



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

**BRNO UNIVERSITY OF TECHNOLOGY**

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

**FACULTY OF INFORMATION TECHNOLOGY**

**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

**DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA**

**WEBOVÁ APLIKACE PRO PŘEDPOVĚĎ POČASÍ**

**WEB APPLICATION FOR WEATHER FORECAST**

**BAKALÁŘSKÁ PRÁCE**

**BACHELOR'S THESIS**

**AUTOR PRÁCE**

**AUTHOR**

**KAMIL MUCHA**

**VEDOUCÍ PRÁCE**

**SUPERVISOR**

**Ing. BARTÍK VLADIMÍR, Ph.D.**

**BRNO 2019**

## Zadání bakalářské práce



21896

Student: **Mucha Kamil**  
Program: Informační technologie  
Název: **Webová aplikace pro předpověď počasí**  
**Web Application For Weather Forecast**  
Kategorie: Informační systémy

### Zadání:

1. Seznamte se s principy tvorby webových aplikací. Prostudujte také existující servery, ze kterých lze stahovat informace o předpovědi počasí. Zvolte alespoň dva servery vhodné pro tuto práci.
2. Analyzujte požadavky na webovou aplikaci pro předpověď počasí, která bude stahovat, uchovávat a aktualizovat data o předpovědi v databázi. Aplikace bude data zobrazovat různými způsoby, bude možné zvolit lokaci např. pomocí mapy a přesný čas od-do. Pokud se data pro zadanou lokaci v databázi nenacházejí, bude vyhledána nejbližší dostupná lokalita. Data budou získávána z více serverů poskytujících informace o počasí.
3. Navrhněte aplikaci zahrnující výše uvedené požadavky.
4. Zvolte vhodné implementační prostředí a implementujte navrženou aplikaci.
5. Na vzorku dat, vybraném po dohodě s vedoucím, ověřte funkčnost.
6. Zhodnoťte dosažené výsledky a diskutujte možnosti jeho dalšího rozšíření.

### Literatura:

- Welling, L., Thomsonová, L.: PHP a MySQL: rozvoj webových aplikací. Vyd. 1. Praha: SoftPress, 2003, 910 s. ISBN 80-86497-60-7.
- Žára, O.: JavaScript - Programátorské techniky a webové technologie, Computer Press, 2015. ISBN: 978-80-251-4573-9

Podrobné závazné pokyny pro vypracování práce viz <http://www.fit.vutbr.cz/info/szz/>

Vedoucí práce: **Bartík Vladimír, Ing., Ph.D.**  
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.  
Datum zadání: 1. listopadu 2018  
Datum odevzdání: 15. května 2019  
Datum schválení: 16. října 2018

## **Abstrakt**

Cieľom tejto bakalárskej práce je navrhnúť a implementovať webovú aplikáciu pre predpoveď počasia. K tomu bude potrebné získať dáta o predpovedi počasia z cudzích zdrojov, takže je kľúčové zvoliť vhodný zdroj dát. Tieto dáta budú uložené v databázi a budú pravidelne aktualizované. Na základe aktuálneho počasia si aplikácia taktiež bude vytvárať aj históriu počasia. Na webovej stránke budú tieto dáta zobrazené formou grafu, kde si užívateľ môže vybrať lokalitu a zvoliť si presný čas od-do pre ktorý sa dané dáta zobrazia. Pokiaľ pre danú lokalitu nie sú uložené žiadne dáta v databáze, je presmerovaný na najbližšiu lokalitu s dátami.

## **Abstract**

The aim of this bachelor thesis is to design and implement a web application for weather forecast. This will require collecting the data about weather forecast from public data sources, so it is crucial to choose the appropriate data source. These data will be stored in the database and will be updated on a regular basis. Based on the current weather, the app will also create weather history. On the web page, these data will be displayed in the form of a graph where user can select the location and choose the exact time from which the data will be displayed. If there is no data stored in the database for that location, it is redirected to the nearest location with available data.

## **Kľúčové slová**

Webová aplikácia, Predpoveď počasia, HTML, CSS, PHP, MySQL, SQL, JavaScript, databáza, haversine formula

## **Keywords**

Web application, Weather forecast, HTML, CSS, PHP, MySQL, SQL, JavaScript, database, haversine formula

## **Citácia**

MUCHA, Kamil. Webová aplikace pro předpověď počasí. Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Bartík Vladimír, Ph.D.

# **Webová aplikace pro předpověď počasí**

## **Prehlásenie**

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Bartík Vladimír Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Kamil Mucha  
16.5.2019

## **Pod'akovanie**

Týmto by som chcel poďakovať pánovy Ing.Vladimíru Bartíkovi, Ph.D za odbornú pomoc a rady pri návrhu a implementácií tejto práce a za možnosť pracovať na vlastnej téme.

# Obsah

<b>1 Úvod</b>	<b>3</b>
<b>2 Získavanie údajov</b>	<b>4</b>
2.1. Web API	4
2.2. Porovnanie vybraných zdrojov predpovede počasia	5
2.3. Apixu	5
2.4. Dark Sky	5
2.5. Open Weather Map	6
2.6. Weatherbit	6
2.7. AccuWeather	7
2.8. Yahoo!	7
2.9. World Weather Online	8
2.10. Zemepisné súradnice od Ip-Api.com	8
2.11. Zemepisné súradnice od Google APIs	8
<b>3 Grafický návrh</b>	<b>10</b>
3.1. Voľba farieb	10
3.2. Hlavička stránky	10
3.3. Telo webovej stránky	11
<b>4 Použité technológie</b>	<b>12</b>
4.1. HTML	12
4.2. CSS - Kaskádové štýly	12
4.3. JavaScript	13
4.4. SQL a MySQL	13
4.5. PHP	14
4.6. phpMyAdmin	14
4.7. Chart.js	15
4.8. Date Range Picker	16
4.9. Cron a Crontab	16
<b>5 Návrh a implementácia aplikácie</b>	<b>18</b>
5.1. Voľba lokalít	19
5.2. Návrh databáze	19
5.3. Stahovanie predpovede počasia	20
5.4. Zemepisné súradnice a výpočet vzdialenosti medzi nimi	21
5.5. Voľba času	23
5.6. Mini-aplikácia na zobrazenie predpovede počasia na 10 dní na základe lokácie návštevníka	23
5.7. Hlavný graf predpovede počasia	24
5.8. Mapa predpovede počasia	26
5.9. Webová administrácia	27
5.10. Zhrnutie a obsah web stránky	28
<b>6 Možné rozšírenia</b>	<b>29</b>

<b>7 Záver</b>	<b>30</b>
<b>8 Literatúra</b>	<b>31</b>
<b>9 Prílohy</b>	<b>34</b>

## Zoznam obrázkov a tabuliek

Tabuľka 2.1: Zdroje predpovede počasia. . . . .	5
Obrázok 3.1: Návrh palety farieb pre webovú aplikáciu. . . . .	10
Obrázok 3.2: Návrh dizajnu pre menu. . . . .	10
Obrázok 3.3: Návrh dizajnu pre header. . . . .	11
Obrázok 3.4: Mini-aplikácia predpovede počasia na 10 dní . . . . .	11
Obrázok 4.1: Zjednodušený HTML DOM zobrazujúci HTML elementy. . . . .	13
Obrázok 4.2: Ukážka chart.js grafov. . . . .	15
Obrázok 4.3: Date range picker. . . . .	16
Tabuľka 5.1: Vybrané lokality a jejich pomocné údaje . . . . .	19
Obrázok 5.2: ER diagram databáze. . . . .	20
Obrázok 5.3: Ukážka funkcie autocomplete od Google . . . . .	21
Obrázok 5.4: Výber lokality za pomoci mapy a značky od Google . . . . .	22
Tabuľka 5.5: Vzdialenosť vybraných lokalít od mesta Nové Zámky . . . . .	23
Obrázok 5.6: Nastaviteľný graf predpovede počasia. . . . .	26
Obrázok 5.7: Mapa predpovede počasia so sliderom ktorým sa dá posúvať v čase . . . . .	27
Obrázok 5.8: Náhľad do tela webovej administrácie. . . . .	28

# Kapitola 1

## Úvod

Existuje mnoho webových stránok ktoré ponúkajú predpoveď počasia. Avšak väčšina týchto stránok zobrazuje tieto údaje formou stĺpcov, kde jeden stĺpec zvyčajne zobrazuje počasie pre jeden deň, a detailnejšia predpoveď je dostupná až po rozkliknutí týchto stĺpcov.

K tomuto by som zvolil iný prístup, dáta o predpovedi počasia by bolo možné zobrazit' formou grafu ktorý by bolo možné jednoducho nastaviť a nechať si zobrazit' detailné dáta pre lokalitu a časové rozpätie ktoré užívateľa zaujímajú.

Hlavným problémom tejto aplikácie bude ale získanie dostatočne kvalitných dát o predpovedi počasia. Získanie týchto dát vyžaduje informácie z mnohých meteorologických staníc. K tomu je ešte potrebný vývoj matematického modelu pre výpočet predpovede počasia a dostatočná počítačová výpočtová sila ktorá by mohla tieto výpočty vykonať. Avšak existuje niekoľko firiem ktoré sa týmto už zaoberajú a niektoré poskytujú prístup k jejich celosvetovým dátam o predpovedi počasia zdarma. Preto by som skoršie zvolil tento prístup a dáta si pravidelne sťahoval z viacerých takýchto serverov a vhodne ich pri zobrazovaní kombinoval. Viac ohľadom dát o predpovedi počasia v kapitole 2.

Taktiež by bolo vhodné tieto dáta niekde skladovať, miesto toho aby som si ich musel neustále sťahovať z cudzích zdrojov za každým čo moju webovú aplikáciu niekto načíta. Preto bude potrebné riešiť problematiku návrhu databáze a parsovania, uloženia a aktualizácie dát do databáze.

Ďalším problémom ktorý bude potrebné riešiť budú lokality pre ktoré nebudem mať žiadne dostupné dáta. Pokiaľ si užívateľ zvolí túto lokalitu bude potrebné zistiť si jej zemepisné súradnice a poskytnúť mu predpoveď počasia pre najbližšiu lokalitu pre ktorú mám dostupné dáta v databázi.

Taktiež som si všimol že niektoré aplikácie pre predpoveď počasia využívajú mapy, preto ako rozšírenie mojej práce by som vypracoval mapu s počasím ktorá by sa dala ovládať jednoduchým bežcom ktorým by sa dalo posúvať v čase. Čiže týmto by som zobrazoval všetky lokality s nastavitelným časom.

Samotná implementácia projektu je opísaná v kapitole 5. Vzhľad webovej stránky je založený na grafickom návrhu z kapitoly 3, a pri tvorbe boli použité technológie ktoré sú opísané v kapitole 4. Kapitola 6 opisuje možné rozšírenia tohto projektu, a na záver v kapitole 7 sú opísané dosiahnuté ciele tohto projektu.

## Kapitola 2

# Získavanie údajov

Riešenie projektu nieje možné bez získania údajov z iných zdrojov. Hlavným cieľom bude získanie kvalitných dát o predpovedi počasia na niekoľko dní dopredu. Existuje viacero webových stránok a serverov ktoré tieto dáta ponúkajú, avšak väčšina z nich je prístupná iba za mesačné poplatky. Pre tento projekt som sa rozhodol vyhľadať niekoľko zdrojov ktoré sú buď to zdarma, alebo za dostupnú cenu. Najhlavnejší bude prístup k hodinovým záznamom o počasi na niekoľko dní.

Pre rozšírenejšie služby tohto projektu bude potrebné získať aj dáta o zemepisných súradniciach. Dáta o predpovedi počasia väčšinou zahŕňajú aj údaj zemepisnej šírky a dĺžky danej lokality ku ktorej sa vzťahujú. Na základe toho by som si mohol vytvoriť zoznam lokalít pre ktoré mám dostupné dáta.

Avšak ku kompletnému vypracovaniu aplikácie bude ešte potrebné získať zemepisné súradnice lokality pre ktorú si chce užívateľ zobraziť dáta v grafe. K tomuto by sme ešte pre rozšírenie mohli nájsť zdroj ktorý by nám zistil približnú lokáciu návštevníka webovej stránky a to napríklad na základe jeho ip adresy.

### 2.1. Web API

Web API (skratka z anglického pojmu Application Programming Interface) je rozhranie ktoré funguje na základe komunikácie typu dotaz-odpoveď. Umožňuje webovému serveru alebo prehliadaču získať informácie cez web, zvyčajne vo formáte JSON alebo XML, za použitia HTTP protokolu [1], [2].

Pre získavanie dát od ostatných serverov, ktoré budú neskôr zmienené, bude z našej strany potrebné si získať správne URL adresy. Prístupom na tieto URL adresy si dokážeme od serveru vyžiadať konkrétne dáta o ktoré máme záujem. Server túto požiadavku spracuje a pokiaľ sú dané dáta k dispozícii, budú predané vo vybranom formáte.

API key slúži pre identifikáciu užívateľa ktorý posielajú žiadosť. Tento údaj sa pridáva do URL adresy avšak nie každý server ho požaduje. Typicky je každému užívateľovi tento kľúč pridelený po registrácii alebo si ho je potrebné vygenerovať. Na základe tohto kľúča dokáže server rozpoznať kto si žiada aké dáta a môže týmto aj monitorovať počet prístupov k dátam.



## 2.2. Porovnanie vybraných zdrojov predpovede počasia

Všetky vybrané zdroje poskytujú aspoň základné údaje o počasí ako je:

- teplota,
- rýchlosť a smer vetra,
- oblačnosť,
- atmosférický tlak,
- vlhkosť,
- zrážky,
- textový údaj opisujúci počasie.

Pri hľadaní dostupných zdrojov som sa snažil hľadať hlavne tie, ktoré poskytujú hodinové záznamy o predpovedi počasia. Taktiež sú dôležité aj limity ktoré treba dodržiavať pri sťahovaní týchto dát, v tabuľke 2.1 sú zobrazené tieto limity pre jednotlivé servery a jednotlivé cenové ponuky ktoré som sa rozhodol využiť.

Tabuľka 2.1: Využitie zdroje predpovede počasia

	Apixu	Dark Sky	Open Weather Map	Weatherbit	Accu-Weather
predpoveď na	10 dní	7 dní	5 dní	2 dni	12hodín
Interval predpovede	1 hod	1 hod	3 hod	1 hod	1 hod
Limity na volania API	50000/mesiac	1000/deň	60/min a 7200/deň	1000/deň	50/deň
Využitá cenová ponuka	\$4.95/mesiac	zdarma	zdarma	zdarma	zdarma

## 2.3. Apixu

Apixu ponúka celosvetovú predpoveď počasia a taktiež poskytuje aj niektoré zemepisné funkcie [28]. Zdarma je však dostupná iba predpoveď počasia na 7 dní v jednodňových intervaloch. Mňa ale najviac zaujíma predpoveď počasia v hodinových intervaloch, a tá je dostupná v balíčku “Silver” za \$4.95 čo je celkom prijateľná cena. S týmto balíkom sa taktiež zvyšuje limit na volania API z 10000/mesiac na 50000/mesiac. K tomu je taktiež prístupná história počasia na 30 dní dozadu.

Zemepisné funkcie a históriu počasia som sa rozhodol nevyužiť, avšak predpoveď počasia z balíku “Silver” mi pomôže ako jeden zo zdrojov dát a taktiež si z neho vytvorím vlastnú históriu.

## 2.4. Dark Sky

Dark Sky hodnotím ako najlepší zdarma dostupný zdroj dát o predpovedi počasia. Ponúkajú totiž hodinovú predpoveď počasia na 7 dní ku ktorej je možné pristupovať zdarma 1000 krát za deň [29]. Oproti konkurencii používajú iný model platenia pri ktorom má užívateľ prístup ku

všetkému a platí iba pokiaľ chce prekročiť limit 1000 volaní za deň. Po vyčerpaní limitu stojí každý ďalší prístup \$0.0001, čiže ďalších 10 tisíc prístupov za dolár. K ponuke majú taktiež rozsiahlu históriu počasia, aktuálne počasie, a počasie minútu po minúte pre nasledujúcu hodinu. Taktiež je možné pri volaní API využiť zemepisné súradnice a nieje potrebné zisťovať názvy alebo identifikačné čísla lokalít.

Tento zdroj predpovedí počasia mi prišiel ako najlepší. Kvalitne spracovaná dokumentácia, zdarma prístup k detailnej predpovedi počasia na 7 dní, voľnejšie limity na volanie API a jednoduchý platobný model. Preto som sa rozhodol tieto dáta využiť do hlavného grafu, ale aj pre mapu počasia.

## 2.5. Open Weather Map

Open Weather Map poskytuje velice podrobné a bohaté dáta o aktuálnom počasi a o predpovedi počasia. Vo svojej databáze majú vyše 200 000 lokalít a jejich služby sú založené na údajoch z vyše 40 000 meteorologických staníc [4].

Zdarma majú v ponuke aktuálne počasie a predpoveď počasia na 5 dní pre všetky dostupné lokality. Predpoveď na 5 dní obsahuje informácie o predpokladanom počasi v trojhodinových intervaloch, týmto môžeme získať až 40 záznamov pre jednu lokalitu. Taktiež ešte poskytujú údaje pre mapu počasia, UV index lokalít a znečistenie vzduchu lokalít. Na všetky tieto dáta sa síce dá pristupovať zdarma ale interval prístupov je obmedzený. Limity sú momentálne nastavené na 7200 prístupov za deň a 60 prístupov za minútu. Taktiež pristupovať na rovnakú lokalitu sa dá iba raz za 10 minút.

Spoplatnená ponuka je rozdelená na viacero kategórii, pre “Startup” ktorý je dostupný za 40\$ na mesiac je k dispozícii predpoveď počasia na 16 dní, avšak táto predpoveď obsahuje iba jeden záznam na každý deň. K tomu je ešte zvýšenie limitu prístupov z 60 za minútu na 600 za minútu. Pre platobný plán “Developer” ktorý je za 180\$ na mesiac je zvýšený limit prístupov na 3 000 prístupov za minútu, ale najzaujímavejším je “Professional” ktorý nielen že zvyšuje limit na 30 000 prístupov za minútu ale povoľuje aj funkciu “Bulk download”, s ktorou je možné sťahovať viac lokalít naraz. Pokiaľ by sme chceli našu aplikáciu rozšíriť o väčší počet lokalít bolo by vhodné získať prístup k tejto funkcii. Taktiež tento platobný plán ponúka prístup k predpovedi počasia na 16 dní kde sú záznamy pre každú hodinu. Tým by sme mohli získať až 384 záznamov pre každú lokalitu.

Open Weather map by som hodnotil ako velice dobrý zdroj dát vhodný pre zložitejšie aplikácie. Najhlavnejšia výhoda tohto zdroja je široký objem dát ktorý zahrňuje všetky údaje o počasi ktoré by správna webová aplikácia o predpovedi počasia mala poskytovať. Zdarma je možné získať predpoveď na 5 dní kde sú záznamy rozložené v trojhodinových intervaloch. Avšak pri bezplatnom užívaní tejto služby, je počet prístupov k dátam cez web API limitovaný.

Kvôli objemu a rozsahu dát som sa rozhodol použiť tento zdroj, ponúkané záznamy naozaj obsahujú všetky potrebné informácie o predpovedi počasia, aj keď záznamy sú iba na 5 dní dopredu v trojhodinových intervaloch.

## 2.6. Weatherbit

Weatherbit zdarma ponúka prístup k hodinovým záznamom predpovede počasia na dva dni [30]. K tejto predpovedi je možné pristupovať 1000 krát za deň, takže tento limit by zatiaľ

nemal robiť žiadne problémy. V prípade rozšírenia o nové lokality je možné zvýšiť limit na 50000/deň za \$35, a pri Advanced balíčku je za \$470 možné už získať hodinovú predpoveď na 4 dni.

## 2.7. AccuWeather

AccuWeather má značne limitovanú ponuku zdarma [33]. Toto API je možné použiť iba 50 krát za deň, a hodinové záznamy sú dostupné iba na 12 hodín. Aj napriek tomu som sa rozhodol využiť tento zdroj s tým, že bude nutné prejsť z hodinových aktualizácií na osem hodinové aktualizácie. Takže tento zdroj sa mi bude hodiť aspoň na históriu počasia, a neskôr by bolo možné prejsť na jeden zo spoplatnených balíčkov tohto zdroja a predpoveď na viacej dní.

Z týchto spoplatnených balíčkov ma zaujali hlavne “Prime” a “Elite” balíčky. Prime je za \$250 a obsahuje už hodinovú predpoveď na 3 dni s limitom až 1,8 milión API volaní za mesiac. U Elite je ponuka na hodinovú predpoveď na 4 dni s limitom 2,4 milióna API volaní za mesiac, a to za \$500.

## 2.8. Yahoo!

Yahoo poskytuje prístup k celosvetovým dátam o počasi. Tieto dáta sú verejne dostupné a je možné ich sťahovať bez registrácie. Avšak predpoveď počasia je velice obmedzená a v ponuke je iba predpoveď na 10 dní ktorá zahŕňa najvyššiu a najnižšiu teplotu pre každý deň a textový údaj opisujúci počasie v danom dni.

YQL Query (Yahoo! Query Language) slúži pre vyhľadávanie, filtrovanie a kombinovanie dát od Yahoo zdrojov [3]. Používa syntax ktorá je velice podobná SQL. Za pomoci tejto platformy si dokážeme vybrať dáta o ktoré máme záujem, takže napríklad aktuálne počasie a predpoveď na 10 dní pre mesto Brno v metrických jednotkách by sme mohli získať cez

```
select * from weather.forecast where woeid = 786869 AND u = 'c'
```

Tento select ale ešte musíme vložiť do aplikácie <https://developer.yahoo.com/weather/> ktorá nám vygeneruje verejný link cez ktorý je už možné získať aktuálne dáta pre mesto Brno. (Ukážka výsledných dát v prílohách)

V kóde select bola zmienaná položka WOEID (Where On Earth IDentifiers), toto je jednoznačné číselné označenie mesta ktoré Yahoo! používa pri ukladaní dát do jejich databáze. Takže miesto písania názvu miest textovou formou na ne môžeme jednoznačne ukázať cez jejich WOEID. Pre mesto Brno to je číslo 786869 a ostatné mestá môžeme nájsť napríklad na <http://woeid.rosselliot.co.nz/lookup>.

Yahoo! by som odporučil ako zdroj dát pre nejaké jednoduchšie aplikácie, prístup k dátam je velice jednoduchý, stačí si vygenerovať verejný odkaz z ktorého sa dáta dajú sťahovať. Limit prístupov k dátam tiež nieje žiadny problém. Je dovolených až 2000 prístupov za deň a hlavne vďaka tomu že je možné si stiahnuť dáta pre viacero miest naraz, by kvôli tomuto limitu nemali vznikať vôbec žiadne obmedzenia. Predpoveď počasia je ale velice obmedzená, toto je hlavný dôvod prečo som sa rozhodol tento zdroj dát nevyužiť.

## 2.9. World Weather Online

World Weather Online [5] poskytuje kompletné celosvetové dáta o aktuálnom počasi a o predpovedi počasia až do 14 dní. K tomuto majú v ponuke aj históriu počasia a sústredia sa taktiež aj na predpoveď počasia pre surferov, námorníkov, ale aj horolezcov a lyžiarov. K tomuto používajú iný typ záznamov, ktorý pre lokality na mori tieto záznamy obsahujú podrobnosti o vlnách ako perióda a výška vln a záznamy v horských lokalitách ponúkajú údaje ako očakávané sneženie, šanca na sneženie a úroveň námraze.

Pre začiatok je možné využiť 60 dní zdarma na vyskúšanie, ale potom je potrebné prejsť na platobný plán pri ktorom je možné si povoliť iba funkcie o ktoré máme záujem [6]. Takže podrobná predpoveď počasia na 14 dní ku ktorej by sme mohli pristupovať 5 000 krát denne by nás podľa aktuálnej ceny vyšla na 43,13€. Takáto podrobná predpoveď počasia sa skladá zo záznamov ktoré sú usporiadané v hodinových intervaloch. Týmto by sme pre každú lokalitu dostali až 336 záznamov.

Dáta ktoré tento zdroj poskytuje zahŕňajú úplne všetko ohľadom počasia a predpovedi počasia. Dokonca sú k dispozícii aj záznamy pre aktuálne počasi a predpoveď počasia na mori a na horách. Tento zdroj dát by bol perfektný pre tento projekt, avšak kvôli nutnosti prejsť na spoplatnený režim po 60 dňoch používania som sa rozhodol tento zdroj dát nevyužiť vo svojej práci.

## 2.10. Zemepisné súradnice od Ip-Api.com

Ip-Api.com je jednoduchá webová aplikácia ktorá slúži ako nástroj pre získanie lokácie podľa IP adresy. Lokácia väčšinou smeruje na sídlo alebo lokálnu pobočku poskytovateľa internetového spojenia ktorý danú IP adresu spravuje, preto je dosť možné že výsledky tohto nástroja nebudú až tak presné a budú často smerovať na väčšie mestá. Na základe IP adresy sa dá teda získať údaje o poskytovateľovi ktorý danú IP adresu spravuje. Tieto údaje zahŕňajú:

- plný názov a skratku štátu, kraja a mesta,
- zemepisné súradnice,
- časová zóna,
- a podrobnosti poskytovateľa internetového pripojenia.

Tieto údaje sa dajú stiahnuť cez web API. Sú dostupné vo formáte JSON, XML, CSV, Newline Separated, Serialized PHP. Pre jejich stiahnutie vo formáte JSON je možné použiť URL <http://ip-api.com/json/<ip adresa>>. Na túto službu je ale limit 150 prístupov za minútu [7].

Túto službu by bolo vhodné využiť na zistenie približnej lokality návštevníka webovej stránky, avšak je treba brať ohľad na to že výsledky nie sú až tak presné a dajú sa oklamať používaním VPN.

## 2.11. Zemepisné súradnice od Google APIs

Google vlastní a spravuje mnoho projektov [8]. Jedným z nich je Google Maps [9]. Kvôli tomuto projektu si Google mapuje celý svet, ukladajú si údaje o mestách, uliciach, podnikoch a rôznych iných zaujímavých lokalitách. Ku všetkým týmto informáciám je možné získať dočasný prístup zdarma cez web API (je ale potrebný API kľúč) a tohto by sme mohli využiť

pre získanie zemepisných súradníc lokalít o ktoré majú návštevníci webovej aplikácie záujem. Kompletne zemepisné údaje pre mesto Brno by sme teda mohli vyhľadať za pomoci funkcie geocoding prístupom na toto URL. (Ukážka výsledných dát v prílohách)

```
https://maps.googleapis.com/maps/api/geocode/json?address=Brno,+Czechia&key=KEY
```

Google taktiež ponúka dočasný prístup zdarma k omnoho viacej funkciám a dátam. Pre tento projekt budú stačiť funkcie geocoding, autocomplete a samotné mapy google maps. Zmienená funkcia autocomplete [10] napomáha užívateľom správne zadať názov lokality do vyhľadávania. Využitím tejto funkcie znížime počet chybne zadaných lokalít a napomôžeme používateľovi zadať názov mesta. Navyše zaistíme že pokiaľ si užívateľ vyberie jednu z ponúkaných lokalít tak bude určite správne zadaná a dostupná pre ostatné funkcie ako geocoding.

## Kapitola 3

# Grafický návrh

K webovej stránke by som rád vypracoval svoj vlastný dizajn. Ako základom bude potrebné vhodne zvoliť farby a navrhnuť štruktúru stránky tak, aby každý plánovaný prvok mal dostatok potrebného priestoru. Návrh dizajnu som vypracoval cez program Adobe Photoshop, a následne som implementoval CSS na základe tohto návrhu.

### 3.1. Voľba farieb

Pre webovú aplikáciu pre predpoveď počasia by som zvolil bledo modrú farbu k téme a niekoľko čiernobielych farieb pre eleganciu webovej stránky.

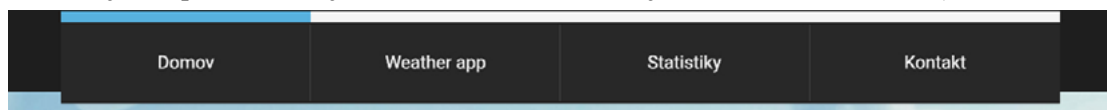


Obrázok 3.1: Návrh palety farieb pre webovú aplikáciu

- Modrá ..... #59acde
- Čierna ..... #111111
- Blede čierna ..... #1a1a1a
- Sivá ..... #2a2a2a
- Biela ..... #f3f3f3

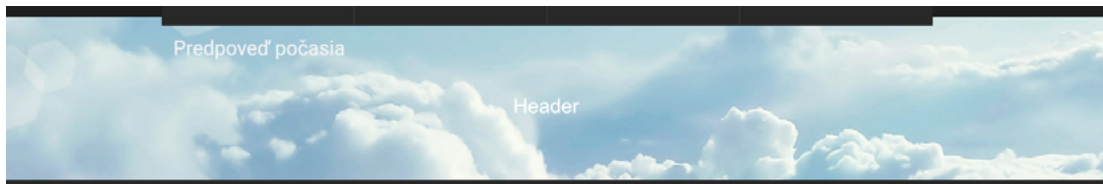
### 3.2. Hlavička stránky

Štruktúra webovej stránky bude celkom jednoduchá, jadrom celej práce je graf s predpoveďou počasia pre ktorý musíme vyhradiť dostatok miesta. Preto som sa rozhodol vypracovať dizajn pre vodorovné menu. Týmto by sme mali ušetriť nejaké miesto v tele stránky. (text na obrázku je iba placeholder, jedná sa iba o obrázok dizajnu, nie o finálnu stránku)



Obrázok 3.2: Návrh dizajnu pre menu

Pre header som sa rozhodol použiť obrázok ako pozadie a na to pridať text ako placeholder, na toto miesto by neskôr mohlo prísť logo a názov firmy.



Obrázok 3.3: Návrh dizajnu pre header

Telo webovej stránky bude rozdelené horizontálne na dve časti, prvá bude zobrazovať predpoveď počasia od Yahoo! na 10 dní. K tejto predpovedi nebudú žiadne nastavenia, bude to len rýchle zobrazenie počasia na základe lokácie užívateľa. Druhá časť bude hlavný graf ktorý bude využívať dáta získané z Open Weather Map. Nad grafom budú nástroje ktorým sa bude dať ovládať. Toto by malo zahŕňať vyhľadávanie lokalít, voľba časového okna a voľba typu informácie (teplota, vietor, dážď...). Čiže bude možné zobraziť všetky typy informácií v jednom grafe a pár tlačítkami si prepínať medzi teplotou, vetrom, zrážkami, oblačnosťou a vlhkosťou ovzdušia.

### 3.3. Telo webovej stránky

Do tela webovej stránky príde nastaviteľný graf s predpoveďou počasia ktorý bude zobrazovať dáta od Open Weather Map, ale ešte predtým by bolo dobré vytvoriť nejakú menšiu aplikáciu pre rýchle zobrazenie predpovede počasia. Táto mini-aplikácia by bola vhodná ako prototyp u ktorého by som taktiež otestoval funkcie zistenia lokality návštevníka a funkciu pre zistenie vzdialenosti od lokalít s dátami. K tomuto by bolo dobré využiť dáta od Yahoo! ktoré obsahujú predpoveď na 10 dní ktorá zahŕňa najnižšiu a najvyššiu teplotu dna.

Predpoveď na 10 dní podľa vašej lokácie, Zlín, 143km.									
14 Máj	15 Máj	16 Máj	17 Máj	18 Máj	19 Máj	20 Máj	21 Máj	22 Máj	23 Máj
21°/13°	18°/10°	17°/10°	16°/11°	18°/10°	18°/9°	21°/10°	22°/11°	23°/13°	25°/15°

Obrázok 3.4: Mini-aplikácia predpovede počasia na 10 dní

Pre hlavný graf bude potrebné presne vybrať aké k nemu chceme nastavenia. Určite bude potrebný výber lokality, k tomuto by sme mohli využiť `<input>` okienko s funkciou autocomplete a vstup posielat' na geocoding od google. Pre časové rozpätie by bolo vhodné vybrať nejaký nástroj ktorý by zobrazil interaktívny kalendár pre výber dňa a času. A na záver by bolo vhodné vytvoriť sadu zistíme ktoré by menili typ dát ktoré graf zobrazuje. Pre tieto tlačítka by som 5 kategórii:

- teplota
- vietor
- zrážky
- vlhkosť
- oblačnosť

## Kapitola 4

# Použité technológie

Pred tým než sa dostaneme k samotnej implementácii tohto projektu, rád by som zmienil a vysvetlil základné údaje a princípy technológií ktoré boli využité.

### 4.1. HTML

HTML (skratka pre HyperText Markup Language) je značkový jazyk pre tvorbu webových stránok. Hlavné využitie tohto jazyka je zadanie obsahu webovej stránky.

Obsah je rozdelený na jednotlivé HTML elementy. Tieto elementy sú vymedzené značkami (anglicky tags), ktoré sú odlišené od textu tým, že sú vložené do znakov '<' určujúci začiatok značky a znaku '>' určujúci ukončenie značky. Niektoré z týchto značiek ako `<input />` a `<br />` tvoria jeden HTML element a priamo pridávajú nejaký obsah do webovej stránky. Iné, párové značky, ako `<h1>` iba obklopujú nejaký text alebo iné HTML elementy a pridávajú mu nejaké vlastnosti.

Tvorba webovej stránky si vyžaduje niekoľkých HTML elementov z ktorých bude zložená základná štruktúra stránky. Ako prvé je ale potrebné deklarovať typ dokumentu značkou `<!DOCTYPE html>`. Zvyšok HTML obsahu bude obalený párovou značkou `<html>`. Vnútri tejto značky bude značka `<head>` kde je možné špecifikovať názov stránky, popis, autor, kódovanie a mnoho ďalšieho. Za ňou bude nasledovať značka `<body>` ktorá už bude obsahovať samotný viditeľný obsah webovej stránky.

### 4.2. CSS - Kaskádové štýly

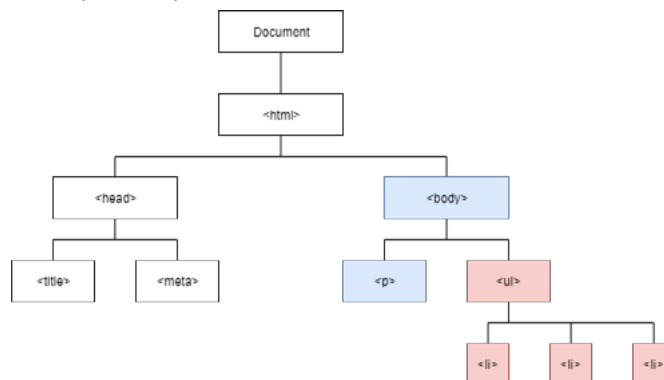
CSS (skratka z anglického pojmu Cascading Style Sheets) vzniklo v roku 1996 ako rozšírenie pre HTML. V tomto čase bolo problémom že HTML kód obsahoval štruktúru webovej stránky, ale aj jej vzhľad. Týmto sa stával kód dosť často neprehľadný, bolo ťažké ho upravovať a vznikalo veľa duplicitného kódu pri viacerých prvkoch s rovnakým vzhľadom. Preto bolo potrebné tieto dva prvky oddeliť. Týmto vznikli kaskádové štýly a jejich hlavným zameraním bola čisto prezentácia dát.

Kaskádové štýly je možné vytvárať v hlavičke HTML súboru, avšak omnoho lepšie je vytvorenie externého .css súboru. Takýto externý CSS súbor je možné pripojiť k viacerým HTML súborom značkou `<link>` v hlavičke. Tým môžeme zabezpečiť rovnaký vzhľad všetkých podstránok bez nejakého duplicitného kódu.

CSS, okrem toho že oddeľuje vzhľad od štruktúry, taktiež pomáha v aplikácii štýlov. Aplikácia štýlov je spracovaná vo forme selektorov. Predtým než ale spomeniem ako sa tieto



selektory aplikujú na HTML je potrebné si vysvetliť pojem DOM (skratka pre anglický pojem Document Object Model). DOM dokáže zobrazíť HTML ako stromovú štruktúru elementov, atribútov a textu. V hornej vrstve je koreň stránky a strom sa postupne rozvetvuje a znázorňuje všetky prvky stránky až do najspodnejšej úrovne. Pri aplikácii CSS sa niektoré vlastnosti prenášajú na prvky do nižších úrovní, ako napríklad veľkosť a farba textu. Avšak toto prenášanie je možné predvídať, pretože má vždycky prednosť ten selektor, ktorý priamo zachytáva prvok, alebo zachytáva prvok ktorý je najbližšie smerom nahor. Toto som sa snažil znázorniť na obrázku 4.1. Aplikáciou modrej farby textu na `<body>`, sa modrá farba neprenesie na `<ul>`, pokiaľ aplikujeme červenú farbu priamo na značku `<ul>`. [17] Takýmto spôsobom získame flexibilitu pri úpravách a aplikácii týchto štýlov.



Obrázok 4.1: Zjednodušený HTML DOM zobrazujúci HTML elementy

### 4.3. JavaScript

JavaScript je používaný ako interpretovaný programovací jazyk, to znamená že jeho kód je možné priamo vykonať bez kompilácie. Spolu s HTML a CSS tvoria trojicu základných technológií pre tvorbu webových stránok [18].

JavaScript zaručuje dynamickú stránku webových stránok a aplikácii. Vykonáva logické a matematické operácie, prácu s premennými, má prístup k údajom o prehliadači ktorý návštevník webu používa a mnoho ďalšieho. Taktiež má prístup k DOM štruktúre webovej stránky a tým vie pristupovať k jednotlivým prvkom webovej stránky ako text, obrázky, štýly a upravovať ich [19].

Kód JavaScriptu sa vykonáva na strane klienta a to až po stiahnutí stránky. Kvôli tomu je možné naviazať nami vytvorené funkcie na nejaké udalosti na webovej stránke, ako napríklad kliknutie na tlačítko, stlačenie klávesy, alebo posun myšou. Týmto vznikajú ale nejaké obmedzenia, užívateľovi je vždy poslaný celý zdrojový kód aby si ho mohol spustiť, lenže dá sa do neho ľahko nahliadnuť. Taktiež JavaScript nedokáže pracovať so súbormi, pretože by tým mohol ohroziť bezpečie užívateľa.

### 4.4. SQL a MySQL

MySQL je open-source systém správy relačnej databáze. Samotná databáza je tvorená štruktúrou jednej alebo viacerých tabuliek. Stĺpce tabuliek vyjadrujú jeden typ dát. Takýto typ dát má meno a data typ (int, float...). A každý riadok v tabuľke je jednotlivý záznam s dátami. Tieto

tabuľky majú medzi sebou nejaké vzťahy ktoré sú založené na cudzích kľúčoch. Za pomoci týchto cudzích kľúčov je možné tabuľky spojovať, napríklad v našom prípade je možné spojiť každý jeden záznam z *open\_weather\_map\_5d* tabuľky s práve jedným záznamom z *cities\_geo* a tým k údajom o počasí pripojíme podrobnosti o lokalite. Takže tým že skladujeme mestá a predpoveď počasia oddelene, ale ponechávame iba nejakú reláciu medzi nimi, sa vyhneme duplicitným informáciám v databázy [20].

SQL (Structured Query Language) je skratka pre štruktúrovaný dotazovací jazyk. Tento jazyk je využívaný pre prácu s relačnými databázami. Príkazy SQL sa delia na tri základné skupiny a to:

- správa tabuliek (CREATE, DROP...)
- manipulácia dát (SELECT, UPDATE, INSERT, DELETE...)
- správa užívateľov a jejich práv (GRANT, CREATE USER...)
- a ostatné

Takže za pomoci SQL si dokážeme vytvoriť tabuľky databáze, pridať hodnoty do tabuliek a neskôr ich čítať alebo meniť, prípadne upravovať samotné tabuľky.

Dôležitý je ale príkaz SELECT. Tento príkaz nám velice uľahčuje prácu pri výbere dát o ktoré máme záujem, pretože je možné presne špecifikovať ktoré dáta majú byť vybrané, ako majú byť zoradené a čo musia obsahovať aby boli vybrané.

Takže pre nejaké menšie aplikácie si možno vystačíme s ukladaním údajov do textového súboru. Avšak pri väčších projektoch ktoré musia mať niekde trvalo uložené dáta a vyžadujú ľahký prístup k čítaniu a modifikácii týchto dát je omnoho lepšie využiť takéhoto databázového systému.

## 4.5. PHP

PHP je open-source skriptovací jazyk. Tento jazyk bol vytvorený hlavne pre vývoj webových stránok, avšak je často používaný aj ako univerzálny programovací jazyk. Dokáže pracovať s premennými a vykonávať logické a matematické operácie.

PHP pracuje na strane servera, to znamená, že v prípade webových stránok sa všetky php skripty vykonajú ešte pred odoslaním webovej stránky návštevníkovi. Kvôli tomuto PHP dokáže na servere pracovať so súborami. Ale taktiež vďaka tomu že sa návštevníkovi webovej stránky neposiela zdrojový kód ako u JavaScriptu, je bezpečné s ním pristupovať k databázi servera. Nevýhodou tohto je, že PHP funkcie už nieje možné naviazať na udalosti webovej stránky ako stlačenie tlačítka či posun myšou, takže v tejto oblasti PHP JavaScript nenahradí [21], [22].

PHP kód je možné priamo vkladať do HTML štruktúry, je avšak potrebné potom už zmeňovať koncovku súboru na .php. V tomto kóde je možné dynamicky tvoriť HTML štruktúru, meniť štýly, pracovať s databázou, predávať premenné do JavaScriptu a omnoho viac.

## 4.6. phpMyAdmin

phpMyAdmin je voľne dostupný open source nástroj pre správu MySQL a MariaDB databáze [23]. Tento nástroj je vytvorený v jazyku PHP. Samotná správa databáze je spracovaná ako webová aplikácia, v ktorej je možné vykonať často používané funkcie nad MySQL databázou cez užívateľské rozhranie, ale taktiež je prístupné aj manuálne písanie a vykonávanie SQL

príkazov cez toto rozhranie. K tomuto všetkému je zverejnená dokumentácia ktorá opisuje všetky prvky tohto nástroja a jejich používanie [24].

## 4.7. Chart.js

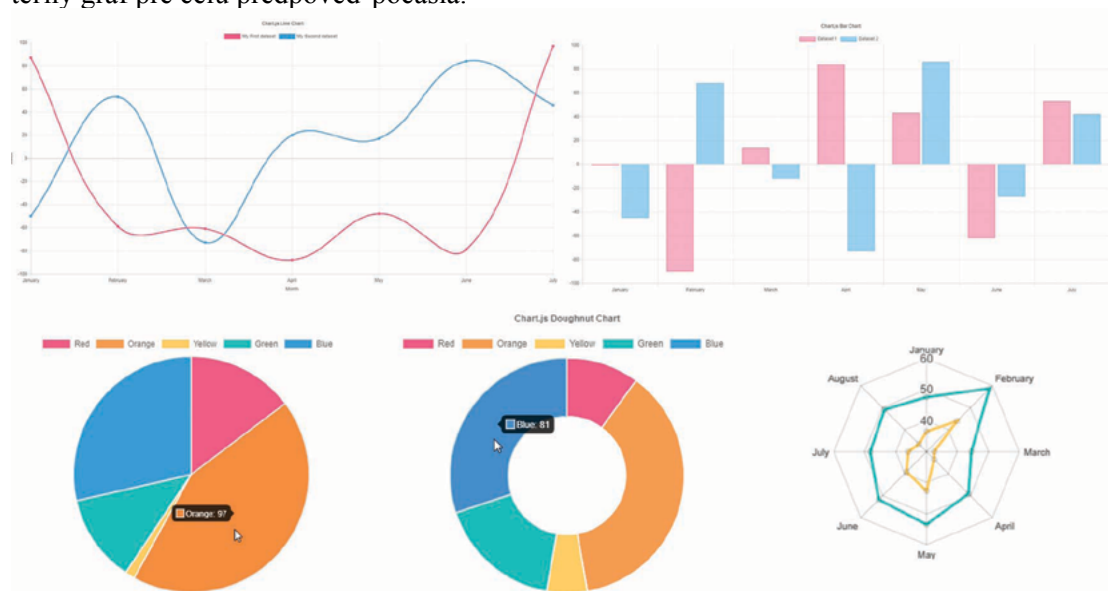
Chart.js je nástroj ktorý uľahčuje tvorbu grafov pre webové stránky. Je napísaný v JavaScripte. Verzia 2.7.2 ktorú budem pre tento projekt používať sa dá velice jednoducho získať kódom:

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.7.2/Chart.bundle.js"></script>
```

ktorý je potrebné napísať do HTML kódu webovej stránky. V HTML štruktúre potom musíme vymedziť priestor pre graf tagom <canvas> do ktorého je možné vložiť akýkoľvek typ grafu. Na výber sú grafy typu:

- linerárny
- stĺpcový
- kruhový
- koblihový
- graf rozptylu
- graf polárnej oblasti

Niektoré z týchto typov grafov majú taktiež viacej podtypov, hlavne lineárny a stĺpcový graf. K tomu je k nim pekný výber nastavení takže je možné ich jednoducho upraviť a modifikovať aby ľahšie dokázali vyjadriť informácie ktoré sa snažíme prezentovať. Ale hlavne pre naše potreby v tomto projekte je užitočné to, že sa dajú dynamicky meniť dáta ktoré zobrazujú, čiže miesto niekoľko grafov ktoré by obsahovali iba jeden typ informácie nám bude stačiť jeden predstaviteľný graf pre celú predpoveď počasia.



Obrázok 4.2: Ukážka chart.js grafov

## 4.8. Date Range Picker

Zadávanie textových a číslcových údajov na webovej stránke je problematické, od užívateľa môže prísť k preklepu, alebo v horšom prípade je potrebné dodržať presný formát v akom vstup musí byť. Toto je problematické hlavne u dátumov s časom. Jednoduchým riešením by mohlo byť rozdelenie okienok do viacerých, tak aby každý údaj ako mesiac, deň, hodina, minúta, mali zvlášť svoje okienko vstupu. Existuje ale viacero JavaScriptových nástrojov ako date range picker ktorý modifikujú tieto vstupné okienka a obohatia ich o interaktívny kalendár na ktorom je možné si dátum naklikať. Týmto je zadávanie presného časového rozpätia omnoho pohodlnejšie pre užívateľa a po výbere z kalendára je zaručené že čas bude zadaný v správnom formáte.

Navyše nástroj date range picker je možné jednoducho nastaviť, zadať východzí čas a dátum pri načítaní stránky. Je možné si taktiež upraviť názvy dní v týždni, názvy mesiacov a text v tlačítkach. U dátumu a času si je možné nastaviť vlastný formát, toto bude hlavne užitočné pokiaľ chceme dosiahnuť plného prekladu front-endu tohto nástroja do slovenského jazyka. V tom prípade by formát dátumu a času bol zapísaný ako DD.MM.YYYY HH:mm.

Čas:

03.05.2018 00:53 - 06.05.2018 00:53

03.05.2018 00:53 06.05.2018 00:53 Uložiť Zrušiť

0 : 00 0 : 00

< Máj 2018 Jún 2018 >

Po	Ut	St	Št	Pi	So	Ne	Po	Ut	St	Št	Pi	So	Ne
30	1	2	3	4	5	6	28	29	30	31	1	2	3
7	8	9	10	11	12	13	4	5	6	7	8	9	10
14	15	16	17	18	19	20	11	12	13	14	15	16	17
21	22	23	24	25	26	27	18	19	20	21	22	23	24
28	29	30	31	1	2	3	25	26	27	28	29	30	1
4	5	6	7	8	9	10	2	3	4	5	6	7	8

Obrázok 4.3: Date range picker

## 4.9. Cron a Crontab

Cron je program ktorý je používaný v operačných systémoch Unix, Solaris a Linux. Jeho úlohou je spúšťanie príkazov z tabuliek Crontab v určitých intervaloch. To vykonáva tým že každú minútu nahliada do týchto tabuliek a kontroluje či je potrebné niektorý z príkazov vykonať. Tohto by sme mohli využiť pri php kódach pokiaľ ich chceme spúšťať napríklad každú hodinu [25], [26].

Crontab (CRON TABLE) sú, ako už bolo zmienené, tabuľky v ktorých je uložený interval spúšťania a konkrétny príkaz ktorý je potrebné vykonať v daných intervaloch. K modifikácii týchto tabuliek sa využíva príkaz *crontab -e* ktorým sa z príkazovej riadky dostaneme do editora tabulky prihláseného používateľa.

## Kapitola 5

# Návrh a implementácia aplikácie

Ako prvé bolo potrebné vybrať iba nejaký menší počet miest či lokalít. Servery ktoré poskytujú údaje o predpovedi počasia sú väčšinou značne obmedzené, a taktiež počasie sa nezvykne až tak moc líšiť pri neakých malých vzdialenostiach. Preto som si zo začiatku zvolil krajské mestá Českej republiky [Kapitola 5.1].

Ďalším dôležitým krokom bol návrh databáze. Tu budú uložené zvolené lokality, záznamy predpovede počasia a prihlasovacie údaje do administrácie webu [Kapitola 5.2].

Pri implementácii jednotlivých skriptov v jazyku PHP ktoré majú zabezpečiť stahovanie a aktualizovanie záznamov o predpovedi počasia som hlavne bral ohľad na možné rozšírenie o nové lokality, preto pri každej aktualizácii si tieto skripty najprv načítajú tabuľku s mestami a na základe nej urobia aktualizáciu záznamov z dát získaných cez API [Kapitola 5.3].

Ďalším problémom ktorý bolo potrebné vyriešiť boli lokality pre ktoré dáta nestahujem. V databázi si ku každému mestu ukladam presné zemepisné súradnice, a s použitím nástrojov od google by bolo možné vypracovať mapu v ktorej sa bude dať zakliknúť lokalita o ktorú má užívateľ záujem, prípadne taktiež pridať aj okienko pre textové vyhľadávanie. Tým získam zoznam súradníc pre ktoré mám dáta a súradnice o ktoré má užívateľ záujem. K výpočtu vzdialenosti medzi dvojicou zemepisných súradníc som sa rozhodol použiť haversine formulu [Kapitola 5.4].

Ďalším, celkom jednoduchým krokom bolo umožniť užívateľovi aby si mohol vybrať časové rozpätie o ktoré má záujem, k tomu som využil Date Range Picker [Kapitola 5.5].

Následne bolo potrebné vypracovať samotný graf v ktorom sa budú zobrazovať údaje o predpovedi počasia, k tomuto som využil voľne dostupný Chart.js do ktorého sa dajú cez PHP a JavaScript preniesť údaje z databáze [Kapitola 5.6]. Toto mi umožnilo zobrazovať predpoveď počasia presne pre tú jednu lokalitu a časové okno o ktoré má užívateľ záujem, avšak stále ma zaujímal ešte jeden spôsob zobrazovania počasia za pomoci mapy. U tohto je ale problém že na mape nieje vhodné zobrazovať viacero dní/hodín počasia naraz, iba pre jednu konkrétnu hodinu či deň. Preto som sa rozhodol k mojej mape implementovať bežec ktorým sa bude dať posúvať v čase. Týmto bude možné takto zobraziť všetky lokality a časy na jednej a tej istej mape [Kapitola 5.7].

Na záver by bolo dobré implementovať administráciu pre túto webovú aplikáciu. Vzhľadom k tomu že dáta o predpovedi počasia sú pridávané a aktualizované na základe údajov z databáze, stačí cez administráciu robiť mierne úpravy tabuľky lokalít v databáze, prípadne mazať staré záznamy.

## 5.1. Voľba lokalít

Servery pre predpoveď počasia poskytujú dáta po celom svete, a napríklad Open Weather Map poskytuje dáta pre 1496 lokácií v Českej republike [11]. Avšak kvôli limitom ktoré obmedzujú bezplatný počet prístupov k týmto lokalitám nieje možné sťahovať takýto počet lokácií a aktualizovať ich v nejakom rozumnom intervale. Práve preto by bolo vhodné využiť iba niekoľko z týchto lokácií na začiatok, a nechať priestor na možné rozšírenia, ktoré už budú niečo stáť, na neskôr. Takže by som zvolil iba krajské mestá Českej republiky. Týmto by sme mali získať celkom pekné pokrytie a všetky ostatné mestá ktoré nebudú v databáze budú presmerované na najbližšie krajské mesto. K týmto mestám bude potrebné vyhľadať jejich zemepisné súradnice a identifikačné čísla týchto lokalít od niektorých serverov ktoré neposkytujú vyhľadávanie na základe zemepisných súradníc. V tabuľke 5.1 sú všetky tieto údaje zobrazené.

Tabuľka 5.1: Vybrané lokality a jejich pomocné údaje

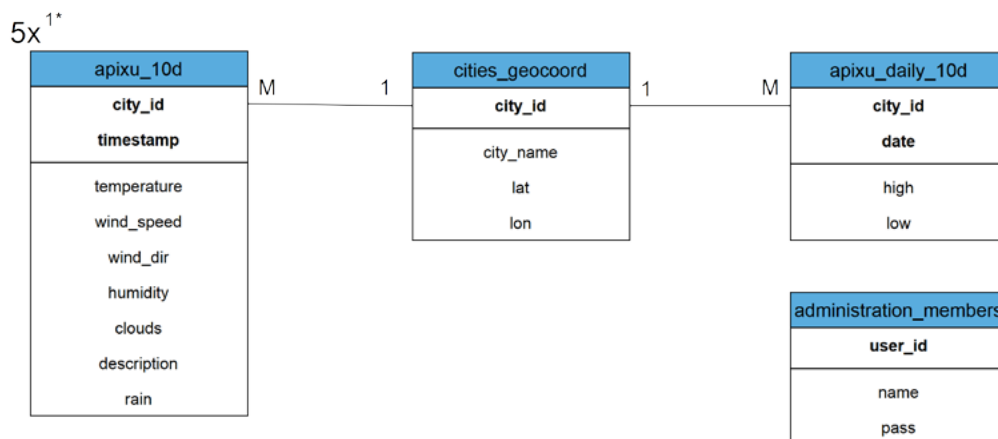
Mesto	Zemepisná šírka	Zemepisná dĺžka	Open Weather Map ID	AccuWeather ID
Praha	50.0833	14.4667	3067695	125594
Ostrava	49.8346	18.2820	3068799	124690
Brno	49.1952	16.6080	3078610	123291
Ústí nad Labem	50.6607	14.0323	3063548	126812
Olomouc	49.5955	17.2518	3069011	124966
České Budějovice	48.9745	14.4743	3077916	123206
Zlín	49.2167	17.6667	3061370	126755
Plzeň	49.7475	13.3776	3068160	125505
Hradec Králové	50.2092	15.8328	3074967	123377
Jihlava	49.3961	15.5912	3074199	126629
Pardubice	50.0407	15.7766	3068582	125084
Liberec	50.7671	15.0562	3071961	124033
Karlovy Vary	50.2327	12.8712	3073803	123294

## 5.2. Návrh databáze

Návrh databáze je celkom jednoduchý. Ako prvé je potrebné vytvoriť tabuľku kde budú uložené všetky lokality. K nim sú potrebné zemepisné súradnice a identifikačné tabuľke od Open Weather Map a AccuWeather. Dáta o predpovedi počasia plánujem stahovať z piatich rôznych zdrojov. Preto som sa rozhodol vytvoriť päť tabuliek, tak aby každý zdroj mal jednu, a pre jednoduchšie kombinovanie dát som tieto tabuľky urobil identické. Tieto tabuľky budú obsahovať identifikačné číslo lokality ku ktorej sa vzťahujú, a základné údaje o počasí ako teplota, vlhkosť, oblačnosť atď.

Ako primárne kľúče pre tabuľky s predpovedou počasia som zvolil dvojice identifikačné číslo mesta a časovú stopu daného záznamu. Táto dvojica je pre každý záznam unikátna a identifikačné číslo mesta sa dá použiť pre spojenie s tabuľkou lokalít pre vyhľadanie názvu mesta a zemepisných súradníc ku ktorým sa daný záznam vzťahuje.

Ďalej bude potrebná tabuľka administrátorov, ktorá bude obsahovať prihlasovacie údaje do webovej administrácie. Táto tabuľka sa nebude nijako viazať s ostatnými tabuľkami.



Obrázok 5.2: ER diagram databáze  
<sup>1</sup>Päť identických tabuliek tohto typu

### 5.3. Stahovanie predpovede počasia

Dáta o predpovedi počasia plánujem stahovať z piatich vybraných zdrojov. Pri sťahovaní týchto dát bude najdôležitejšie myslieť na neskoršie pridávanie ďalších lokalít. Preto som sa rozhodol vytvoriť pre každý server jeden PHP skript. Tieto skripty si ako prvé stiahnu celú tabuľku *cities\_geocoord* z databáze cez MySQLi, a na základe tejto tabuľky využije potrebné identifikačné číslo alebo zemepisné súradnice pre správne doplnenie URL pre volanie API. Z tohto volania web API získam textový retazec s JSON štruktúrou, ktorý premením na pole pomocou funkcie `json_decode(string, true)`. V tomto poli je už jednoduché jednotlivé prvky správne adresovať.

Dáta z niektorých zdrojov je ešte potrebné trochu upraviť, tabuľky do ktorých sú ukladané sú identické pre každý server, avšak v stahovanej JSON štruktúre sa dosť líšia, niektoré jednotky je potreba pozmeniť, ale po zmenení dostaneme všetko metrické jednotky rovnakého typu do databáze. Tým mám záznamy o predpovedi počasia pripravené pre každé mesto z tabuľky *cities\_geocoord* a už ich iba postupne, mesto po meste, vložím do databáze.

Mnoho webov taktiež ponúka históriu počasia, avšak vždy je spoplatnená. Preto som si taktiež chcel začať vytvárať vlastnú históriu počasia pre moju aplikáciu, a vďaka tomu že v tabuľkách so záznamami o predpovedi počasia som si vybral identifikačné číslo mesta a časovú stopu ako primárny kľúč tabuľky, toto môžem ľahko docieľiť nasledujúcim SQL príkazom

```
INSERT INTO ds_7d (...) VALUES ... ON DUPLICATE KEY UPDATE ...
```

Týmto získam to, že sa mi predpoveď počasia aktualizuje, a staré záznamy o aktuálnom počasi v tabuľke aj naďalej zostanú, a tým sa mi tvorí história počasia pre každé krajské mesto Českej republiky z piatich rôznych serverov, a taktiež by sa tvorila aj pre ďalšie, neskôr pridané mestá.

Z hľadiska testovania boli tieto skripty uložené na mojom servere a boli pravidelne spúšťané crontabom už vyše 60 dní. Do databáze som pravidelne nahliadal a všetky záznamy a dáta



boli správne ukladané a aktualizované. V crontab tabuľkách boli tieto PHP skripty spúšťané každú hodinu týmto kódom, okrem skriptu pre zdroj AccuWeather, ktorý sa spustí iba každých 8 hodín kvôli limitom pre API.

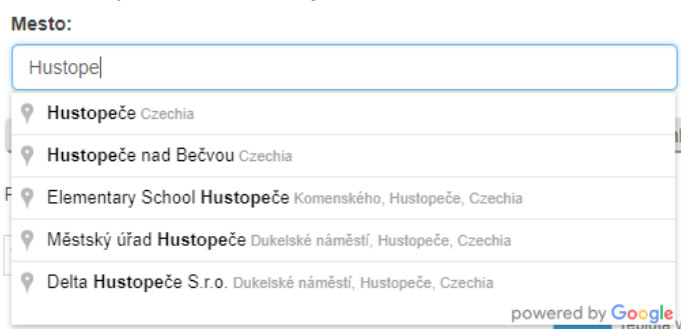
```
05 * * * * wget -q http://renezis.eu/forecast/dark_db_update.php
06 * * * * wget -q http://renezis.eu/forecast/owm_db_update.php
07 0,8,16 * * * wget -q http://renezis.eu/forecast/accu_db_update.php
08 * * * * wget -q http://renezis.eu/forecast/weatherbit_db_update.php
09 * * * * wget -q http://renezis.eu/forecast/apixu_db_update.php
25 * * * * wget -q http://renezis.eu/forecast/apixu_daily_db_update.php
```

Výpis 5.1: Obsah CRON TABLE pre spúšťanie zberačov dát o predpovedi počasia

## 5.4. Zemepisné súradnice a výpočet vzdialenosti medzi nimi

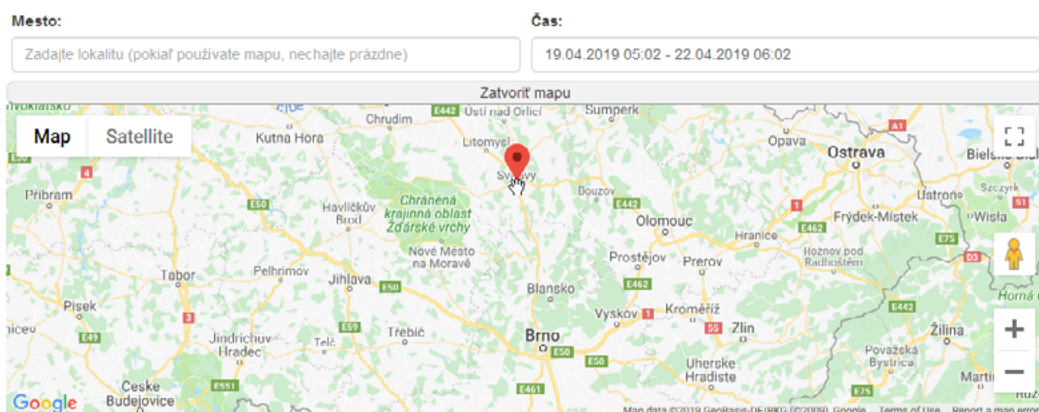
Zo záznamov o predpovedi počasia som získal zemepisné súradnice ku ktorým sa vzťahujú. Z tých som si vytvoril zoznam lokalít pre ktoré mám dostupné dáta. Problém vzniká pokiaľ ale návštevník má záujem o lokalitu pre ktorú nemám žiadne záznamy v databázy. Preto som vypracoval výber lokácie dvomi rôznymi spôsobmi, oba za pomoci google API.

Ako prvým si užívateľ môže vybrať lokalitu o ktorú má záujem pomocou textového poľa ktoré používa funkciu autocomplete od google API [8]. Týmto spôsobom môže zadať čiastočný názov lokality, a budú mu ponúknuté nejaké možnosti na výber. Pri odoslaní formulára bude tento užívateľom zadaný textový retazec poslaný na google geocoding z ktorého dostanem zemepisné súradnice lokality o ktorú má záujem.



Obrázok 5.3: Ukážka funkcie autocomplete od Google

Druhý spôsob som vypracoval priamo cez mapu od google. Do týchto máp je možné vkladať značky (ang. marker), a taktiež ich za pomoci JavaScriptu priebežne meniť. Takže som vypracoval jednu mapu ktorá reaguje na kliknutie užívateľa, a na kliknuté súradnice uloží značku. Taktiež si tieto súradnice uloží do skrytých vstupných polí, a tým sa mi tieto súradnice prenesú pri odoslaní formulára užívateľom.



Obrázok 5.4: Výber lokality za pomoci mapy a značky od Google

Čiže sa mi podarilo vypracovať dva spôsoby ktorými si užívateľ môže vybrať lokalitu, už je iba potrebné, na základe dvoch dvojíc zemepisných súradníc, nájsť najbližšie mesto k tejto lokalite pre ktoré mám dostupné dáta.

Pre zemepisnú šírku (anglicky latitude) platí že každý stupeň zemepisnej šírky na povrchu zeme je vzdialený približne 111 kilometrov od susedných stupnov. Kvôli elipsoidnému tvaru zeme je táto vzdialenosť najnižšia u rovníku, kde to je 110,567 kilometrov a u pólův je najvyššia vzdialenosť 111,699 kilometrov [12].

Problém je ale u zemepisnej dĺžky (anglicky longitude) ktorá náš výpočet o dosť komplikuje. Pretože jeden stupeň zemepisnej dĺžky u rovníka by sa na povrchu zeme rovnal vzdialenosti 111.321 kilometrov. Avšak čím bližšie sa dostávame k pólom zemegule tak sa táto vzdialenosť znižuje až na nulu u pólůch. Čiže na základe týchto údajov by sme mohli zostrojiť vzorec:

$$v = \cos(lat) \cdot 111.321$$

Kde  $v$  je vzdialenosť jedného stupňa zemepisnej dĺžky v kilometroch a  $lat$  je zemepisná šírka v stupňoch.

Za pomoci týchto údajov by sme dokázali vytvoriť vzorec pre približný výpočet vzdialenosti dvoch bodov, ktorý by ale ignoroval akékoľvek zaoblenie zeme. Čiže takýto vzorec by bol použiteľný iba do vzdialenosti približne 400 až 500 kilometrov a pre väčšie vzdialenosti by sme už dostávali úplne nepresné výsledky.

Našťastie, všetky výpočty pre webovú aplikáciu bude vykonávať počítač a preto som sa rozhodol využiť **sférickú trigonometrickú haversine formulu** [13].

$$a = \sin^2(\Delta lat/2) + \cos(lat_1) \cdot \cos(lat_2) \cdot \sin^2(\Delta lon/2)$$

$$c = 2 \cdot \text{atan2}(\text{sqrt}(a), \text{sqrt}(1-a))$$

$$d = R \cdot c$$

$R$  je polomer zeme, ktorý je 6378.137 kilometrov [27], avšak pre výsledkoch v metroch ho zadáme ako 6 378 137 metrov. Implementácia tejto formuly je na základe vzorca už celkom jednoduchá, pretože v PHP už máme k dispozícii funkcie ako sinus, kosínus, arkus tangens..

Jediná nevýhoda tejto formuly je to, že ignoruje elipsoidný tvar zeme a miesto toho používa pre výpočet perfektnú guľu bez nerovností, toto môže spôsobiť výchyľky až 0,3% ktoré sú pre naše účely ale zanedbateľné [14].

Taktiež som vypracoval úvodné vyhľadávanie na základe polohy užívateľa, čiže aj pri prvom načítaní stránky mu je zobrazená predpoveď počasia podľa jeho lokality. Tohto som dosiahol za pomoci ip-api.com, čím viem zistiť jeho približné súradnice podľa jeho IP adresy.

Z hľadiska testovania som na webovej stránke vytvoril podstránku “Štatistiky”, kde výsledná implementácia tejto funkcie vypočíta vzdialenosť všetkých lokalít v databáze od približnej lokácie návštevníka webovej stránky. Tuto sú moje výsledky z lokality Nové Zámky, Slovensko.

Tabuľka 5.5: Vzdialenosť vybraných lokalít od mesta Nové Zámky, vypočítané za pomoci haversine formuly

Vaša približná lokácia:	Nové Zámky
Vaša zemepisná šírka:	47.9899
Vaša zemepisná dĺžka:	18.1666
Vzdialenosť od mesta Praha:	402.276 km
Vzdialenosť od mesta Ostrava:	205.376 km
Vzdialenosť od mesta Brno:	192.518 km
Vzdialenosť od mesta Ústí nad Labem:	471.819 km
Vzdialenosť od mesta Olomouc:	196.1 km
Vzdialenosť od mesta České Budějovice:	345.207 km
Vzdialenosť od mesta Zlín:	143.435 km
Vzdialenosť od mesta Plzeň:	467.448 km
Vzdialenosť od mesta Hradec Králové:	322.072 km
Vzdialenosť od mesta Jihlava:	276.768 km
Vzdialenosť od mesta Pardubice:	311.32 km
Vzdialenosť od mesta Liberec:	414.069 km
Vzdialenosť od mesta Karlovy Vary:	531.665 km

## 5.5. Voľba času

Pri výbere časového rozpätia je najväčší problém naviesť užívateľa aby ho napísal v správnom formáte. Vzhľadom nato že moja aplikáciu bude mať aj históriu počasia je potrebné aby užívateľ napísal čas a dátum. Toto som sa rozhodol zjednodušiť využitím nástroja Date Range Picker. Pri otvorení okienka s časom sa užívateľovi zobrazí kalendár v ktorom si môže požadovaný dátum a čas naklikat', čo výrazne zjednodušuje zadávanie vstupu pre aplikáciu, ale hlavne pomáha užívateľovi aby zadal tento vstup v správnom formáte.

## 5.6. Mini-aplikácia na zobrazenie predpovede počasia na 10 dní na základe lokácie návštevníka

Túto mini-aplikáciu som chcel vytvoriť hlavne ako medzikrok pred implementovaním samotného grafu. Dost' mi pomohla na otestovanie mojej funkcie výpočtu vzdialenosti a pre osvojenie

základov zobrazovania údajov z databáze na webovej stránke.

Táto mini-aplikácia si za pomoci PHP zistí ip adresu návštevníka a to z premennej `$_SERVER['REMOTE_ADDR']`. Následne je táto ip adresa poslaná ako súčasť URL pri prístupe na web API od ip-api.com. Táto služba nám poskytne nejaké základné informácie o tejto ip adrese, ale hlavne týmto získame približné zemepisné súradnice návštevníka webovej stránky. Za pomoci už vytvorenej funkcie pre zistenie vzdialenosti na základe dvoch zemepisných súradníc už vieme zistiť vzdialenosť návštevníka webovej stránky od všetkých lokalít pre ktoré máme dostupné dáta. Potom už iba potrebujeme vybrať 10 záznamov pre túto lokalitu. Zobrazenie dátumov, najvyššej a najnižšej teploty dňa je taktiež vypracované v PHP, avšak u dátumov je potrebné zmeniť lokalizáciu aby názvy mesiacov boli vypísané v správnom jazyku. Kvôli tomuto bolo potrebné vygenerovať slovenskú lokalizáciu na servery, takže pri prenesení tejto webovej aplikácie na iný server ju bude potrebné znovu vygenerovať.

## 5.7. Hlavný graf predpovede počasia

Pri implementácii hlavného grafu som použil JavaScriptový nástroj Chart.js. Za pomoci HTML tagu `<canvas>` bolo možné vyhradiť priestor pre graf. Zobrazenie dát v grafe bolo bezchybné, problémom ale bolo vybrať správne dáta z databáze cez PHP, preniesť ich do JavaScriptu a vložiť ich do grafu.

Pre výber správnych dát je potrebné aby užívateľ webovej aplikácie vyplnil dva údaje a to presnú lokalitu a časové rozpätie pre ktoré si chce dané dáta v grafe zobrazit'. Spracovanie týchto vstupných údajov som implementoval v jazyku PHP. Ako prvým problémom bolo potrebné riešiť prvotné načítanie stránky, kedy užívateľ ešte nezadal žiadne vstupné údaje. Toto som sa rozhodol riešiť tým, že pre lokalitu vyberiem tú, ku ktorej je najbližšie na základe jeho IP adresy a ohľadom času mu zobrazím v grafe podrobnú predpoveď počasia na 3 dni. Informácie ohľadom polohy budú odoslané na geocoding web API od google. V prípade že sa užívateľ rozhodne funkciu autocomplete nevyužiť a zadá lokalitu v nejakom inom formáte, tak vo väčšine prípadoch to vyzerá tak, že google geocoding je stále schopný to spracovať a odoslať chcené výsledky. Z funkcie geocoding týmto získam zemepisné súradnice lokality o ktorú má užívateľ záujem a na základe týchto súradníc je už jednoduché vypočítať ktorá lokalita s dátami je k nej najbližšie. Týmto už máme skoro všetko hotové pre výber vhodných dát z databáze, stačí už iba trochu upraviť formát časového rozpätia.

Taktiež, užívateľ má na výber z piatich zdrojov a môže si vybrať akúkoľvek kombináciu z nich. Pri výbere viacerých zdrojov som chcel v grafe zobrazit' priemerované hodnoty týchto zdrojov. Toto som sa rozhodol vypracovať priamo do SQL selectu v ktorom vyberiem správnu lokalitu, orežiem dáta o zvolený čas a spriemerujem hodnoty z viacerých identických tabuliek. Takže v prípade výberu zdrojov Dark Sky, Open Weather Map a Weatherbit by tento select vyzeral nasledovne:

```

SELECT
    timestamp,
    ROUND(AVG(temperature), 2) AS temperature_avg,
    ROUND(AVG(wind_speed), 2) AS wind_speed_avg,
    ROUND(AVG(humidity), 2) AS humidity_avg,
    ROUND(AVG(clouds), 2) AS clouds_avg,
    ROUND(AVG(rain), 2) AS rain_avg
FROM
    (
        SELECT * FROM ds_7d
        UNION ALL
        SELECT * FROM owm_5d
        UNION ALL
        SELECT * FROM wb_2d
    ) AS A
LEFT JOIN cities_geocoord ON A.city_id = cities_geocoord.
city_id
WHERE
    city_name = 'Zlín'
    AND timestamp >= '2019-03-20 02:24:00'
    AND timestamp <= '2019-03-23 03:24:00'
GROUP BY
    A.city_id,
    timestamp

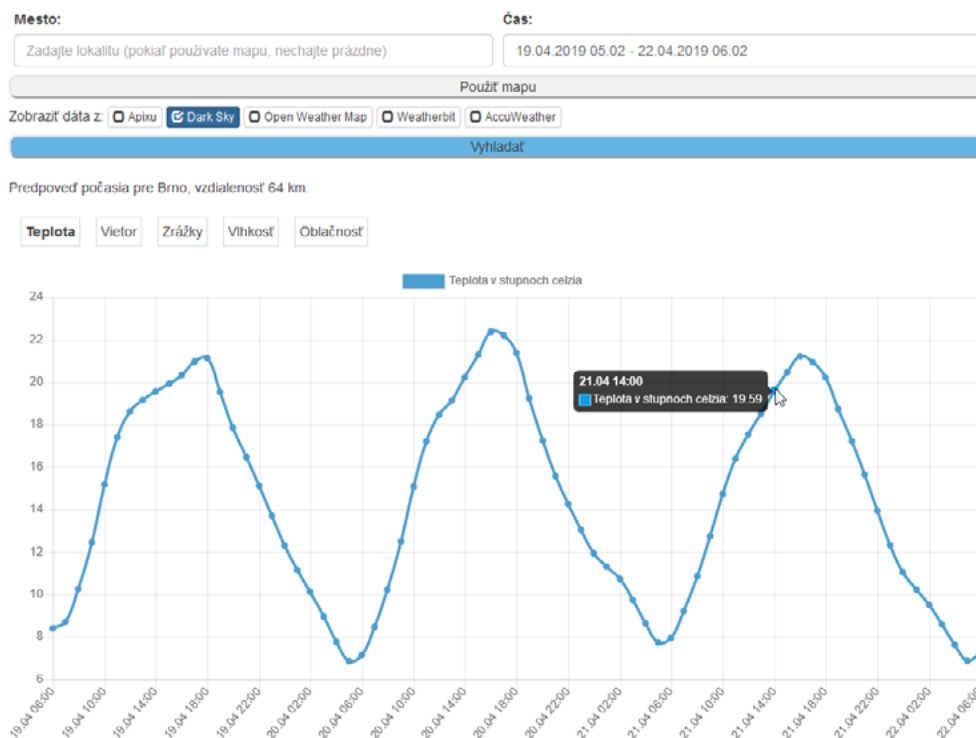
```

Výpis 5.2: Výber dát pre nastaviteľný graf z viacerých identických tabuliek

Tieto dáta už je iba potrebné preniesť do JavaScriptu, pretože s nimi bude potrebné pracovať aj po načítaní stránky. Týmto som v JavaScripte dostal 6 polí a to:

- forecast\_timestamp . . . . časová známka
- forecast\_temperature . . . teplota
- forecast\_wind\_speed . . . rýchlosť vetra
- forecast\_rain . . . . . zrážky
- forecast\_humidity . . . . . vlhkosť
- forecast\_clouds . . . . . oblačnosť

Pole časových známok je pevne vložené do x-ovej osi grafu a menia sa iba hodnoty na y-ovej ose. Tie sú napojené na tlačítka nad grafom a vždy je možné zobrazit' iba jednu z nich.



Obrázok 5.6: Nastaviteľný graf predpovede počasia

## 5.8. Mapa predpovede počasia

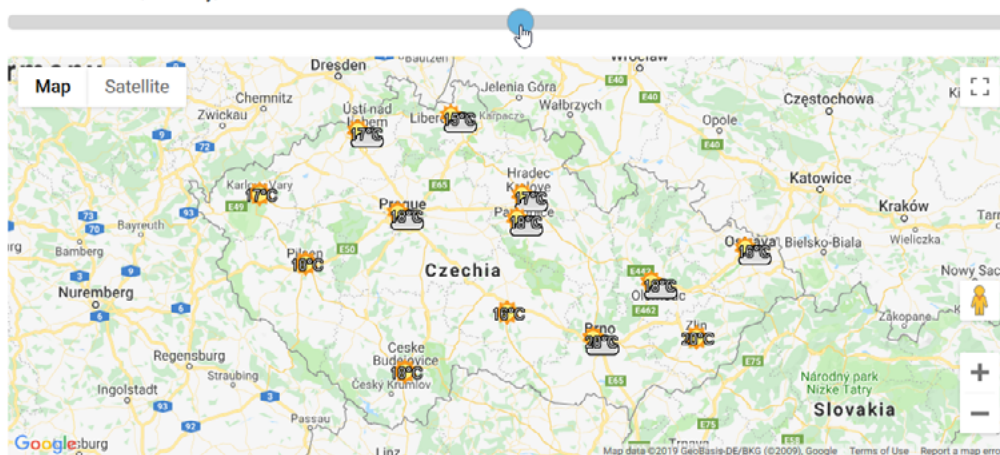
Ku grafu som taktiež chcel vypracovať mapu predpovede počasia. Na tejto mape by boli zobrazené všetky lokality z databáze, a ktomu by som vypracoval bežec ktorým by sa dalo posúvať v čase.

K vypracovaniu samotnej mapy som použil mapy od Google. Na týchto mapách sa dá vytvárať značky ktorým sa dá meniť ikona a dá sa im aj pridať nejaký text [31]. Pri vytvorení značky je potrebné zadať jej neaké súradnice, tie znovu čerpám z databáze, takže taktiež pri pridaní nového mesta by bolo pridané aj na túto mapu.

Úvodným plánom teda bolo to, že pri zmene polohy bežca by sa načítala nová ikona a nový text pre všetky značky. K textu a ikonám som sa rozhodol použiť dáta zo zdroja Dark Sky, pretože sa mi podarilo nájsť presne tie ikony ktoré patria k textovému údaju opisujúci počasie. Hlavným problémom bolo ale že Google maps nie sú prispôbené na viacej riadkové opisy značiek. Taktiež sa mi nepodarilo nájsť žiadny spôsob ako textu pridať okraje, takže vo výsledku ikony fungovali dobre, ale text bol nepriehľadný, a na jednom riadku. Toto som sa rozhodol riešiť cez PHP GD knižnicu [32] a text priamo pridať do obrázka. V prvotnej implementácii boli problémy s výkonom, pri každom pohybe bežca či načítaní stránky bolo potrebné pre každé mesto vybrať vhodnú ikonu a pridať jej text s okrajmi. Toto som optimalizoval tým že pri prvom vygenerovaní ikony s určitou teplotou sa ikona poslala užívateľovi, ale bola aj uložená do súboru cache, a pri ďalšom načítaní jednoducho odoslaná cez `file_get_contents()` čo značne zrychlilo načítavanie ikon pre mapu.

## Mapa predpovede počasia na 7 dní

Čas: 22.04.2019, Monday, 19:00:00



Obrázok 5.7: Mapa predpovede počasia so sliderom ktorým sa dá posúvať v čase

## 5.9. Webová administrácia

K tejto webovej aplikácii som taktiež vypracoval aj administráciu, cez ktorú je možné po prihlásení pridávať a mazať lokality v tabuľke *cities\_geocoord*.

Prihlasovanie do webovej administrácie som vypracoval formou formulára, do ktorého si administrátor napíše meno a heslo. Tento formulár je odosielaný spôsobom POST. Odoslané prihlasovacie údaje sú porovnávané s údajmi v databáze v tabuľke *administration\_members*. Pokiaľ sa zhodujú s jedným zo záznamov tejto tabuľky budú užívateľovi nastavené premenné do  $\$_SESSION['názov']$ , a po úspešnom prihlásení sa mu zobrazí formulár s pridaním novej lokality, zmazaním aktuálnych a tlačítko odhlásiť sa.

Pridávanie lokalít je taktiež spracované formulárom. U tohto formuláru je potrebné vyplniť všetky údaje tabuľky *cities\_geocoord* ako názov lokality, zemepisné súradnice a identifikačné čísla Open Weather Map a AccuWeather. Údaj *city\_id* bude automaticky pridelený v databáze za pomoci funkcie *auto increment*. Novo pridaná lokalita avšak ešte nebude mať žiadne záznamy o predpovedi počasia. K tomuto je buď to treba manuálne rozbehnúť PHP skripty pre zber dát, alebo počkať kým budú automaticky spustené *crontabom*.

Pre mazanie lokalít som na stránke administrácie urobil kompletný zoznam momentálne aktívnych lokalít, a ku každému z nich som pridal odkaz na zmazanie ktorý smeruje na aktuálnu stránku s argumentom *del\_city="city\_id"*. Týmto dosiahnem toho že mi bude cez GET odoslané *city\_id* na zmazanie, a toto mesto zmažem z tabuľky *cities\_geocoord*. Problémom sú tam ale záznamy o predpovedi počasia, a záznamy o histórii počasia. Momentálne tieto záznamy taktiež mažem pri zmazaní mesta, ale bolo by dobré mať nejaký záložný server alebo priestor kam by sa mohli pred zmazaním zálohovať.

### Pridať novú lokalitu


### Zmazať lokalitu

ID	Názov mesta	Šírka	Dĺžka	OWM ID	Accu ID	Zmazať
1	Praha	50.0833	14.4667	3067695	125594	<a href="#">Zmazať</a>
2	Ostrava	49.8346	18.282	3068799	124690	<a href="#">Zmazať</a>
3	Drno	49.1952	16.608	3070610	123291	<a href="#">Zmazať</a>
4	Ústí nad Labem	50.6607	14.0323	3063548	126812	<a href="#">Zmazať</a>
5	Olomouc	49.5955	17.2510	3069011	124966	<a href="#">Zmazať</a>
6	České Budějovice	48.9746	14.4743	3077916	123206	<a href="#">Zmazať</a>
7	Zlín	49.2167	17.6667	3061370	126755	<a href="#">Zmazať</a>
8	Plzeň	49.7475	13.3776	3068160	125505	<a href="#">Zmazať</a>
9	Hradec Králové	50.2092	15.8328	3074967	123377	<a href="#">Zmazať</a>
10	Jihlava	49.3961	15.5912	3074199	126629	<a href="#">Zmazať</a>
11	Pardubice	50.0407	15.7766	3068582	125084	<a href="#">Zmazať</a>
12	Liberec	50.7671	15.0562	3071961	124033	<a href="#">Zmazať</a>
13	Karlovy Vary	50.2327	12.8712	3073803	123294	<a href="#">Zmazať</a>

Obrázok 5.8: Náhľad do tela webovej administrácie

## 5.10. Zhrnutie a obsah web stránky

Takže, v konečnom výsledku dostávame webovú aplikáciu v ktorej si dokážeme cez administráciu pridať a odobrať lokality. Tie su uložené v databázy a na základe týchto lokalít bude každú hodinu stiahnutá či aktualizovaná predpoveď počasia pre tieto lokality, a tieto predpovede je možné zobrazit' tromi spôsobmi. Prvým je miniaplikácia ktorá obsahuje iba dennú predpoveď na 10 dní podľa lokácie návštevníka. Druhou je nastaviteľný graf v ktorom si je možné vybrať lokalitu a časové okno pre ktoré budú dáta zobrazené. Treťou je mapa s bežcom ktorým sa dá posúvať v čase, sú tam teda zobrazené všetky lokality a časy, avšak pre každú lokalitu je tam zobrazená iba ikona a teplota.

Výsledná webová stránka bola vytvorená v HTML. Taktiež k nej bolo vytvorené príslušné CSS. Stránka obsahuje domovskú stránku a 3 podstránky:

- domov s grafom,
- mapa s bežcom,
- štatistiky s výpočtom vzdialeností,
- administrácia pre pridávanie a mazanie lokalít.



## Kapitola 6

# Možné rozšírenia

Webová aplikácia je momentálne vypracovaná k tomu aby si získavala dáta o 13 lokalitách z dvoch rôznych zdrojov. Jediný problém pri rozširovaní o nové lokality či mestá je získavanie dát do databáze. Pokiaľ sú tabuľky databáze vhodne naplnené údajmi o X lokalitách, webová stránka ich už dokáže zobrazit' a taktiež vie vyhľadať najbližšiu lokalitu s dátami. K tomuto som taktiež vypracoval webovú administráciu cez ktorú je možné jednoducho pridávať ďalšie mestá. Avšak problémom sú limity k API prístupom.

U AccuWeather už teraz potrebujem znížiť počet aktualizácií na tri krát za deň. Limit povoľuje iba 50 API volaní za deň, čo je velice málo, keďže každé mesto pridáva jedno volanie API pri každej aktualizácii. Čiže už pri trinástich mestách by som potreboval volať API 312 krát za deň pri hodinových aktualizáciach, ostatné servery majú voľnejšie limity, ale u tohto by sa to dalo riešiť buď prechodom na platenú verziu, alebo upustením tohto zdroja.

Z pohľadu user experience je webová stránka vypracovaná aby bola priehľadná a dalo sa s ňou pohodlne pracovať. Tomuto hlavne pomáha funkcia autocomplete a funkcia date range picker pri zadávaní vstupných údajov. Avšak jednotlivé elementy stránky by bolo možné ešte doladiť.

Taktiež by bolo dobré vytvoriť nejaký nastaviteľný nástroj na mazanie histórie počasia, prípadne na jej presun alebo zálohovanie na inom servery. Pretože momentálne si vytváram históriu počasia u každého z piatich zdrojov, tým dostávam pre každé mesto 104 záznamov denne. Čiže podľa situácie bude potrebné tieto záznamy neskôr mazať alebo ich presúvať na nejaký vysoko kapacitný server.

Ďalší vývoj tejto aplikácie by som pokračoval prácou na vlastnom grafe, u tohto je problém že chart.js je vypracovaný tak aby zobrazoval nejaké všeobecné dáta, a nieje špecificky zameraný na zobrazovanie predpovede počasia, takže zostáva celkom veľký priestor pre vylepšenie jak vzhľadu, tak aj funkčnosti.

Prípadne k webovej aplikácii by bolo možné pridať mobilnú aplikáciu. Bolo by vhodné ju vypracovať tak, aby si dokázala stiahnuť dáta z webovej stránky, najlepšie cez vlastné web API. K tomu by bolo možné pridať možnosť ukladania údajov do pamäte pre neskoršie offline zobrazenie. K tomuto by som využil detailnú predpoveď počasia na 10 dní od Apixu. Avšak predpoveď počasia nieje až tak presná a dosť často sa mení, preto by bolo taktiež dobré zobrazovať údaj poslednej aktualizácie.

## Kapitola 7

# Záver

Cieľom tejto práce bolo navrhnúť a implementovať webovú aplikáciu pre predpoveď počasia. K tomu mala táto aplikácia byť schopná stahovať predpoveď počasia aspoň z dvoch zdrojov, niekde si tieto dáta ukladať, a dokázať ich zobrazit' formou grafu v ktorom si je možné nastaviť lokalitu a čas.

Ako prvé som si vyhlíadol dostupné zdroje predpovede počasia. Z týchto som si zvolil štyri voľne dostupné, a u piateho som si dokonca zaplatil prístup k hodinovým dátam a volnejším limitom. Ďalej som vypracoval tabuľku s trinástimi lokalitami pre ktoré si budem stahovať dáta z týchto zdrojov, avšak samotné zberače dát som vypracoval tak nech fungujú na základe údajov tejto tabuľky, nie na pevných trinástich lokalitách.

Následne som vypracoval tri rôzne spôsoby zobrazovania týchto údajov o predpovedi počasia. Ako prvý bol skoršie testovací prototyp ktorý zobrazí dennú predpoveď na 10 dní podľa polohy návštevníka tejto webovej aplikácie. Čiže pri jeho tvorbe som implementoval výpočet vzdialenosti od dostupných lokalít. Ako druhý som vypracoval nastaviteľný graf ktorý je jadrom mojej práce. U grafu je možné si vybrať lokalitu dvoma spôsobmi, a to buď v textovom poli napísať názov lokality (s doplnením od google maps), alebo si priamo na mape zakliknúť nejakú polohu. K tomu je taktiež možné vybrať si časové okno od-do za pomoci nástroja Date Range Picker. V grafe sa ale dá zobrazit' iba jediná lokalita, a k tomu ma ešte zaujímalo zobrazovanie cez mapu, kde by bolo možné zobrazit' všetky lokality. Takže ako prvé rozšírenie mojej aplikácie som vypracoval mapu s bežcom ktorým sa dá posúvať v čase. Táto mapa zobrazuje predpoveď na 7 dní, a ohľadom údajov zobrazuje ikonu počasia a presnú teplotu.

Ohľadom zdrojov ktoré sú využívané, miniaplikácia využíva dáta z Apixu, pretože tam mám zaplatený balíček ktorý obsahuje predpoveď na 10 dní. V grafe je možné si vybrať jeden či viacero z piatich dostupných zdrojov. Pri výbere viacerých zdrojov sú dáta rovnakej časovej stopy spriemerované. A pre mapu sú kvôli ikonám použité dáta z Dark Sky.

Zo všetkých piatich zdrojov si taktiež vytváram históriu počasia ktorú je možné zobrazit' v hlavnom grafe, a taktiež aj históriu je možné kombinovať z viacerých zdrojov.

Na záver som túto prácu rozšíril o administráciu ktorou je možné po prihlásení pridávať či odoberať lokality. Treba ale dávať pozor na to že pri mazaní lokalít sa maže aj celá história záznamov tej lokality.

# Literatúra

- [1] Web API. 2018, [Online; navštívené 29.04.2018]  
URL [https://en.wikipedia.org/wiki/Web\\_API](https://en.wikipedia.org/wiki/Web_API)
- [2] What is Web API. 2018, [Online; navštívené 01.05.2018]  
URL <http://www.tutorialsteacher.com/webapi/what-is-web-api>
- [3] Yahoo Query Language. 2018, [Online; navštívené 14.04.2018]  
URL <https://developer.yahoo.com/yql/>
- [4] Open Weather Map. 2018, [Online; navštívené 13.04.2018]  
URL <https://openweathermap.org/about>
- [5] World Weather Online. 2018, [Online; navštívené 28.04.2018]  
URL <https://developer.worldweatheronline.com/>
- [6] World Weather Online, premium API pricing. 2018, [Online; navštívené 28.04.2018]  
URL <https://developer.worldweatheronline.com/api/pricing.aspx>
- [7] IP Geolocation API. 2015, [Online; navštívené 05.05.2018]  
URL <http://ip-api.com/docs/>
- [8] List of Google products. 2019, [Online; navštívené 02.05.2019]  
URL <https://developers.google.com/products/>
- [9] Google Maps. 2018, [Online; navštívené 01.05.2018]  
URL <https://www.google.com/maps>
- [10] Autocomplete for Addresses and Search Terms. 2018,  
[Online; navštívené 16.04.2018]  
URL <https://developers.google.com/maps/documentation/javascript/places-autocomplete>
- [11] Open Weather Map: Zoznam miest a jejich ID. 2018, [Online; navštívené 15.04.2018]  
URL <http://bulk.openweathermap.org/sample/city.list.json.gz>
- [12] Matt Rosenberg: *What Is the Distance Between a Degree of Latitude and Longitude?*. 2018, [Online; navštívené 18.04.2018]  
URL <https://www.thoughtco.com/degree-of-latitude-and-longitude-distance-4070616>
- [13] Moveable Type Scripts: *Calculate distance, bearing and more between Latitude/Longitude points*. 2018, [Online; navštívené 20.04.2018]  
URL <https://www.movable-type.co.uk/scripts/latlong.html>
- [14] Error when approximating Earth as sphere. 2012, [Online; navštívené 20.04.2018]  
URL <https://gis.stackexchange.com/questions/25494/how-accurate-is-approximating-the-earth-as-a-sphere#25580>

- [15] Jason Summerfield: *Mobile Website vs. Apps*. 2017, [Online; navštívené 04.05.2018]  
URL <https://www.hswsolutions.com/services/mobile-web-development/mobile-website-vs-apps/>
- [16] Raluca Budiu: *Mobile: Native Apps, Web Apps, and Hybrid Apps*. 2013, [Online; navštívené 04.05.2018]  
URL <https://www.nngroup.com/articles/mobile-native-apps/>
- [17] Jeffrey Elkner: The Element Tree. 2017, [Online; navštívené 05.05.2018]  
URL <http://www.openbookproject.net/tutorials/getdown/css/lesson4.html>
- [18] Žára, O.: *JavaScript - Programátorské techniky a webové technologie*, Computer Press, 2015. ISBN: 978-80-251-4573-9
- [19] Deborah Lee Soltesz: *What Does JavaScript Do?*. [Online; navštívené 06.05.2018]  
URL <https://www.techwalla.com/articles/what-does-javascript-do>
- [20] MySQL: *Documentation*. 2018, [Online; navštívené 26.04.2018]  
URL <https://dev.mysql.com/doc/>
- [21] Welling, L., Thomsonová, L.: *PHP a MySQL: rozvoj webových aplikací*. Vyd. 1. Praha: SoftPress, 2003, 910 s. ISBN 80-86497-60-7.
- [22] php: *MySQL Improved Extension*. 2015, [Online; navštívené 16.04.2018]  
URL <http://php.net/manual/en/book.mysql.php>
- [23] phpMyAdmin. 2018, [Online; navštívené 05.05.2018]  
URL <https://www.phpmyadmin.net/>
- [24] phpMyAdmin: *Documentation*. 2018, [Online; navštívené 05.05.2018]  
URL <https://www.phpmyadmin.net/docs/>
- [25] Pantz: *Cron*. 2015, [Online; navštívené 02.05.2018]  
URL <https://www.pantz.org/software/cron/croninfo.html>
- [26] Admin's Choice: *Crontab - Quick Reference*. 2013, [Online; navštívené 02.05.2018]  
URL <http://www.adminschoice.com/crontab-quick-reference>
- [27] NASA, Earth Fact Sheet. 2019, [Online; navštívené 02.05.2019]  
URL <https://nssdc.gsfc.nasa.gov/planetary/factsheet/earthfact.html>
- [28] Apixu, pricing. 2019, [Online; navštívené 21.04.2019]  
URL <https://www.apixu.com/pricing.aspx>
- [29] Dark Sky, documentation. 2019, [Online; navštívené 21.04.2019]  
URL <https://darksky.net/dev/docs>

[30] Weatherbit, pricing. 2019, [Online; navštívené 21.04.2019]  
URL <https://www.weatherbit.io/pricing>

[31] Google maps, JavaScript documentation. 2019, [Online; navštívené 21.04.2019]  
URL <https://developers.google.com/maps/documentation/javascript/tutorial>

[32] PHP, GD documentation. 2019, [Online; navštívené 21.04.2019]  
URL <https://www.php.net/manual/en/book.image.php>

[33] AccuWeather, pricing. 2019, [Online; navštívené 21.04.2019]  
URL <https://developer.accuweather.com/packages>

# Prílohy

## Príloha A

### Obsah priloženého CD

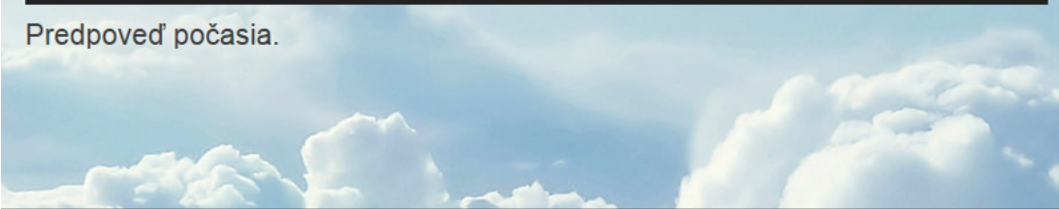
- **index.php** - domovská stránka aplikácie
- **map.php** - podstránka pre mapu s bežcom
- **stats.php** - podstránka pre štatistiky o polohe a vzdialenosti
- **sprava.php** - podstránka pre webovú administráciu
- **main.css** - súbor obsahujúci kaskádové štýly ku všetkým stránkam
- **rens\_functions.php** - často používané php funkcie
- **part\_body1.php** - miniaplikácia pre každú stránku
- Collectors
  - **apixu\_db\_update.php** - stahovanie hodinovej predpovede od Apixu
  - **dark\_db\_update.php** - stahovanie hodinovej predpovede od Dark Sky
  - **owm\_db\_update.php** - stahovanie hodinovej predpovede od Open Weather
  - **wb\_db\_update.php** - stahovanie hodinovej predpovede od Weatherbit
  - **accu\_db\_update.php** - stahovanie hodinovej predpovede od AccuWeather
  - **apixu\_daily\_db\_update.php** - stahovanie dennej predpovede od Apixu
- Dokumentacia
  - **dokumentacia.indd** - zdrojový súbor dokumentácie (Adobe InDesign)
  - **dokumentacia-web.pdf** - dokumentácia v nízkej kvalite pre rýchle zobrazenie
  - **dokumentacia-tlac.pdf** - dokumentácia v tlačovej kvalite
  - **ostatné** - obrázky a jejich zdrojové súbory
- Grafický návrh
  - **index.psd** - zdrojový súbor grafického návrhu (Adobe Photoshop)
  - **index.jpg** - obrázok grafického návrhu
  - **graf.png** - obrázok chart.js grafu
- Samples
  - **JSON ip-api.txt** - vzorová odpoveď od ip-api.com
  - **JSON pre Brno od Google.txt** - vzorová odpoveď od google geocoding API
  - **JSON pre Brno od Open Weather Map.txt** - ukážka predpovede počasia
  - **JSON pre Brno od YAHOO.txt** - ukážka predpovede počasia od Yahoo!
- Ostatne
  - **www.powerpointhintergrund.com cloud-wallpaper-13.jpg** - header

# Príloha B

## Náhľad aplikácie (PC)

Domov      Mapa počasia      Štatistiky      Správa

### Predpoveď počasia.



Predpoveď na 10 dní podľa vašej lokácie. Zlín, 144km.

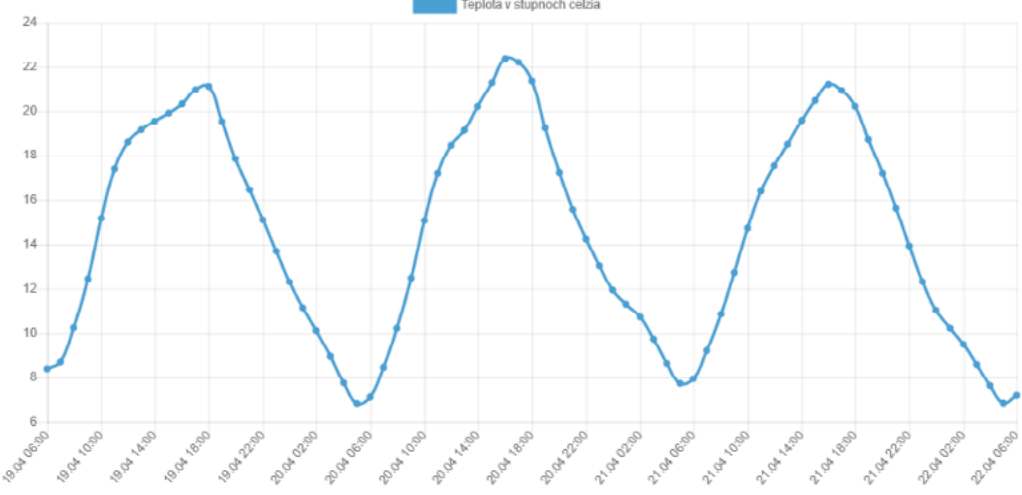
19 Apr	20 Apr	21 Apr	22 Apr	23 Apr	24 Apr	25 Apr	26 Apr	27 Apr	28 Apr
18°/7°	18°/8°	16°/6°	17°/7°	11°/5°	14°/10°	18°/10°	20°/13°	18°/10°	9°/8°

Mesto:  Čas:

Zobrazíť dáta z:  Aqiru  Dark Sky  Open Weather Map  Weatherbit  AccuWeather

Predpoveď počasia pre Brno, vzdialenosť 64 km.

Teplota v stupnoch celzia



Time	Temperature (°C)
19.04.06:00	8
19.04.10:00	12
19.04.14:00	18
19.04.18:00	21
19.04.22:00	16
20.04.02:00	10
20.04.06:00	7
20.04.10:00	12
20.04.14:00	18
20.04.18:00	22
20.04.22:00	16
21.04.02:00	10
21.04.06:00	7
21.04.10:00	12
21.04.14:00	18
21.04.18:00	21
21.04.22:00	16
22.04.02:00	10
22.04.06:00	7

Vypracované ako bakalárska práca pre Vysoké Učenie Technické v Brně,  
Fakulta informačních technologií, 2018-2019, Kamil Mucha