



**MATEMATICKO-FYZIKÁLNÍ
FAKULTA**
Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

David Nápravník

Softwarové řešení digitálních archivů

Katedra teoretické informatiky a matematické logiky

Vedoucí bakalářské práce: Mgr. Macková Kateřina

Studijní program: Informatika (B1801)

Studijní obor: IPSS (1801R048)

Praha 2021

Prohlašuji, že jsem tuto bakalářskou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného titulu.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V dne

Podpis autora

TODO Podekovani:
Petra Hoffmannová
Kateřina Macková

Název práce: Softwarové řešení digitálních archivů

Autor: David Nápravník

Katedra: Katedra teoretické informatiky a matematické logiky

Vedoucí bakalářské práce: Mgr. Macková Kateřina, katedra

Abstrakt: TODO Abstrakt cz

Klíčová slova: digitální archiv web databáze

Title: Software solution for digital archives

Author: David Nápravník

Department: Department of Theoretical Computer Science and Mathematical Logic

Supervisor: Mgr. Macková Kateřina, department

Abstract: TODO Abstrakt en

Keywords: digital archive web database

Obsah

1	zadani	4
2	existujici produkty	5
3	Vyber technologii	6
3.1	Frontend	6
3.1.1	Single page aplication	6
3.1.2	React	6
3.1.3	Dalsi mozne technologie	6
3.2	Backend	6
3.2.1	Express.js	7
3.2.2	MongoDB	7
3.2.3	Dalsi mozne technologie	7
4	diagram systemu	8
5	implementace backendu	9
5.1	server	9
5.2	knihovny	9
5.2.1	Express.js	9
5.2.2	mongoose	9
5.3	dokumentace	9
5.3.1	vlastni mini knihovna pro dokumentaci	9
5.4	routes	9
5.4.1	uzivatel	9
5.4.2	autorizace	9
5.4.3	zaznam	9
5.4.4	stranka	9
5.4.5	nahravani souboru	9
5.5	models	9
5.5.1	zaznam	9
5.5.2	stranka	9
5.5.3	uzivatel	9
6	implementace frontendu	10
6.1	server	11
6.1.1	kompilace	11
6.1.2	npm	11
6.2	knihovny	11
6.2.1	React	11
6.2.2	Material-ui	11
6.2.3	i18n	11
6.2.4	babel	11
6.2.5	webpack	11
6.3	rozhrani	11
6.3.1	sceny	11

6.3.2	komponenty	11
6.3.3	moduly	11
6.4	Lokalizace	11
7	moduly	12
8	provazani Backendu a Frontendu, API	13
9	instalace a spusteni	14
10	vysledny web	15
11	vyuziti	16
	Závěr	17
	Seznam použité literatury	18
	Seznam obrázků	19
	Seznam tabulek	20
	Seznam použitých zkratk	21
A	Přílohy	22
A.1	První příloha	22

Úvod

1. zadani

Cilem tohoto projektu je vytvoreni webového rozhrani pro ukladani a zobrazovani historickych i soucasnych zaznamu z oblasti Krkonos.

2. existující produkty

3. Vyber technologii

3.1 Frontend

Vzhledem k rychle se menicim trendum v oblasti webovych technologii, jsem se rozhodl jit cestou kterou vyvynul Facebook a jeho tym programatoru. Jedna se o technologii **Single page application**, jez je implementovana v knihovne **React**.

3.1.1 Single page application

Single page application je technologie umoznujici vykresleni jine stranky, bez nutnosti posilani requestu na server. Uzivatel si pri prvnim spusteni webu stahne cely balicek webu a pri opetovnem nascteni vetsinou saha jen do sve cache. Javascriptova knihovna (v tomto pripade React) pote stranku prekresluje pri uzivatelske interakci. V pripade nutnosti stazeni / posilani dat mezi serverem a uzivatelem (napr. editace zaznamu, nebo nacteni existujiciho zaznamu) se vola pouze request k API webové služby a telo requestu obsahuje pouze uzitecne informace.

3.1.2 React

Knihovna React je knihovna poskytujici single page application technologii. Jedna se o dobre udrzovanou knihovnu, jez byla vyvinuta Facebookem, jakozto nahrada zastaraleho konceptu renderovani stranky na serveru. Diky tomu servery nemuseli ztratet vykon s kazdou zmenou na strance a vykon k renderovani se bere z PC uzivatele. Jadro teto knihovny je velmi dobre optimalizovate a poskytuje i radu debugovacich nastroju, coz je pro vetsi projekty nepostradatelna vyhoda.

3.1.3 Dalsi mozne technologie

Velmi casto vykreslovani stranek probiha na serveru, se systemy jako jsou WordPress, psany PHP. Takovyto system je velmi dobre uzivatelsky privetivy, ale z pohledu vykonu ma velmi obrovsky overhead. V pripade implementace knihovniho systemu by to znamenalo vykreslovat celou stranku (hlavicku, telo i zapati) na serveru, na druhe strane single page application nic nerenderuje, pouze posle informaci o knize.

3.2 Backend

Mit single page aplikaci na frontendu znamena, ze na backendu musi existovat API, od ktereho bude frontend cerpat data. Navic zde potrebujeme i system pro staticke odesilani baliku cele webové stránky. V ramci udrzitelnosti jsem se rozhodl vyuzit jazyk Javascript stejny jako pro frontend. Express.js je knihovna ktera umoznuje komplexni spravu requestu a stala se tudiz jasnou volbou.

3.2.1 Express.js

Express.js poskytuje odesílání statických stránek (Reactiho balíku v našem případě), custom requesty pro rozmanité API a také odesílání a lokální ukládání statických souborů, jako obrázku, word i pdf dokumentu atd.

3.2.2 MongoDB

MongoDB je databázový systém typu non-sql. Což primárně znamená, že data neuchovává v tabulkách, ale v tzv. schématech. Což má mnoho výhod, největší je, že nekompletní záznamy nezabírají svými nevyplněnými daty místo v DB a ukládá se opravdu jen to, co je potřeba. Další výhodou je styl ukládání dat a komunikace s DB. Databáze si data uchovává ve formátu BSON (binární JSON rozšířený o datové typy). O data si aplikace žádá pomocí query, která je zcela odlišná od těch u sql-like databázi, primárně se zde neposílá query ve formátu string ale JSON, díky čemuž např. nenastane známá SQL injection. Znovu ve formátu JSON poté data vrací aplikaci.

3.2.3 Další možné technologie

Díky oddělení front-endu a back-endu (narozdíl např. u WordPressu) je možné na backend nainstalovat téměř cokoliv, co umí posílat requesty. Příkladem tomu mohou být skripty v jazycích PHP, C#, Python, nebo Perl. Ale vzhledem k tomu, že jedním z modulů bude neuronová síť na vyhledávání, vybíral jsem mezi Pythonem a JavaScriptem, protože jsou 2 jazyky, které mají velmi dobré knihovny pro práci s neuronovými sítěmi.

4. diagram systemu

5. implementace backendu

5.1 server

5.2 knihovny

5.2.1 Express.js

nadstavba pro nahravani souboru

5.2.2 mongoose

5.3 dokumentace

/api/documentation

5.3.1 vlastni mini knihovna pro dokumentaci

5.4 routes

5.4.1 uzivatel

5.4.2 autorizace

5.4.3 zaznam

5.4.4 stranka

5.4.5 nahravani souboru

5.5 models

5.5.1 zaznam

5.5.2 stranka

5.5.3 uzivatel

6. implementace frontendu

6.1 server

6.1.1 kompilace

6.1.2 npm

6.2 knihovny

6.2.1 React

6.2.2 Material-ui

6.2.3 i18n

6.2.4 babel

6.2.5 webpack

6.3 rozhrani

6.3.1 sceny

amin

cms

homepage

page

login

kontakt

search

show

edit

6.3.2 komponenty

KomboBox

Zapati

Navigacni menu

validationTextField

Uploadfile

Indexy

6.3.3 moduly

hologram

6.4 Lokalizace

7. moduly

8. provazani Backendu a Frontendu, API

9. instalace a spusteni

10. vysledny web

11. využití

Závěr

Seznam použité literatury

Seznam obrázků

Seznam tabulek

Seznam použitých zkratek

A. Přílohy

A.1 První příloha