidable: [20]
- Jimp: [25]
- Lodash: [28]
- React: [38]
- Open: [31]
- React flexbox grid: [35]
- React image gallery: [36]
- React modal: [37]
- React paginate: [39]
- React router dom: [40]
- React scripts: [41]
- React select: [42]
- React styled flexboxgrid: [43]
- Styled components: [44]
- Electron: [14]
- Electron builder: [12]
- Tailwind: [46]

Zdroj pozadí frontendu aplikace: [10]

Zdroj ikony aplikace: [21]

## **5.1 Frontend**

Frontend je vytvořen pomocí frameworku React.js a skládá se z dvou stránek. Úvodní stránka umožňuje uživateli nahrání nových fotografií a zadání metadat potřebných pro jejich další zpracování. Druhá stránka aplikace je samotná galerie umožňující prohlížení fotografií, jejich mazání a stahování.

### **5.1.1 Individuální komponenty**

Frontend aplikace obsahuje několik individuálně vytvořených komponent. Mezi tyto komponenty patří například komponenty pro vytvoření stylovaného `nav` anebo `footer` elementu. Ukázky kódu jsou zjednodušené, je například vyňat atribut `className` umožňující zadat CSS třídy pro stylování komponent.

## TopContainer

TopContainer vytváří nav HTML element. Mimo jiné umožňuje zadat nadpis stránky přes parametr `title`, když tento parametr není zadán, je na místo nadpisu použita defaultní hodnota Gallery.

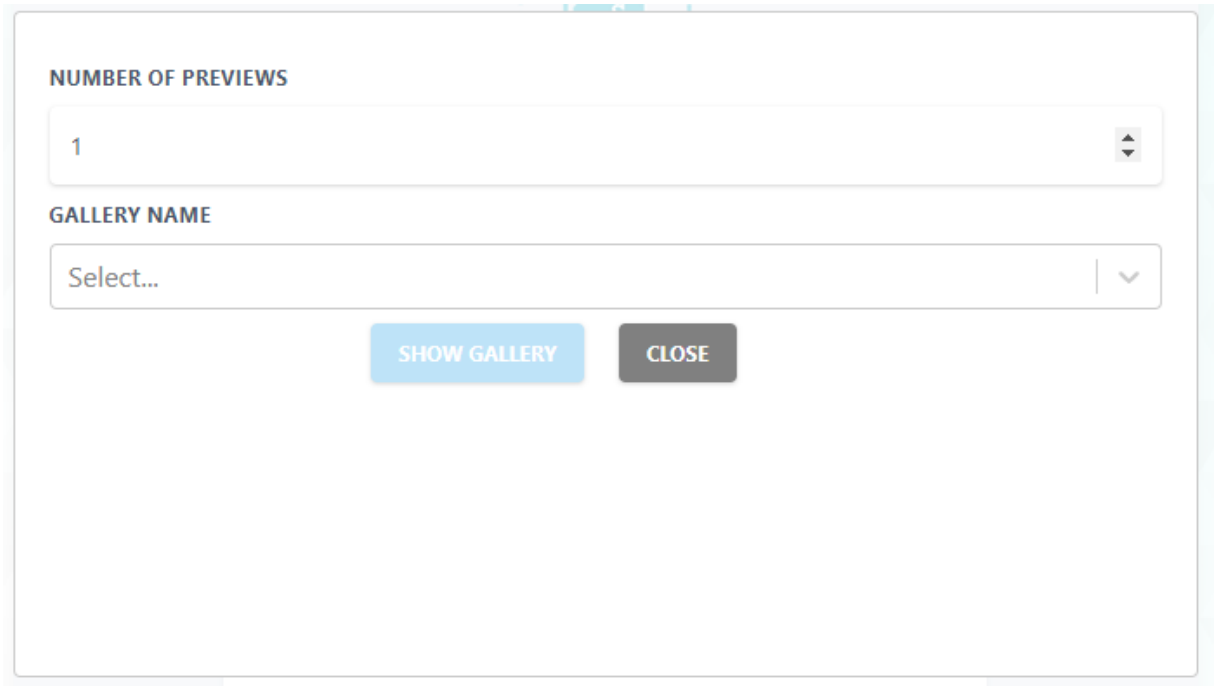


Obrázek 1: Komponenta TopContainer

```
const TopContainer = ({ title }) => {
  return (
    <nav
      style={{ background: 'rgba(255,255,255,0.5)' }}>
      <Row>
        <Col xs={8}>
          <p>
            // jestliže není zadán parametr title, bude vyrendrován text
            'Gallery'
            {title || 'Gallery'}
          </p>
        </Col>
        <Col xs={4}>
          Author: Martin Svoboda
        </Col>
      </Row>
    </nav>
  );
}
```

## GalleryModal

Tato komponenta vytváří modální okno, které nabízí uživateli možnost vybrat galerii, kterou si přeje zobrazit. Samotné modální okno je vytvořeno pomocí modulu react-modal. Pro výběr galerie slouží modul Select, který umožňuje vytvořit stylovaný select HTML element.



Obrázek 2: Komponenta GalleryModal

```
<Modal
// atribut určující, zdali má být okno otevřeno či ne
  isOpen={open}
  style={{
    content: {
      width: '50%',
      height: '50%',
      top: '25%',
      left: '25%'
    }
  }}>
</Modal>
```

## Footer

Footer vytváří `footer` HTML element umístěný v zápatí stránky. Přijímá parametr `children`, což je element, či pole elementů, které jsou potomky elementu `footer`.



Obrázek 3: Komponenta Footer

```
const Footer = ({ children }) => (  
  <footer>  
    <div>  
      // horizontální čára  
      <hr/>  
      <Row middle='xs'>  
        <Col xs={8}>  
          // elementy, které vstoupily do funkce jako parametr  
          {children}  
        </Col>  
        <Col xs={4}>  
          Martin Svoboda  
        </Col>  
      </Row>  
    </div>  
  </footer>  
)
```



## Button

Button vytváří stylovaný button HTML element. Obsahuje mimo jiné parametr `disabled`, jenž tlačítku odebere funkčnost a změní jeho styl.

```
const Button = ({ onClick, type, text, disabled, style }) => {
  return (
    <button
      type={type || 'button'}
      onClick={!disabled && onClick}
      style={{ transition: "all .15s ease", ...style }}
    >
      {text}
    </button>
  );
};
```

## Checkbox

Checkbox vyvábí stylovaný `input` HTML s typem `checkbox`. Komponenta používá modul `styled-components` k vytvoření základní designu.

Zdroj použitého kódu: [5]

```
const Styled = styled.div`attributes`;

const Checkbox = ({ checked, onChange, label }) => {
  return (
    <Styled
      onClick={() => onChange(!checked)}
    >
      <input
        type="checkbox"
        checked={checked}
      />
      <label>{label}</label>
    </Styled>
  );
}
```

## Field

Komponenta Field slouží k vytváření `input` a `label` HTML elementů pro použití ve formuláři.

```
const Field = ({ id, label, props }) => {
  return (
    <div>
      // když je zadán parametr label, dojde k vytvoření label elementu
      {label && <label for={id}>
        {label}
      </label>}
      <input id={id} {...props} />
    </div>
  )
}
```

### 5.1.2 Hlavní stránka

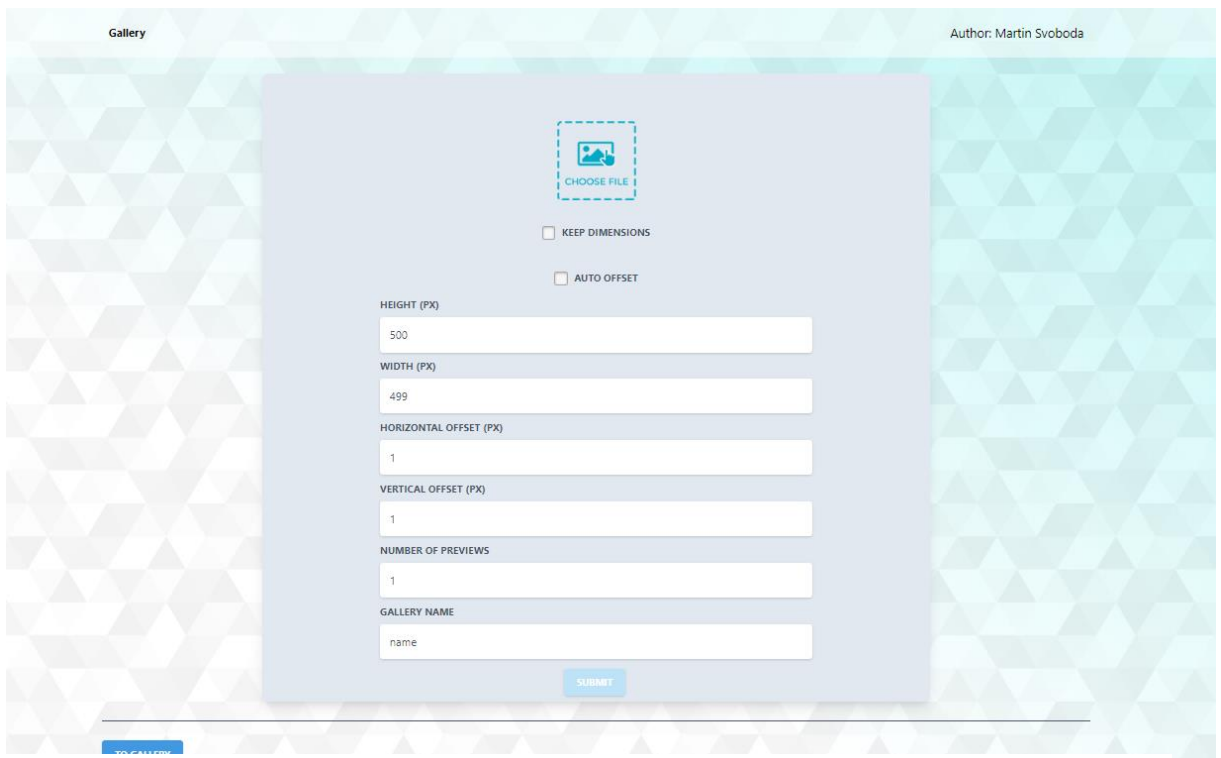
Hlavní stránka aplikace obsahuje formulář pro nahrání fotografií a jejich přidání do galerie. Uživatel vybere obrázky k přidání, zadá nové rozměry fotografií, počet náhledů na jednu stránku a název galerie. Je také možné zvolit i možnost pro zachování rozměrů fotografií.

Uživatel má při nahrávání fotografií možnost zaškrtnout volbu „auto offset“, která zajistí to, že fotografie bude oříznuta rovnoměrně podél okrajů. Například při nahrání fotky, která má na šířku 200 px a zadání nové šířky 100 px, automaticky se ubere 50 px z každého okraje. Pokud bude nahrána ta samá fotografie s šířkou 200 px, zadána nová šířka 100 px a uživatel zvolí horizontální odsazení (x offset) na hodnotu 80 px, z fotky se odřízne 80 px z levého okraje a 20 px z pravého okraje. Nulové souřadnice pro horizontální a vertikální odsazení se nachází v levém horním rohu fotografie.

Data je možné odeslat, až když jsou vybrané nějaké fotografie ke zpracování a je zadán název galerie, do které se mají obrázky přidat.

Po odeslání dat se obrázky přetransformují, uloží a dojde k přesměrování do galerie samotné.

Je možno přejít rovnou do vybrané galerie. Uživatel si vybere z listu galerií, kterou si přeje prohlížet a ta je mu poté zobrazena.



The screenshot shows a web interface for a gallery application. At the top left, it says "Gallery" and at the top right, "Author: Martin Svoboda". The main content is a light blue form with a dashed box containing a "CHOOSE FILE" button. Below this are two checkboxes: "KEEP DIMENSIONS" and "AUTO OFFSET", both of which are unchecked. The form contains several input fields: "HEIGHT (PX)" with the value "500", "WIDTH (PX)" with "499", "HORIZONTAL OFFSET (PX)" with "1", "VERTICAL OFFSET (PX)" with "1", "NUMBER OF PREVIEWS" with "1", and "GALLERY NAME" with "name". A "SUBMIT" button is located at the bottom right of the form.

Obrázek 4: Hlavní stránka

## Komponenty hlavní stránky

Hlavní stránka obsahuje znovupoužitelné komponenty `GalleryModal`, `TopContainer`, `Button` a `Footer`. `GalleryModal` zde slouží k zobrazení modálního okna s výběrem galerie, kterou si uživatel přeje zobrazit.

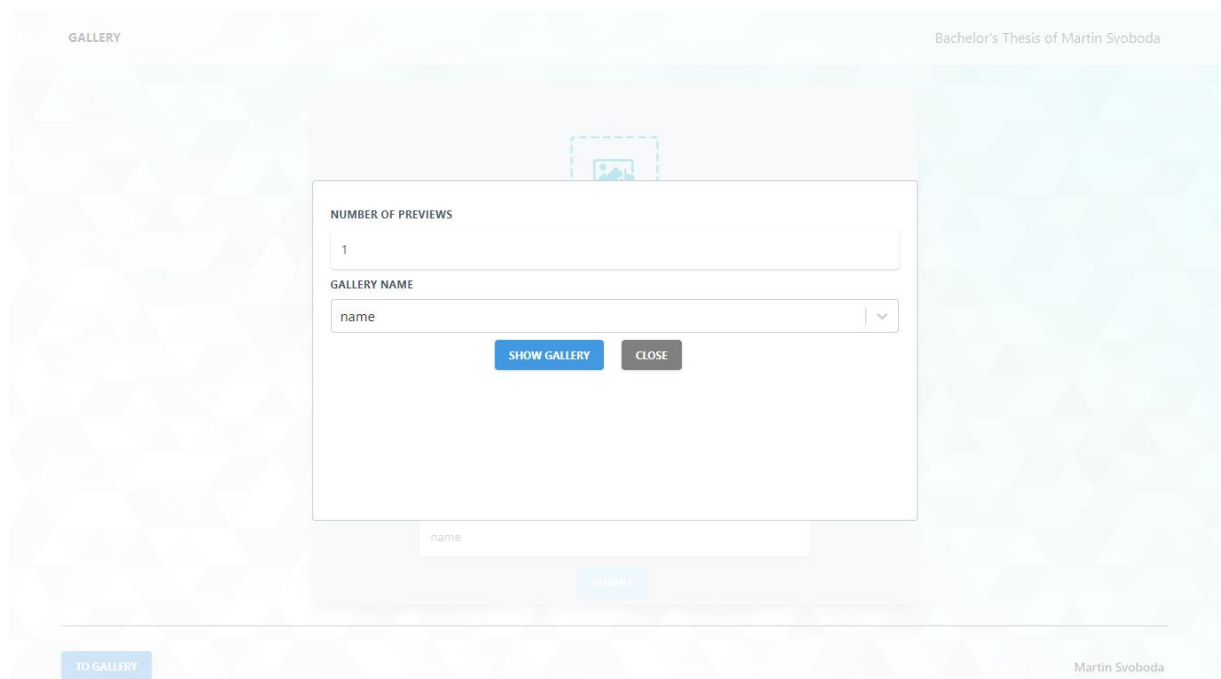
Hlavní funkcionalitu stránky je formulář, kde je možno nahrát obrázky. Formulář je stylován komponentami v modulu `react-flexbox-grid`. Pole formuláře jsou potom tvořena vlastní komponentou `Field`.

V komponentě `Field` je možné nastavit parametry jako `id`, `type` nebo `callback` funkce pro události `onChange` nebo `onBlur`.

```
<Field
  label='HEIGHT (px)'
  type="number"
  name="height"
  id="height"
  min="1"
  value={height}
  onChange={event => this.setChange('height',
event.target.value)}
  onBlur={event => this.setBlur('height', event.target.value)}
/>
```

Při odesílání formuláře dojde k vytvoření těla formuláře pro POST request, tedy odeslání na server.

```
// odeslání formuláře na server
// formData je instance třídy FormData
    axios.post(`${global.FILE_UPLOAD}`,
        formData
    )
        .then(() => {
// přesměrování na stránku s galerií
            history.push({
                pathname: '/gallery',
                state: {
                    previews,
                    title
                }
            });
        });
    });
```



Obrázek 5: Gallery Modal

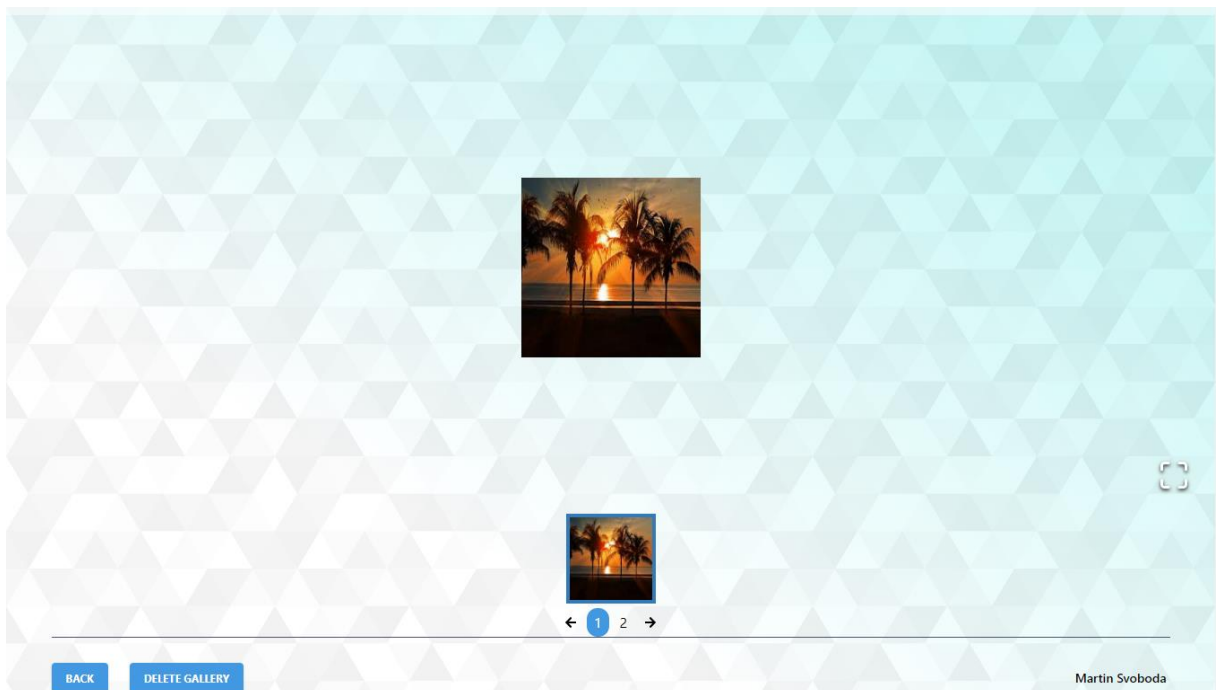
### 5.1.3 Stránka s galerií

Stránka s galerií zobrazuje vybranou galerii. Obrázky galerie jsou rozděleny do stránek podle toho, jaké číslo uživatel zadal jako počet pohledů. Pokud se všechny obrázky mají zobrazit na jedné stránce, kontejner s čísly stránek se nezobrazí.

V uživatelském rozhraní se dále nachází list náhledů obrázků. Při kliknutí na náhled se zobrazí náhled ve větší velikosti a při kliknutí na tlačítko se obrázek zobrazí v originální velikosti.

#### Komponenty stránky s galerií

Stránka s galerií se skládá z komponent `TopContainer`, `Footer` a `Button`. Galerie samotná je vytvořena pomocí modulu `react-image-gallery`.



Obrázek 6: Stránka s galerií

Implementace modulu `react-image-gallery`:

```
// filtrování obrázků k zobrazení na základě zadaného počtu náhledů  
a počtu stran
```

```
const lastIndex = (currentPage + 1) * perPage;
```

```
const firstIndex = lastIndex - perPage;
```

```
const currentImages = images.slice(firstIndex, lastIndex);
```

```
<ImageGallery
```

```
    showPlayButton={false}
```

```
    items={currentImages}
```

```
>
```



Implementace modulu `react-paginate` vytvářející stránkování:

```
// zobrazení stránkování pouze pokud je počet stran větší než 1
{pageCount > 1 &&
<ReactPaginate
  previousLabel='<'
  nextLabel='>'
// nastavení počtu stran
  pageCount={pageCount}
  onPageChange={this.handlePageClick}
  forcePage={currentPage}
// CSS třída pro odkaz s číslem stránky
  pageLinkClassName={linkClass}
  previousLinkClassName={linkClass + ' transition duration-150'}
  nextLinkClassName={linkClass}
/>}
```

## 5.2 Backend

Backend tvoří desktopová aplikace, která je zároveň webovým serverem, přes nějž se obsluhuje samotná aplikace pro generování galerií ve formě webových stránek. Aplikace je vytvořena pomocí frameworku Electron.JS.

### 5.2.1 Hlavní okno a obsluha serveru

Elementární částí backendu aplikace je script vytvářející okno s uživatelským rozhraním a kontrolující webový server.

Vytvoření okna a načtení uživatelského rozhraní ze souboru index.html:

```
import { BrowserWindow } from 'electron';

const window = new BrowserWindow({
  width: 600,
  height: 240,
  maximizable: false,
  resizable: false
});

window.loadFile('src/frontend/index.html');
```

Webový server je vytvářen pomocí funkce listen z modulu express. Funkce přijímá číslo portu, na kterém má server běžet, jako vstupní parametr:

```
function createServer(port?: number){
  express.set('port', port);
  const server = express.listen(express.get('port'));
}
```

### 5.2.2 Webový server

Samotný webový server umožňuje interakci se samotnou aplikací pro generování galerií. Server hostuje webový frontend aplikace, který je možné zobrazit v prohlížeči. Dále poskytuje rozhraní pro získávání obrázků uložených na serverové složce.

V neposlední řadě poskytuje koncové body a s ním spojené funkce pro upload, transformování, mazání a stahování obrázků a galerií.

Zjednodušené příklad funkce, která mění rozměry obrázků:

```
// načtení obrázku, který má být zpracován
    Jimp.read(pathToImage)
    .then((image: NameJimp) => {
        image.name = value.name;
        if (height && width) {
// změna rozměrů a uložení obrázků
            image
                .resize(width, height)
// uložení
                .write(path.join(dirName, title, image.name));
        }
    });
```

Funkce pro čtení souborů ze složky:

```
const images = fs.readdirSync(nazevSlozky);
const result = images.map((item) => {
    // kód pro práci s přečtenými obrázky (vynechán)
});
```

### 5.3 Požadavky na systém

Pro spuštění je nutné disponovat operačním systémem Windows 10. K zobrazení webového rozhraní je nutné mít nainstalován Google Chrome nebo Microsoft Edge.

### 5.4 Instalace a spuštění

Aplikace se instaluje spuštěním souboru Setup.exe. Ten aplikaci automaticky nainstaluje na určené místo v systému a rovněž vytvoří zástupce na ploše.

Po úspěšné instalaci je možné aplikaci spustit otevřením souboru „photogallery.exe“. Otevře se okno aplikace, které hostí webový server. Poté, co je server spuštěn, se otevře prohlížeč na příslušné URL, nebo je možné jej spustit manuálně a zadat příslušnou adresu (defaultně `http://localhost`).

## 5.5 NPM moduly v projektu

Aplikace pro generování galerií ve formě webových stránek používá některé moduly z repozitáře NPM. Podrobněji popsány jsou nejdůležitější z nich.

### 5.5.1 Express

Express je open-source framework usnadňující vývoj webových aplikací v Node.js. Jeho výhody spočívají zejména ve zjednodušení zpracování HTTP požadavků směrem k serveru a jejich směrování, stejně tak jako dynamického renderování HTML pohledů. Základním konceptem je cyklus požadavek-odpověď (request-response), což znamená, že na každý požadavek od klienta server vyšle nějaký typ odpovědi nebo sekvenci odpovědí. [17]

#### Instalace

K instalaci slouží příkaz „`npm install express`“. [17]

#### Funkce směrování

Směrování určuje způsob, jakým aplikace reaguje na uživatelský požadavek na konkrétní koncový bod, který je určen cestou (URI) a specifickým druhem HTTP požadavku GET, POST, PUT nebo DELETE. Podle druhu požadavku dojde k volání příslušné metody, popřípadě metody `all` ke zpracování všech druhů požadavků. [15]

Směrování metody obsahují jako parametr callback funkci, jež je volána, když aplikace obdrží požadavek na určitý směr. [15]

#### Směrovací metody

Směrovací metody mají vždy následující syntaxi: `app.functionName(path, handler)`

- „`app`“ je instance třídy `express`.
- „`functionName`“ je typ metody zpracovávající http požadavek.
- „`path`“ je koncový bod, na který směřuje požadavek.

- „handler“ je funkce, která je volána po přijetí požadavku.

příklady:

- Metoda, která odpovídá na GET požadavek s výchozím směrem (/).
  - `app.get('/', function (req, res) {`  
   `// nějaký kód`  
   `})`
- Metoda, která odpovídá na POST požadavek s výchozím směrem (/).
  - `app.post('/', function (req, res) {`  
   `// nějaký kód`  
   `}) [16]`

### Směrovací cesty

Směrovací cesty určují koncové body, na které může směřovat http požadavek. Směrovací cesty mívají podobu řetězce znaků. Mohou být použity i regulární výrazy, a to za pomoci znaků: „?, +, \* a ()“. [16]

Příklady použití směrovacích cest:

- Cesta, která nasměruje požadavek na výchozí směr „/“.
  - `app.get('/', function (req, res) {`  
   `// nějaký kód`  
   `})`
- Cesta, která nasměruje požadavek na směr „/stranka“.
  - `app.get('/stranka', function (req, res) {`  
   `// nějaký kód`  
   `})`
- Cesta, která nasměruje požadavek na směr „/acd“ nebo „/abcd“.
  - `app.get('/ab?cd', function (req, res) {`  
   `// nějaký kód`  
   `})`
- Cesta, která nasměruje požadavek na směr „/a(b)cd“, přičemž (b) zastupuje samotný znak b, nebo znak b libovolněkrát se opakující (např.: „abcd“, „abbc“, „abbbcd“).
  - `app.get('/ab+cd', function (req, res) {`

```
// nějaký kód
```

```
})
```

- Cesta, která nasměruje požadavek na směr „/a(b)cd“, přičemž (b) zastupuje samotný znak b, nebo znak b s libovolným přípojným řetězcem znaků (např.: „abcd“, „abycd“, „abtextcd“).
  - `app.get('/ab*cd', function (req, res) {`  
`// nějaký kód`  
`})`
- Cesta, která nasměruje požadavek na směr „/abe“ nebo „/abcde“c.
  - `app.get('/ab(cd)?e', function (req, res) {`  
`// nějaký kód`  
`})` [16]

### Směrovací parametry

Směrovací parametry jsou pojmenované segmenty URL, z nichž je možno dostat hodnoty na základě jejich pozice v adrese URL. Tyto hodnoty je posléze možné zpracovat v příslušné POST metodě, a to přes objekt `req.params`, který vždy do metody vstupuje jako parametr.

Příklady:

Při zadání cesty se vzorem `/cesta/:parametr1/:parametr2` do URL řádku jako `http://localhost:3000/cesta/1/2`, bude mít atribut `params` parametru `req` podobu `{ "parametr1" : "1", "parametr2": "2" }`. [16]

### 5.5.2 File System

File system (zkráceně fs) je modul obsahující funkce pro práci se složkami a soubory, jako například čtení, zápis či kopírování. Většina funkcí je jak synchronní, tak asynchronní. [18]

### Instalace

K instalaci slouží příkaz „`npm install file-system`“. [18]

## Funkce a příklady jejich použití

- Funkce pro práci se složkami
  - `fs.mkdir(path, [mod?], function (err) {  
// nějaký kód  
})`
    - Asynchronně vytvoří složku na zadané cestě určené parametrem `path`, funkce má přetížení, ve kterém je možné zadat mód souboru, callback funkce je volána po vytvoření složky.
  - `fs.mkdirSync(path, [mod?])`
    - Synchronně vytvoří složku na zadané cestě určené parametrem `path`, funkce má přetížení, ve kterém je možné zadat mód souboru.
  - `fs.mkdtmp(prefix, [options?], function (err, directory) {  
// nějaký kód  
})`
    - Asynchronně vytvoří dočasnou složku na místě, kde se nachází script, ze kterého je metoda volána; funkce k parametru `prefix` přidá šestici náhodných znaků, které pak tvoří název složky; funkce má přetížení s parametrem `options`, kde je možno zadat například typ kódování; callback funkce je volána po vytvoření složky.
  - `fs.mkdtmpSync(prefix, [options?])`
    - Synchronně vytvoří dočasnou složku na místě, kde se nachází script, ze kterého je metoda volána; funkce k parametru `prefix` přidá šestici náhodných znaků, které pak tvoří název složky; funkce má přetížení s parametrem `options`, kde je možno zadat například typ kódování.
  - `fs.readdir(path, [options?], function (err, files) {  
// nějaký kód  
})`
    - Asynchronně přečte všechny soubory nacházející se ve složce určené parametrem `path`; funkce má přetížení s parametrem

`options`, kde je možno zadat například typ kódování; callback funkce je volána po přečtení všech souborů ve složce a je s nimi možno dále pracovat.

- `fs.readdirSync(path, [options?])`
  - Synchronně přečte všechny soubory nacházející se ve složce určené parametrem `path`; funkce má přetížená s parametrem `options`, kde je možno zadat například typ kódování; funkce vrací názvy souborů ve složce.
- Funkce pro práci se soubory
  - `fs.open(path, flag, [mod?], function (err, fd) {  
// nějaký kód  
})`
    - Asynchronně otevře soubor určený parametrem `path`; funkce má přetížená s parametrem `mod`; callback funkce je volána po otevření souboru, je zde možné pracovat s parametry `err` a `fd`; tedy chyba a file descriptor.
  - `fs.openSync(path, flag, [mod?])`
    - Synchronně otevře soubor určený parametrem `path`; funkce má přetížená s parametrem `mod`; funkce vrací file descriptor, tedy celé číslo.
  - `fs.write(fd, text, [position?] [encoding?], function (err, written, str) {  
// nějaký kód  
})`
    - Asynchronně vepíše do souboru určeného parametrem `fd` (file descriptor) obsah proměnné `text`; do přetížení funkce je také možno zadat pozici, na kterou se má text napsat nebo kódování textu; callback funkce je volána po dokončení psaní do souboru; má parametry `err`, `written` – celé číslo informující o počtu vepsaných znaků a `str` – řetězec znaků, který byl vepsán.
  - `fs.writeSync(fd, text, [position?], [encoding?])`
    - Synchronně vepíše do souboru určeného parametrem `fd` (file descriptor) obsah proměnné `text`; do přetížení funkce je také



možno zadá pozici, na kterou se má text napsat nebo kódování textu; funkce vrací počet vepsaných znaků.

- `fs.read(fd, buffer, offset, length, position, callback)`
  - Asynchronně přečte data ze souboru určeného parametrem `fd` (file descriptor) a uloží je do proměnné `buffer`; parametr `offset` určuje pozici na kterou se data do bufferu uloží; parametr `length` určuje počet bajtů, které mají být přečteny; parametr `position` určuje pozici, od které mají být data ze souboru čtena.
- `fs.readFileSync(fd, buffer, offset, length, position, callback)`
  - Synchronně přečte data ze souboru určeného parametrem `fd` (file descriptor) a uloží je do proměnné `buffer`; parametr `offset` určuje pozici na kterou se data do bufferu uloží; parametr `length` určuje počet bajtů, které mají být přečteny; parametr `position` určuje pozici, od které mají být data ze souboru čtena. [18]

### 5.5.3 Formidable

Modul Formidable umožňuje parsování formulářů, zejména při nahrávání souborů v Node.js projektech. [20]

#### Vlastnosti

- Rychlost asi 500mb za sekundu.
- Automatické ukládání souborů a disk.
- Nízká paměťová náročnost
- Dobré možnosti zpracování chyb.

#### Instalace

K instalaci slouží příkaz „`npm install formidable`“. [20]

#### Funkce, atributy a datové typy

Nejdůležitější datové typy z modulu Formidable jsou typy `IncomingForm` a `File`.

## **Formidable.IncomingForm**

Funkce Incoming Form vytvoří novou instanci formuláře.

Atributy instance třídy Incoming Form:

- `form.encoding`
  - Nastavení kódování polí příchozího formuláře.
- `form.uploadDir`
  - Nastavení složky pro ukládání nahraných souborů.
- `form.type`
  - Nastavení atributu na „multipart“ nebo „urlencoded“.
- `form.maxFieldsSize`
  - Nastavení limitu paměti, kterou mohou využít všechna pole formuláře (kromě souborů), výchozí hodnota je 20 MB.
- `form.maxFiels`
  - Omezení počtu polí formuláře, které bude modul zpracovávat, výchozí hodnota je 1000, 0 znamená žádné omezení.
- `form.multiples`
  - Pokud je tento atribut nastaven na `true`, bude při volání metody `Parse`, bude atribut `files` obsahovat pole souborů odeslané z HTML formuláře.
- `form.bytesReceived`
  - Počet bajtů, které formulář odeslal.
- `form.parse(request, [callback])`
  - Zpracování příchozího požadavku obsahujícího data, pokud je zadán volitelný parametr `callback`, jsou pole a soubory z formuláře poslány do této callback funkce. [20]

## **Formidable.File**

Atributy datového typu File:

- `file.size`
  - Velikost nahrávaného souboru v bajtech, pokud nahrávání stále trvá, udává, kolik bajtů již bylo nahráno do paměti.
- `file.path`
  - Cesta, která určuje, kam má být soubor nahrán.

- `file.name`
  - Původní jméno souboru.
- `file.type`
  - Originální typ média.
- `file.lastModifiedDate`
  - Datum poslední úpravy. [20]

## Události

Modul `Formidable` umožňuje registrovat `handlers` na celou řadu událostí vztahujících se u práci se soubory. Výčet událostí modulu `Formidable`:

- `progress`
  - Vyvolána, když je parsována první část dat.
  - Příklad použití:
    - `form.on('progress', function (bytesReceived, bytesExpected) {`  
`// nějaký kód`  
`})`
- `field`
  - Vyvolána v okamžiku přijetí formulářového pole s hodnotou.
  - Příklad použití:
    - `form.on('field', function (name, value) {`  
`// nějaký kód`  
`})`
- `fileBegin`
  - Vyvolána, když je detekován nový soubor při nahrávání na server
  - Příklad použití:
    - `form.on('fileBegin', function (name, file) {`  
`// nějaký kód`  
`})`
- `file`
  - Vyvolána v okamžiku přijetí formulářového pole se souborem, parametr `file` je datového typu `File`.
  - Příklad použití:

- `form.on('file', function (name, file) {  
    // nějaký kód  
})`
- error
  - Vyvolána, když dojde k chybě při zpracování formuláře.
  - Příklad použití:
    - `form.on('error', function (err) {  
    // nějaký kód  
})`
- aborted
  - Vyvolána, když uživatel přeruší přenos požadavku na server, po této události následuje událost error.
  - Příklad použití:
    - `form.on('aborted', function () {  
    // nějaký kód  
})`
- end
 

Vyvolána v okamžiku přijetí celého požadavku na server a zapsání souborů na disk.

  - Příklad použití:
    - `form.on('end', function () {  
    // nějaký kód  
}). [20]`

### 5.5.4 Jimp

Jimp je zkratkou slov JavaScript Image Manipulation Program. Jedná se o javascriptovou knihovnu obsahující sadu funkcí pro práci s obrázkovými soubory, ať už se jedná o ořezávání, změnu velikosti, úpravu barev či dodání efektů. Jimp rovněž disponuje funkcemi pro čtení a zápis souborů. Podporovanými formáty jsou JPEG, PNG, BMP, TIFF a GIF. [25]

#### Instalace

K instalaci slouží příkaz „`npm install jimp`“. [25]

## Funkce

Příklady některých funkcí z modulu Jimp:

- úprava velikosti obrázku
  - `image.resize(w, h, [mode])`
    - Upraví obrázek na danou šířku a výšku.
- ořezání obrázku
  - `image.autocrop([tolerance, frames])`
    - Automaticky ořízne obrázek, pokud mají okraje obrázku stejnou barvu.
  - `image.crop(x, y, w, h)`
    - Ořízne obrázek na velikost `w, h` v prostotu `x, y`, přičemž `x` je počet pixelů od levého horního rohu doprava a `y` počet pixelů od levého horního rohu dolů.
- otáčení a rotace
  - `image.flip(horz, vert)`
    - Otočení obrázku horizontálně či vertikálně, oba parametry funkce jsou logické výrazy.
  - `image.rotate(deg, [mode])`
    - Otočí obrázek o zadaný počet stupňů, pokud je v přetížení této funkce zadán parametr `mode` s hodnotou `false`, rozměry obrázku se při otočení nezmění.
- úprava barev
  - `image.brightness(value)`
    - Upraví jas obrázku na zadanou hodnotu, ta je omezena intervalem `<-1, 1>`.
  - `image.contrast(value)`
    - Upraví kontrast obrázku na zadanou hodnotu, ta je omezena intervalem `<-1, 1>`.
  - `image.greyscale()`
    - Provede konverzi na černobílý obrázek.
  - `image.invert()`
    - Provede inverzi barev.
- efekty

- `image.posterize(n)`
  - Posterizuje obrázek o n úrovní.
- `image.sepia()`
  - Dodá obrázku sépiový efekt.
- zápis souborů
  - `image.write(cesta, callback)`
    - Asynchronně uloží obrázek, jehož lokace je určena. [25]

### 5.5.5 React paginate

React paginate je modul umožňující snadno vytvářet stránky s obsahem v HTML dokumentu. [39]

#### Instalace

K instalaci slouží příkaz „`npm install react-paginate`“. [39]

#### Atributy

- `pageCount` (povinný parametr)
  - Udává počet stran.
- `previousLabel`
  - Element, který se zobrazí na tlačítku na přechod na předchozí stránku.
- `nextLabel`
  - Element, který se zobrazí na tlačítku na přechod na následující stránku.
- `initialPage`
  - Index stránky, která se zobrazí při načtení.
- `containerClassName`
  - Název CSS třídy pro kontejner, který obsahuje čísla stránek.
- `pageClassName`
  - Název CSS třídy pro štítek s číslem stránky.
- `activeClassName`
  - Název CSS třídy pro štítek aktuálně zobrazené stránky.
- `previousClassName`
  - Název CSS třídy pro štítek pro přechod na předchozí stránku.
- `nextClassName`
  - Název CSS třídy pro štítek pro přechod na následující stránku. [39]

### 5.5.6 React Carousel Image Gallery

React Carousel Image Gallery je knihovna pro React.js, která umožňuje vytvářet responsivní galerie obrázků. Nabízí i řadu programovatelných možností. Je tedy vytvořit galerii s náhledy či bez náhledů, galerii automaticky přehrávanou či nikoliv apod. [36]

#### Instalace

K instalaci slouží příkaz „`npm install react-image-gallery`“. [36]

#### Použití

V poli objektů `images` jsou deklarovány zdroje obrázků, přičemž atribut `original` určuje originální obrázek a atribut `thumbnail` jeho náhled. Toto pole objektů poté vstupuje jako parametr `items` do React Image Gallery. [36]

```
import ImageGallery from 'react-image-gallery';
```

```
const images = [  
  {  
    original: 'http://someurl.com',  
    thumbnail: ' http://someotherurl.com ',  
  }  
];
```

```
const Gallery = () => {  
  return <ImageGallery items={images} />;  
}
```

#### Atributy

Příklady některých funkcí z modulu React Carousel Image Gallery:

- `items`: (povinný atribut)
  - Obsahuje pole obrázků, které budou rendrovány.
- `infinite`
  - Atribut určující, zdali se má galerie točit v nekonečné smyčce.
- `thumbnailPosition`
  - Nastavení pozice náhledů obrázků.

- Hodnoty: top, right, bottom, left.
- showFullScreenButton
  - Atribut určující, zdali má být zobrazeno tlačítko pro zobrazení obrázků přes celou obrazovku.
- showPlayButton
  - Atribut určující, zdali má být zobrazeno tlačítko pro automatické přehrávání obrázků.
- startIndex
  - Atribut určující index obrázku, který má být vyrenderován jako první.
- onThumbnailClick
  - Callback funkce umožňující definovat chování náhledu obrázku při kliknutí na něj.
- renderItem
  - Callback funkce umožňující naprogramovat chování renderovací funkce. [36]

## **Funkce**

- play()
  - Spustí přehrávání obrázků.
- pause()
  - Pozastaví přehrávání obrázků.
- fullScreen()
  - Aktivuje full screen mód.
- exitFullScreen()
  - Deaktivuje full screen mód.
- slideToIndex(index)
  - Zobrazí obrázek na zadaném indexu.
- getCurrentIndex()
  - Vrátí index aktuálně zobrazeného obrázku. [36]

### **5.5.7 Styled components**

Styled components umožňuje vytvářet statické stylizované komponenty pomocí CSS. Oproti vytváření komponent přes standardní React.js, jsou komponenty vytvořené se Styled



components pouze statické, nemohou mít vlastní atributy, metody apod. K jejich stylování se používá doslovná textová CSS syntaxe. [44]

### **Instalace**

K instalaci slouží příkaz „npm install styled-components“. [44]

### **Použití**

Vytvoření komponenty odvozené on h1 HTML elementu: [44]

```
const Title = styled.h1`  
  font-size: 1.5em,  
  text-align: center,  
  color: palevioletred,  
`
```

Vytvoření komponenty odvozené on button HTML elementu s atributem color nastaveným na hodnotu grey:

```
const Button = styled.button`color: grey`;
```

## **5.5.8 React Flexbox Grid**

React Flexbox Grid je modul, který umožňuje snadnou tvorbu rozložení elementů v React komponentách. Modul rovněž nabízí snadnou tvorbu responzivních komponent, respektive změnu rozložení v závislosti na šířce a výšce komponenty. [35]

### **Instalace**

K instalaci slouží příkaz „npm install react-flexbox-grid“. [35]

### **Použití**

Příklad vytvoření tabulky se šířkou sloupců v poměru 1:1 při šířce komponenty do 712 pixelů a v poměru 2:1 při šířce stránky větší než 712 pixelů: [35]

```
<Row>  
  <Col xs={6} sm={8}>Sloupec 1</Col>  
  <Col xs={6} sm={4}>Sloupec 2</Col>  
</Row>
```

### 5.5.9 React select

React Select je modul, který umožňuje vytváření stylovaných elementů umožňujících výběr z listu možností. Tento modul namísto klasického `select` HTML element vytváří stylované `div` elementy. To zajišťuje, že element vytvořený tímto modulem bude ve všech prohlížečích vypadat stejně, na rozdíl právě od standardního `select` elementu, který je vytvářen každým prohlížečem individuálně a může mít jiný design. [42]

#### Instalace

K instalaci slouží příkaz „`npm install react-select`“. [42]

#### Použití

Konstanta `RenderSelect` je funkcí, která vyrenderuje `Select` element s možnostmi obsaženými v konstantě `options`. Proměnná `options` je polem objektů obsahujících atributy `value` a `label`, přičemž `value` je hodnota dané možnosti a `label` štítek, který je zobrazen uživateli: [42]

```
import React from 'react',
import Select from 'react-select',

const options = [
  { value: 'možnost1', label: 'Možnost1' },
  { value: 'možnost2', label: 'Možnost2' },
  { value: 'možnost3', label: 'Možnost3' },
],

const RenderSelect = () => {
  return <Select
    value={options[0]}
    options={options}
  />
}
```

#### Atributy

Příklady některých funkcí z modulu React Select:

- `autoFocus`
  - Nastaví elementu `focus` atribut, jakmile je element vytvořen.
- `className`
  - Název CSS třídy.
- `isDisabled`
  - Zneaktivnění elementu.
- `isMulti`
  - Umožnění výběru vícero možností.
- `isSearchable`
  - Umožní filtrování možností.
- `options`
  - Pole možnosti, ze kterých uživatel vybírá.
- `name`
  - HTML atribut `name` pro tento element. [42]

### 5.5.10 Font Awesome

Font Awesome je open–source knihovna pro JavaScript a React.js, jež obsahuje nástroje pro práci s SVG ikonami při tvorbě uživatelských rozhraní pomocí jazyků JavaScript, TypeScript nebo některých z framework odvozených z těchto jazyků. Font Awesome obsahuje několik submodulů, jako například „`free-solid-svg-icons`“, obsahující bezplatnou sadu SVG ikon, nebo „`react-fontawesome`“, tedy knihovnu pro React.js, která umožňuje práci s ikonami, jejich tvorbu a stylování. K dispozici jsou rovněž knihovny pro frameworky jako JQuery, Vue.js, Angular či React Native. [1] [19]

#### Instalace

K instalaci slouží příkaz „`npm install fontawesome`“. [19]

#### Submoduly

V této části jsou popsány některé submoduly knihovny Font Awesome.

## React Font Awesome

Submodul React Font Awesome je integrací modulu Font Awesome pro framework React.js. Umožňuje interakci s bezplatnými Font Awesome sadami ikon, jako např. „free-brands-svg-icons“ či „free-regular-svg-icons“, ale i s profesionálními sadami jako „pro-brands-svg-icons“, „pro-regular-svg-icons“ nebo „pro-light-svg-icons“. [1]

### Instalace

K instalaci slouží příkaz „npm install @fontawesome/react-fontawesome“. [1]

### Příklady použití

Následující kód je příkladem implementace modulu Font Awesome v React.js projektu. Jsou zde uvedeny dva způsoby renderování ikony faCoffee z knihovny Font Awesome:

```
import { FontAwesomeIcon } from '@fortawesome/react-fontawesome',
import { faCoffee } from '@fortawesome/free-solid-svg-icons',

const Element = () => {
  <div>
    // renderování ikony faCoffee přes import objektu faCoffee
    <FontAwesomeIcon icon={faCoffee} size={'1x'} />

    // renderování ikony faCoffee přes použití prefixu a názvu
    ikony
    <FontAwesomeIcon icon={['fa', 'coffee']} size={'1x'} />
  </div>
}
```

### Atributy elementu Font Awesome Icon

Stručný výčet nejdůležitějších atributů modulu Font Awesome Icon:

- icon
  - Název ikon, která se má vyrenderovat.
- color

- Název barvy ikony.
- `spin`
  - Logická hodnota určující, zdali má ikona rotovat či nikoliv.
- `pulse`
  - Logická hodnota určující, zdali má ikona pulsovat či nikoliv.
- `rotation`
  - Číslo určující stupeň pootočení ikony vůči vertikální ose.
- `transform`
  - Vlastnost umožňující např. zrcadlové otočení ikony.
- `size`
  - Vlastnost určující předdefinovanou velikost ikon, např. `xs`, `md`, `xl`. [1]

### 5.5.11 Tailwind CSS

Tailwind CSS je škálovatelný webový framework. Umožňuje díky jednoduché syntaxi stylování HTML elementů a do značné míry je možné jím nahradit individuální CSS třídy, nebo jej použít namísto stylování pomocí JavaScriptu a nahradit tedy jeho poněkud komplikovanou syntaxi. [45] [46]

#### Instalace

K instalaci slouží příkaz „`npm install tailwind --save-dev`“. [46]

#### Příklady použití

Příklad použití tailwind CSS ve frameworku React.js. Tento kód vytváří `div` element s CSS třídami `text-blue-100`, `pb-1` a `pt-3`. v něm je vnořený `p` element s třídou `margin-auto` a textem:

```
<div id='outer' className='text-blue-100 pb-1 pt-3 '>
  <p id='inner' className='margin-auto'>Text</p>
</div>
```

Ten samý element vytvořený čistě pomocí HTML a stylovaný připojeným CSS souborem:

HTML kód:

```
<div id='outer' class='outer'>
  <p id='inner' class='inner'>Text</p>
</div>
```

CSS kód:

```
.outer {
  color: #ebf8ff;
  padding-bottom: 0.25rem;
  padding-top: 0.75rem;
}
.inner {
  margin-top: auto;
  margin-bottom: auto;
}
```

Ten samý kód vytvořený pomocí ES6 JavaScriptu:

```
const outer = document.createElement('div');
outer.className = 'outer';
outer.id = 'outer';
outer.style.color = '#ebf8ff';
outer.style.paddingBottom = '0.25rem';
outer.style.paddingTop = '0.75rem';

const inner = document.createElement('p');
inner.className = 'inner';
inner.id = 'inner';
inner.style.marginTop = 'auto';
inner.style.marginBottom = 'auto';
inner.textContent = 'Text';

outer.appendChild(inner);
```

## 5.6 Nastavení pro vývoj aplikace

V následujících částech jsou jednoduché instrukce, jak nastavit aplikaci, respektive její zdrojové soubory pro další vývoj.

### 5.6.1 Frontend

List instrukcí pro nastavení systému pro vývoj frontendu aplikace:

- Nainstalovat Node.js verze 10.16.0,
- Umístit složku se zdrojovými kódy frontendu na systémový disk
- Otevřít příkazový řádek ve složce s frontendem.
- Spustit příkaz „npm install“.
- Spustit příkaz „npm start“.
- Povolit v prohlížeči „cross-origin resource sharing“ (umožnění komunikace s backendem aplikace).
- Pro sestavení optimalizovaného výstupu spustit příkaz „npm run build“.

- Zkopírování obsahu složky build do složky src/backend/public ve složce se zdrojovým kódem backendu aplikace a nahrazení stávající obsahu složky public.

### **5.6.2 Backend**

List instrukcí pro nastavení systému pro vývoj backendu aplikace:

- Nainstalovat Node.js verze 10.16.0.
- Umístit složku se zdrojovými kódy backendu na systémový disk
- Otevřít příkazový řádek ve složce s backendem.
- Spustit příkaz „npm install“.
- Spustit příkaz „npm start“.
- Pro vytvoření výstupu aplikace a installeru pro Windows spustit příkaz „npm run dist“.



## 6 SOUBORY PRO UKLÁDÁNÍ OBRÁZKŮ

V této části jsou popsány typy souborů používané pro ukládání obrázkových souborů. Základní členění je na ztrátové a bezztrátové formáty.

### 6.1 Ztrátové formáty

U ztrátových formátů je obraz je komprimován, aby vznikl velikostně menší soubor. Kompresi ústí ve ztrátu kvality, obvykle tato ztráta ale nebývá patrná. [29] [24]

#### 6.1.1 JPG/JPEG

JPEG (Joint Photographic Experts Group), je obrázkový formát dat, který je standardem pro obrázky pořizované digitálními fotoaparáty nebo pro obrázky umístěvané na internet. Umožňuje dosáhnout kompresního poměru až 10:1. [8] [26]

Příklady programů pro práci s typem JPG:

- Windows
  - File Viewer Plus
  - Microsoft Windows Photos
  - Microsoft Windows Photo Viewer
  - Microsoft OneDrive
  - Microsoft Paint
  - Adobe Photoshop CC 2019
  - CorelDRAW Graphics Suite 2018
  - Corel PaintShop Pro 2019
  - ACD Systems ACDSee 2018
- Mac
  - Apple Preview
  - Apple Photos
  - Adobe Photoshop CC 2019
  - ACD Systems ACDSee Mac Pro 3
  - MacPhun ColorStrokes
- Linux
  - GIMP
  - Shotwell [26]

## Výhody a použití

Mezi výhody formátu JPEG patří:

- Flexibilita.
- Vysoká míra komprese umožňuje rychlé stahování z webu.
- Nízká míra komprese umožňuje zachovat kvalitu pro tisk.
- Možnost zredukovat velikost souboru na velmi malou velikost. [26]

## 6.2 Bezztrátové formáty

Bezztrátové formáty obrázků zachovávají všechna data v původním souboru. I bezztrátové formáty dat mohou nabízet možnosti komprese dat, následnou dekompresí je nicméně možné získat původní, nijak nemodifikovaná data v originální kvalitě. [29]

### 6.2.1 GIF

GIF (Graphics Interchange Format) je vhodný zejména pro zobrazení vektorových obrázků, např. nápisů, plánek nebo log. Obsahuje pouze 256 indexovaných barev, na druhou stranu ale mohou GIF formáty být zkomprimovány na velmi malou velikost. Na internetu jsou populární hlavně kvůli své schopnosti zobrazovat animace. [22]

Příklady programů pro práci s typem GIF:

- Windows
  - File Viewer Plus
  - Microsoft Windows Photos
  - Microsoft Windows Photo Viewer
  - Adobe Photoshop CC 2019
  - Adobe Photoshop Elements 2019
  - Adobe Illustrator CC 2019
  - CorelDRAW Graphics Suite 2018
  - Corel PaintShop Pro 2019
  - ACD Systems ACDSSee Photo Studio
  - ACD Systems Canvas X 2019
  - Roxio Creator NXT Pro 6
  - Laughingbird The Logo Creator
  - Nuance OmniPage Ultimate
- Mac
  - Apple Preview

- Apple Photos
- Apple Safari
- Adobe Photoshop CC 2019
- Adobe Photoshop Elements 2019
- Adobe Illustrator CC 2019
- Roxio Toast 17
- Laughingbird The Logo Creator
- Linux
  - GIMP [22]

### Výhody a použití

Mezi výhody formátu GIF patří:

- Možnost zobrazení animací.
- Možnost umístění na barevné pozadí díky nastavení transparentnosti.
- Vhodný formát pro tvorbu ikon či log. [22]

### 6.2.2 PNG

PNG (Portable Network Graphic) je soubor obsahující bitmapu indexovaných barev a používá bezztrátovou kompresi (podobně jako GIF). Soubory PNG se běžně používají k ukládání grafiky pro webové obrázky. [33]

Formát PNG byl vytvořen v reakci na omezení ve formátu GIF. Formát GIF byl totiž do roku 2004 chráněn patentem a jeho použití mohlo být pro komerční účely zpoplatněno. Kromě toho, GIF měl a má i technická omezení. Zatímco obrázky GIF podporují pouze zcela neprůhledné nebo zcela průhledné pixely, obrázky PNG mohou obsahovat 8bitový průhledný kanál, který umožňuje barvám měnit průhlednost od stoprocentní průhlednosti po stoprocentní neprůhlednost. Obrázky PNG nelze animovat jako obrázky GIF. Lze však animovat související formát MNG. PNG formát je podporován většinou webových prohlížečů. [33]

Příklady programů pro práci s typem PNG:

- Windows
  - File Viewer Plus
  - Microsoft Windows Photos
  - Microsoft Windows Photo Viewer

- Adobe Photoshop CC 2019
- Adobe Photoshop Elements 2019
- Adobe Illustrator CC 2019
- CorelDRAW Graphics Suite 2018
- Corel PaintShop Pro 2019
- ACD Systems ACDSee Photo Studio
- ACD Systems Canvas X 2019
- Roxio Creator NXT Pro 6
- Laughingbird The Logo Creator
- Nuance OmniPage Ultimate
- Mac
  - Apple Preview
  - Apple Photos
  - Apple Safari
  - Adobe Photoshop CC 2019
  - Adobe Photoshop Elements 2019
  - Adobe Illustrator CC 2019
  - Roxio Toast 17
  - Laughingbird The Logo Creator
- Linux
  - GIMP [32]

### **Výhody a použití**

Mezi výhody formátu PNG patří:

- Možnost měnit stupeň průhlednosti.
- Možnost dosáhnout relativně velké komprese. [33]

### **6.2.3 TIFF**

TIFF (Tagged Image Format File) je obrázkový formát používaný zejména pro fotografie a jejich tisk, a to díky své mimořádně vysoké kvalitě. Má mimo jiné také širokou paletu barev. Může být ukládán v bezeztrátovém formátu TIFF, nebo v komprimované podobě v příponou JPEG. Na webu se skoro nevyskytuje, zdaleka ne všechny webové prohlížeče tento formát podporují. [48]

Příklady programů pro práci s typem TIFF:

- Windows
  - File Viewer Plus
  - Microsoft Windows Photos
  - Microsoft Windows Photo Viewer
  - Adobe Photoshop CC 2019
  - Adobe Photoshop Elements 2019
  - Adobe Lightroom CC 2019
  - Adobe Illustrator CC 2019
  - CorelDRAW Graphics Suite 2018
  - Corel PaintShop Pro 2019
  - ACD Systems ACDSSee Photo Studio
  - ACD Systems Canvas X 2019
  - Nuance PaperPort 14
  - Nuance OmniPage Ultimate
  - Roxio Creator NXT Pro 6
- Mac
  - Apple Preview
  - Apple Photos
  - Adobe Photoshop CC 2019
  - Adobe Photoshop Elements 2019
  - Adobe Illustrator CC 2019
  - ACD Systems ACDSSee Photo Studio for Mac
  - Roxio Toast 17
  - MaxPhun ColorStrokes
- Linux
  - GIMP [48]

### **Výhody a použití**

Mezi výhody formátu TIFF patří:

- Možnost zachování vysoké kvality souboru pro tisk.
- Možnost zachování originálu při skenování. [48]

## 6.2.4 RAW

RAW formát obsahuje nezpracovaná data zachycená digitálním fotoaparátem nebo snímačem skeneru. Obvykle jsou data zpracována a převedena do jiného formátu (např. JPEG nebo TIFF). RAW soubory jsou uloženy ve dvou verzích, zpracované a nezpracované, a to z důvodu uchování původní kvality, aby jejich atributy jako vyvážení bílé nebo expozice, mohly být později upraveny například ve Photoshopu. Formátů RAW obrázků je několik desítek, například CRW, NEF nebo DNG. [34]

Příklady programů pro práci s typem RAW:

- Windows
  - File Viewer Plus
  - Microsoft Windows Photos
  - Microsoft Windows Photo Viewer
  - Adobe Photoshop CC 2019
  - Adobe Photoshop Elements 2019
  - Corel PaintShop Pro 2019
  - ACD Systems ACDSee Photo Studio
  - ACD Systems Canvas X 2019
  - RawTherapee
  - GIMP s UFRaw plug-in
- Mac
  - Apple Preview
  - Adobe Photoshop CC 2019
  - Adobe Photoshop Elements 2019
  - GIMP s UFRaw plug-in
  - MacPhun ColorStrokes
  - WidsMob PhotoVault
- Linux
  - GIMP
  - RawTherapee [34]

## 7 ZÁVĚR

Závěr obsahuje zhodnocení práce autorem a výstup z HTML validátoru.

### 7.1 Zhodnocení

Jako součást této bakalářské práce byla vytvořena aplikace pro generování galerií ve formě webových stránek. Ta umožňuje tvorbu galerií, změnu velikosti nahraných obrázků, zadání počtu pohledů a nastavení nadpisu galerie, přesně tak, jak je popsáno v zadání práce. V práci jsou rovněž popsány technologie použité při tvorbě aplikace, stejně jako soubory používané pro ukládání obrázků. V práci došlo k praktické demonstraci celé řady technologií a frameworků, které se používají po celém světě pro tvorbu různých typů aplikací. Součástí práce je také výstup z HTML validátoru. HTML validátor neodhalil ve vygenerovaných webových stránkách žádnou chybu.

Aplikace včetně zdrojových souborů je k dispozici na přiloženém CD.

### 7.2 Výstup z HTML validátoru

Webové stránka s galerií má podle zadání projít kontrolou na adrese <https://validator.w3.org>. K testování a vývoji byl použit prohlížeč Google Chrome verze 81.0.4044.138. Kontrola HTML kódu probíhala následovně:

1. Otevření adresy <https://validator.w3.org>.
2. Zvolení možnosti „Validate by direct input“.
3. Otevření webové stránky aplikace pro generování galerií ve formě webových stránek.
4. Otevření vývojářských možností (klávesa F12).
5. Otevření vývojářských možností na záložce „Elements“.
6. Kliknutí pravým tlačítkem na `<html>` element `> Copy > Copy outerHTML`.
7. Vložení zkopírovaného kódu na stránku validátoru (nutno rovněž manuálně na začátek kódu element `<!DOCTYPE html>`, který nebyl zkopírován).
8. Kliknutí na tlačítko „Check“ ve validátoru.

#### 7.2.1 Validace úvodní stránky

Validace úvodní stránky nevykazuje při prvním načtení žádnou chybu či varování.

Při přechodu go stránky s galerií a zpět na úvodní stránku vykazuje validátor varování: „The type attribute for the style element is not needed and should be omitted.“, které je

detailněji popsáno v následující části. To že se toto varování nezobrazuje při prvním načtení stránky je způsobeno modulem Font Awesome, který je načítán ve stránce s galerií. Při přechodu zpět na úvodní stránku potom v HTML kódu tak mohou zůstat elementy z tohoto modulu.

### 7.2.2 Validace stránky s galerií

Validace úvodní stránky vykazuje 2 varování a žádnou chybu.

První varování: „The type attribute for the style element is not needed and should be omitted.“ To je způsobeno přítomností elementu `<style type="text/css">`, které je automaticky vytvářeno modulem `@fortawesome/react-fontawesome`, jenž umožňuje vytváření SVG ikon.

Druhé varování „Possible misuse of aria-label.“ je způsobeno přítomností atributu `aria-label` v elementu `<div class="image-gallery-thumbnails-container" aria-label="Thumbnail Navigation" style="transform: translate3d(0px, 0px, 0px),">`, což je kontejner pro náhledy obrázku a je automaticky vytvářen modulem `react-image-gallery`.

Žádné z varování nemá jakýkoliv vliv na správnou funkci aplikace, jedná se spíše o doporučení, aby HTML kód stoprocentně odpovídal striktním standardům HTML5.

## 7.3 Vyjádření k poznámkám z předchozí obhajoby

Vzhledem k tomu, že tato verze práce odstraňuje nedostatky ve verzi minulé, zařazuje autor na úplný závěr vysvětlení toho, jak byly ony nedostatky odstraněny.

### Deformace fotografií při jejich transformaci

Předchozí verze aplikace v určitých případech deformovala nahrané fotografie. Algoritmus, jenž měnil rozměry fotografie rozraštoval pixely a fotografie tím mohla ztratit na kvalitě. Opravená verze aplikace nemění rozměry fotografií jejich prostým natažením či smrsknutím, ale dochází k ořezu fotek. Uživatel má při nahrávání fotografií možnost zaškrtnout volbu „auto offset“, která zajistí to, že fotografie bude oříznuta rovnoměrně podél okrajů.

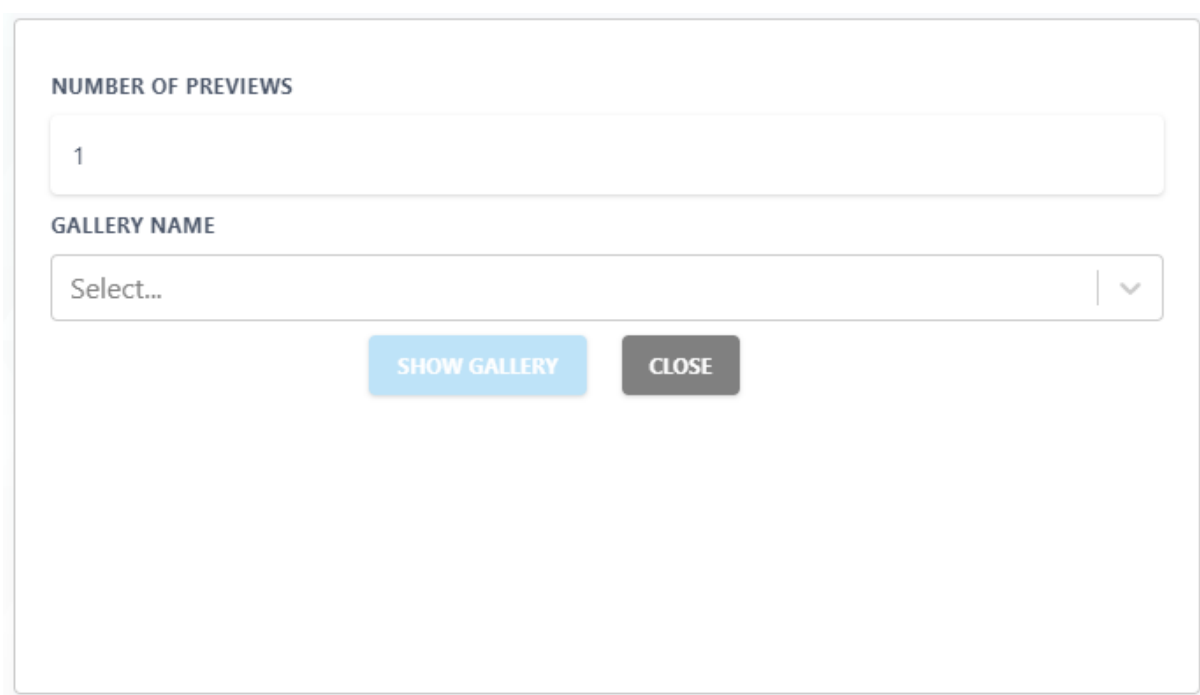
Například při nahrání fotky, která má na šířku 200 px a zadání nové šířky 100 px, automaticky se ubere 50 px z každého okraje. Pokud bude nahrána ta samá fotografie s šířkou 200 px, zadána nová šířka 100 px a uživatel zvolí horizontální odsazení (x offset) na hodnotu 80 px, z fotky se odřízne 80 px z levého okraje a 20 px z pravého okraje.



Nový algoritmus sloužící k transformaci fotografií již neumožňuje zvětšení fotografie. To by totiž znamenalo pouze to, že by se k fotografii byly algoritmem přidány černé okraje.

### **Zadaný počet zobrazených fotografií na stránce fungující pouze pro první náhled**

Předchozí verze aplikace umožňovala zadat počet fotografií na jednu stránku pouze při jejich nahrávání, při opětovném spuštění aplikace a zobrazení galerie již nikoliv. Tento problém byl vyřešen přidáním možnosti pro volbu galerie, kterou chce uživatel zobrazit, která rovněž umožňuje zadání počtu náhledů na stránku.



The image shows a modal dialog box with a light blue border. At the top, the text "NUMBER OF PREVIEWS" is displayed in a bold, dark font. Below this is a text input field containing the number "1". Underneath, the text "GALLERY NAME" is displayed in a bold, dark font. Below this is a dropdown menu with the text "Select..." and a small downward-pointing arrow on the right side. At the bottom of the dialog, there are two buttons: a light blue button labeled "SHOW GALLERY" and a dark grey button labeled "CLOSE".

Obrázek 7: Modální okno

### **Nemožnost uložit vygenerovanou galerii pro pozdější použití**

I předchozí verzi bylo možno galerii uložit a zobrazit později. Přesněji řečeno, transformované obrázky se ukládaly do k tomu určené složky. Nově je možné přes webový prohlížeč zobrazenou galerii ve formě zip souboru, nebo jednotlivé obrázky zvlášť.

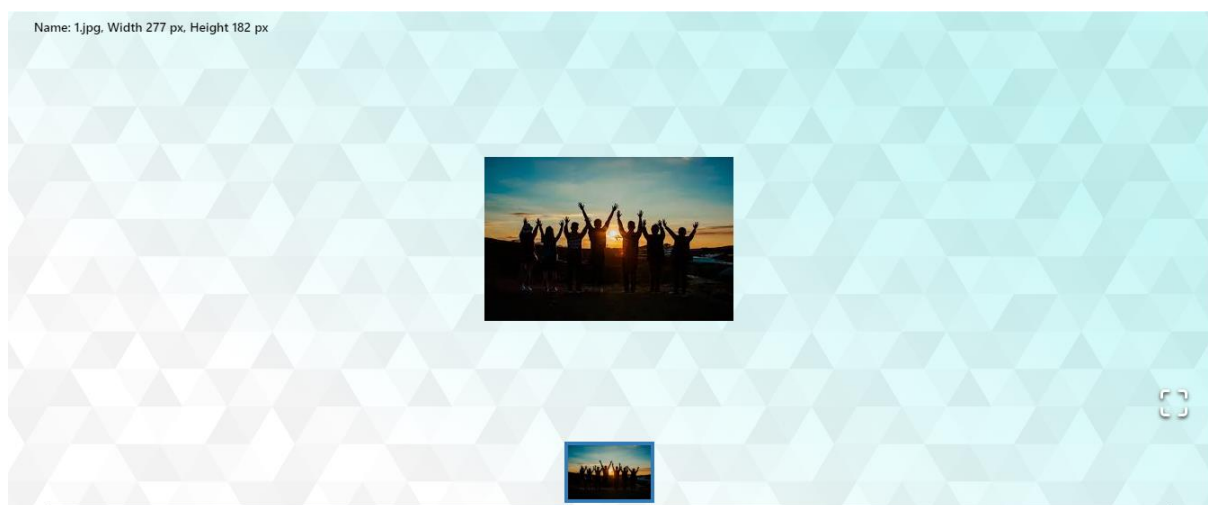
K prohlížení je i nadále nutno mít spuštěnou aplikaci a ke galeriím je možno přistoupit skrze webový prohlížeč.

## Nemožnost vygenerovat více galerií současně

Nově je možno vygenerovat více galerií. Pro každou galerii je vytvořena nová složka, do které jsou obrázky nahrány. Uživatel dostává prostřednictvím modálního okna možnost výběru galerie k prohlížení.

## Chybějící možnost využití metadat a popisků obrázků

Galerie zobrazuje u každého obrázku v jeho levém horním rohu název obrázku a jeho rozměry.



Obrázek 8: Zobrazení metadat

## Aplikace neumí pracovat s některými obrázkovými formáty, které jsou v textu popsány

Typy souborů popsané v práci popisují běžné formáty používané pro ukládání fotografií. Aplikace samotná podporuje formáty GIF, JPG/JPEG a PNG.

Formát RAW používají například digitální fotoaparáty a tyto soubory jsou obvykle převáděny na ztrátový formát JPG pro pozdější využití. Znamená to tedy, že formát RAW se na běžné prohlížení fotografií na počítači nepoužívá.

Formát TIFF nedokáže být aplikací zpracován proto, že většina prohlížečů tento formát standartně nedokáže bez příslušného plug-inu zobrazit. Aplikace proto tento formát nepodporuje.

## **Nedostatečná kvalita uživatelského rozhraní**

Uživatelské rozhraní je kompletně nové. Nabízí lepší uživatelskou přívětivost, možnost mazání jednotlivých fotografií a jejich stahování. K velkému zlepšení došlo v oblasti zpracování chyb.

Samotná galerie také prošla celou řadou zlepšení. Tím zásadním je možnost procházet galerie ve smyčce.

Aplikace je nově k dispozici s installerem pro Windows, jenž aplikaci automaticky nainstaluje a spustí.

## 8 POUŽITÁ LITERATURA

- [1] @fontawesome/react-fontawesome - npm, 2020. *Npm* [online]. [cit. 2020-05-19]. Dostupné z: <https://www.npmjs.com/package/@fortawesome/react-fontawesome#introduction>
- [2] App-root-path, 2020. *Npm* [online]. [cit. 2020-05-21]. Dostupné z: <https://www.npmjs.com/package/app-root-path>
- [3] Axios, 2020. *Npm* [online]. [cit. 2020-05-21]. Dostupné z: <https://www.npmjs.com/package/axios>
- [4] Basic Use | Font Awesome, 2020. *Font Awesome* [online]. Fonticons, Inc. [cit. 2020-05-19]. Dostupné z: <https://fontawesome.com/how-to-use/on-the-web/referencing-icons/basic-use>
- [5] BEAUDRU, Manuel, 2020. Checkbox with styled-components. *GitHub Gist* [online]. San Francisco [cit. 2020-05-14]. Dostupné z: <https://gist.github.com/mbeaudru/d473014009ef12bc9eef076f3fa004ff>
- [6] Bluebird, 2020. *Npm* [online]. [cit. 2020-05-21]. Dostupné z: <https://www.npmjs.com/package/bluebird>
- [7] Body-parser, 2020. *Npm* [online]. [cit. 2020-05-21]. Dostupné z: <https://www.npmjs.com/package/body-parser>
- [8] BRADY, Gavin, 2019. How to Convert an Image to JPG Format. *How-To Geek* [online]. LifeSavvy Media [cit. 2019-11-05]. Dostupné z: <https://www.howtogeek.com/362730/how-to-convert-an-image-to-jpg-format/>
- [9] CLOUDER, Alibaba, 2020. What Is ElectronJS and Why Should You Use It?. *Alibaba Cloud* [online]. [cit. 2020-05-13]. Dostupné z: [https://www.alibabacloud.com/blog/what-is-electronjs-and-why-should-you-use-it\\_581971](https://www.alibabacloud.com/blog/what-is-electronjs-and-why-should-you-use-it_581971)
- [10] Create Customizable, Hi-Def, and Scalable Backgrounds, 2020. *SVG Backgrounds* [online]. [cit. 2020-05-21]. Dostupné z: <https://www.svgbackgrounds.com/#subtle-prism>

- [11] Dotenv. *Npm* [online]. [cit. 2020-05-21]. Dostupné z: <https://www.npmjs.com/package/dotenv>
- [12] Electron-builder, 2020. *Npm* [online]. [cit. 2020-05-21]. Dostupné z: <https://www.npmjs.com/package/electron-builder>
- [13] Electron-compile, 2020. *Npm* [online]. [cit. 2020-05-21]. Dostupné z: <https://www.npmjs.com/package/electron-compile>
- [14] Electron, 2020. *Npm* [online]. [cit. 2020-05-21]. Dostupné z: <https://www.npmjs.com/package/electron>
- [15] Express basic routing, 2017. *Express* [online]. StrongLoop, IBM [cit. 2019-10-06]. Dostupné z: <https://expressjs.com/en/starter/basic-routing.html>
- [16] Express routing, 2017. *Express* [online]. StrongLoop, IBM [cit. 2019-10-06]. Dostupné z: <https://expressjs.com/en/guide/routing.html>
- [17] Express. *Npm* [online]. [cit. 2019-10-06]. Dostupné z: <https://www.npmjs.com/package/express>
- [18] File-system - npm. *Npm* [online]. [cit. 2019-10-06]. Dostupné z: <https://www.npmjs.com/package/file-system>
- [19] Font-awesome, 2020. *Npm* [online]. [cit. 2020-05-19]. Dostupné z: <https://www.npmjs.com/package/font-awesome>
- [20] Formidable - npm. *Npm* [online]. [cit. 2019-10-06]. Dostupné z: <https://www.npmjs.com/package/formidable>
- [21] Gallery - Free multimedia icons, 2020. *Flaticon* [online]. Freepik Company S.L. [cit. 2020-05-21]. Dostupné z: [https://www.flaticon.com/free-icon/gallery\\_833367?term=gallery&page=1&position=6](https://www.flaticon.com/free-icon/gallery_833367?term=gallery&page=1&position=6)
- [22] GIF File Extension: What is a GIF file?, 2019. *FileInfo: The File Extensions Database* [online]. Sharpened Productions [cit. 2019-10-07]. Dostupné z: <https://fileinfo.com/extension/gif>

- [23] GRAY, Tanya, 2019. What the heck is npm?. *Medium* [online]. Medium Corporation [cit. 2019-10-06]. Dostupné z: <https://medium.com/@tanya/what-the-heck-is-npm-b8168f61e3b5>
- [24] Image File Formats - All About Images: Native File Formats, 2018. *Library - University of Michigan* [online]. Ann Arbor, Michigan: University of Michigan [cit. 2019-10-07]. Dostupné z: <https://guides.lib.umich.edu/c.php?g=282942&p=1885348>
- [25] Jimp - npm. *Npm* [online]. [cit. 2019-10-07]. Dostupné z: <https://www.npmjs.com/package/jimp>
- [26] JPG File Extension: What is a JPG file?, 2019. *FileInfo: The File Extensions Database* [online]. Sharpened Productions [cit. 2019-10-07]. Dostupné z: <https://fileinfo.com/extension/jpg>
- [27] KATARIA, Saransh, 2019. How to setup Typescript with Babel and Webpack. *Wisdom Geek* [online]. Wisdom Geek [cit. 2019-10-06]. Dostupné z: <https://www.wisdomgeek.com/development/web-development/how-to-setup-typescript-with-babel-and-webpack>
- [28] Lodash, 2020. *Npm* [online]. [cit. 2020-05-21]. Dostupné z: <https://www.npmjs.com/package/lodash>
- [29] LUNDQUIST, Samuel. Image file formats: when to use each type of file. *99designs* [online]. [cit. 2019-10-07]. Dostupné z: <https://en.99designs.de/blog/tips/image-file-types>
- [30] Node.js - Quick Guide: Node.js - Introduction. *Tutorialspoint* [online]. [cit. 2019-10-06]. Dostupné z: [https://www.tutorialspoint.com/nodejs/nodejs\\_quick\\_guide.htm](https://www.tutorialspoint.com/nodejs/nodejs_quick_guide.htm)
- [31] Open, 2020. *Npm* [online]. [cit. 2020-05-21]. Dostupné z: <https://www.npmjs.com/package/open>
- [32] PANDIT, Nitin, 2020. What and Why React.js. *C# Corner* [online]. [cit. 2020-05-13]. Dostupné z: <https://www.c-sharpcorner.com/article/what-and-why-reactjs/>

- [33] PNG File Extension: What is a PNG file?, 2019. *FileInfo: The File Extensions Database* [online]. Sharpened Productions [cit. 2019-10-07]. Dostupné z: <http://www.fileinfo.com/extension/png>
- [34] RAW File Extension: What is a RAW file?, 2019. *FileInfo: The File Extensions Database* [online]. Sharpened Productions [cit. 2019-10-07]. Dostupné z: <https://fileinfo.com/extension/raw>
- [35] React-flexbox-grid, 2020. *Npm* [online]. [cit. 2020-05-19]. Dostupné z: <https://www.npmjs.com/package/react-flexbox-grid>
- [36] React-image-gallery, 2020. *Npm* [online]. [cit. 2020-05-19]. Dostupné z: <https://www.npmjs.com/package/react-image-gallery>
- [37] React-modal, 2020. In: *Npm* [online]. [cit. 2020-05-21]. Dostupné z: <https://www.npmjs.com/package/react-modal>
- [38] React, 2020. *Npm* [online]. [cit. 2020-05-21]. Dostupné z: <https://www.npmjs.com/package/react>
- [39] React-paginate - npm. *Npm* [online]. [cit. 2020-05-13]. Dostupné z: <https://www.npmjs.com/package/react-paginate>
- [40] React-router-dom, 2020. *Npm* [online]. [cit. 2020-05-21]. Dostupné z: <https://www.npmjs.com/package/react-router-dom>
- [41] React-scripts, 2020. *Npm* [online]. [cit. 2020-05-21]. Dostupné z: <https://www.npmjs.com/package/react-scripts>
- [42] React-select, 2020. *Npm* [online]. [cit. 2020-05-19]. Dostupné z: <https://www.npmjs.com/package/react-select>
- [43] React-styled-flexboxgrid, 2020. *Npm* [online]. [cit. 2020-05-21]. Dostupné z: <https://www.npmjs.com/package/react-styled-flexboxgrid>
- [44] Styled-components, 2020. *Npm* [online]. [cit. 2020-05-19]. Dostupné z: <https://www.npmjs.com/package/styled-components>

- [45] Tailwind CSS - A Utility-First CSS Framework for Rapidly Building Custom Designs, 2020. *Tailwind CSS* [online]. [cit. 2020-05-19]. Dostupné z: <https://tailwindcss.com/#what-is-tailwind>
- [46] Tailwind, 2020. *Npm* [online]. [cit. 2020-05-21]. Dostupné z: <https://www.npmjs.com/package/tailwindcss>
- [47] The package.json guide, 2019. *Flavio Copes* [online]. Flavio Copes [cit. 2019-10-06]. Dostupné z: <https://flaviocopes.com/package-json/>
- [48] TIFF File Extension: What is a TIFF file?, 2019. *FileInfo: The File Extensions Database* [online]. Sharpened Productions [cit. 2019-10-07]. Dostupné z: <https://fileinfo.com/extension/tiff>
- [49] WARCHOLINSKI, Matt, 2020. What is Electron JS?. *Brainhub* [online]. Miami [cit. 2020-05-13]. Dostupné z: <https://brainhub.eu/blog/what-is-electron-js/>
- [50] What is npm?. *Node.js* [online]. [cit. 2019-10-06]. Dostupné z: <https://nodejs.org/en/knowledge/getting-started/npm/what-is-npm/>