



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 2018

Zadání bakalářské práce

Řešitel: **Mucha Kamil**

Obor: Informační technologie

Téma: **Webová aplikace pro předpověď počasí**
Web Application For Weather Forecast

Kategorie: Informační systémy

Pokyny:

1. Seznamte se s principy tvorby webových a mobilních aplikací. Prostudujte také existující servery, ze kterých lze stahovat informace o předpovědi počasí. Zvolte 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 a přesný čas od-do. Pokud se data pro zadanou lokaci v databázi nenacházejí, bude vyhledána nejbližší dostupná lokalita. Mobilní část bude data stahovat z databáze webové aplikace a zobrazovat je vhodnou formou.
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
- Ujbányai, M.: Programujeme pro Android. Grada Publishing, 2012, ISBN 978-80-247-3995-3.

Pro udělení zápočtu za první semestr je požadováno:

- Splnění bodů zadání 1 až 3.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Bartík Vladimír, Ing., Ph.D.**, UIFS FIT VUT

Datum zadání: 1. listopadu 2017

Datum odevzdání: 16. května 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
602 00 Brno, Božetěchova 2

doc. Dr. Ing. Dušan Kolář
vedoucí ústavu

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, 2018. 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
16.5.2018

Pod'akovanie

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

Obsah

1 Úvod	3
2 Získavanie údajov	4
2.1. Web API	4
2.2. Web API pre predpoveď počasia od Yahoo!	4
2.3. Web API pre predpoveď počasia od Open Weather Map	5
2.4. Web API pre predpoveď počasia od World Weather Online	7
2.5. Web API pre zemepisné súradnice od Ip-Api.com	7
2.6. Web API pre zemepisné súradnice od Google APIs	8
3 Návrh Riešenia	9
3.1. Grafický Dizajn	9
3.2. Telo webovej stránky	10
3.3. Voľba lokalít	11
3.4. Databáza	11
3.5. Spracovanie získaných dát	12
3.6. Vzďialenosť dvoch bodov určených zemepisnou šírkou a dĺžkou	12
3.7. Návrh mobilnej aplikácie	13
4 Použité technológie	15
4.1. HTML	15
4.2. CSS - Kaskádové štýly	15
4.3. JavaScript	16
4.4. SQL a MySQL	16
4.5. PHP	17
4.6. phpMyAdmin	17
4.7. Chart.js	18
4.8. Date Range Picker	19
4.9. Cron a Crontab	19
5 Implementácia	21
5.1. Ukladanie a aktualizácie údajov	21
5.2. Implementácia výpočtu vzdialenosti dvoch bodov určených zemepisnou šírkou a dĺžkou v PHP	22
5.3. Web Stránka	23
5.4. Mini-aplikácia na zobrazenie predpovede počasia na 10 dní na základe lokácie návštevníka	24
5.5. Implementácia hlavného grafu predpovede počasia	24
5.6. Hybridná mobilná aplikácia (Cordova)	25
6 Možné rozšírenia	27
7 Záver	28
8 Literatúra	29
9 Prílohy	31

Zoznam obrázkov

2.1 Ukážka funkcie autocomplete od Google	8
3.1 Návrh palety farieb pre webovú aplikáciu	9
3.2 Návrh dizajnu pre menu	9
3.3 Návrh dizajnu pre header	10
3.4 Mini-aplikácia predpovede počasia na 10 dní	10
3.5 ER diagram databáze	12
4.1 Zjednodušený HTML DOM zobrazujúci HTML elementy	16
4.2 Ukážka chart.js grafov	18
4.3 Date range picker	19
5.1 Nastaviteľný graf predpovede počasia	25

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. Takže užívateľ ktorý si chce vyhľadať detailnú predpoveď počasia na konkrétne tri dni si musí najprv vyhľadať svoju lokalitu a potom si musí tieto stĺpce rozkliknúť pre detailnejšiu predpoveď počasia v tých dňoch.

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 takýchto 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 zdarma. Preto by som skoršie zvolil tento prístup a dáta si pravidelne sťahoval.

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

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

Prácu by bolo možné ešte rozšíriť mobilnou aplikáciou ktorá by zobrazovala webovú stránku a ešte by si bolo dobré zistiť približnú polohu návštevníka stránky a pri prvom načítaní stránky mu zobrazit' predpoveď počasia pre jeho lokalitu.

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 strániek a serverov ktoré tieto dáta ponúkajú, niektoré z nich sú avšak spoplatnené. Pre tento projekt by bolo vhodnejšie zvoliť zdroj ktorý poskytuje dáta zdarma.

Pre rozšírenejšie služby tohoto 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 tohoto by sme si mohli vytvoriť zoznam lokalít pre ktoré máme dostupné dáta.

Avšak ku kompletnému vypracovaniu aplikácie bude ešte potrebné získať zemepisné súradnice lokality pre ktorú si užívateľ chce 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 posiela 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 tohoto kľúča dokáže server rozpoznať kto si žiada aké dáta a môže týmto aj monitorovať počet prístupov k dátam.

2.2. Web API pre predpoveď počasia od Yahoo!

Yahoo poskytuje prístup k celosvetovým dátam o počasi. Tieto dáta sú verejne dostupné a je možné ich sťahovať bez registrácie. Ohľadom aktuálneho počasia zahŕňajú:

- teplotu,

- vietor,
- vlhkosť ovzdušia,
- atmosférický tlak,
- viditeľnosť,
- východ a západ slnka,
- zemepisné súradnice lokácie,
- textový údaj opisujúci počasie (zamračené, jasno, dážď...).

Avšak predpoveď počasia je veľmi 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 veľmi 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 https://query.yahooapis.com/v1/public/yql?q=select%20*%20from%20weather.forecast%20where%20woeid%20%3D%20786869%20AND%20u%20%3D%20'c'&format=json&env=store%3A%2F%2Fdatatables.org%2Falltables%2Fwithkeys 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 databázy. 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 doporučil ako zdroj dát pre nejaké jednoduchšie aplikácie, prístup k dátam je veľmi 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 vznikáť vôbec žiadne obmedzenia. Pre aktuálne počasie je dostupných veľa informácií ktoré boli vymenované avšak predpoveď počasia je veľmi obmedzená, toto je hlavný dôvod prečo som sa rozhodol tento zdroj dát využiť iba na menšiu aplikáciu na mojej webovej stránke a pre hlavný graf vyhľadať nejaký iný zdroj.

2.3. Web API pre predpoveď počasia od Open Weather Map

Open Weather Map poskytuje veľmi 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 troj-hodinových

intervaloch, týmto môžeme získať až 40 záznamov pre jednu lokalitu. Tieto údaje zhrňujú:

- id a názov mesta,
- zemepisné súradnice lokácie,
- populáciu mesta,
- unixový čas,
- teplota,
- atmosferický tlak,
- vlhkosť ovzdušia,
- textový údaj opisujúci počasie a jeho podrobnejší popis
- mračnosť,
- rýchlosť a smer vetra,
- dážď,
- sneh,
- dátum a čas vo formáte ISO 8601 (YYYY-MM-DD HH:MM:SS)

Zdarma 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 pre predpoveď počasia na 5 dní sú:

- 7200 prístupov za deň
- 60 za minútu
- prístup na rovnakú lokáciu iba raz za 10 minút.

Spoplatnená ponuka je rozdelená na viacej 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 zvýšeny limit prístupov na 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 viacero 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 tohoto zdroja je široký objem dát ktorý zhrňuje všetky údaje o počasí ktoré by správna webová aplikácia o predpovedi počasia mala poskytovať. Zdarma je možné získať predpoveď na 5 dní kde sú záznamy rozložené 3 hodinových intervaloch.

Avšak pri bezplatnom využívaní tejto služby, je počet prístupov k dátam cez web API velice limitovaný. Preto by bolo vhodné zvoliť iba niektorých pár lokalít pre ktoré si aplikácia bude sťahovať a aktualizovať dáta.

Kvôli objemu a rozsahu dát som sa rozhodol použiť tento zdroj ako základ pre jadro mojej bakalárskej práce. Ponúkané záznamy naozaj obsahujú všetky potrebné informácie o predpovedi počasia, iked' záznamy sú iba na 5 dní dopredu v troj hodinových intervaloch.

2.4. Web API pre predpoveď počasia od World Weather Online

World Weather Online[5] poskytuje kompletne 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 záznami obsahujú podrobnosti o vlnách ako perióda a výška vln a záznami 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 pri každej lokalite dostali až 336 záznamov. Tieto záznami obsahujú

- zemepisné súradnice,
- časová značka,
- východ a západ slnka,
- východ, západ, fáza a osvetlenie mesiaca,
- minimálna a maximálna teplota,
- index tepla,
- rýchlosť a smer vetra,
- popis počasia,
- zrážky a sneh,
- vlhkosť ovzdušia,
- viditeľnosť,
- atmosferický tlak,
- uv index,
- oblačnosť.

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 poplatný režim po 60 dňoch používania som sa rozhodol tento zdroj dát nevyužiť vo svojej práci.

2.5. Web API pre zemepisné súradnice od Ip-Api.com

[Ip-Api.com](https://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 tohoto nástroja nebudú až tak presné a budú často smerovať na väčšie mestá. Na základe IP adresy sa dá získať údaje o poskytovateľovi ktorý danú IP adresu spravuje. Tieto údaje zahrňujú:

- 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].

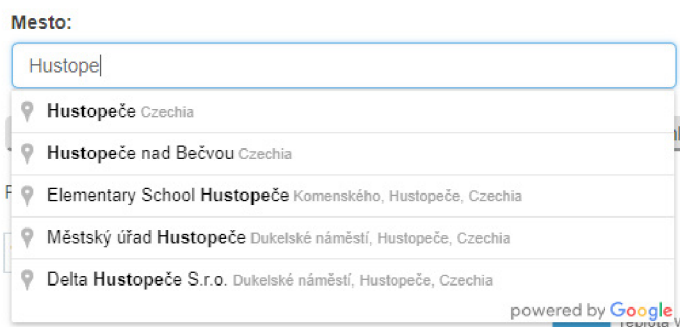
Túto službu by bolo vhodné využiť na zistenie približnej lokality návštevníka webovej stránky.

2.6. Web API pre zemepisné súradnice od Google APIs

Google vlastní a spravuje mnoho projektov.[8] Jedným z nich je Google Maps [9]. Kvôli to-
muto 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 príst-
up zdarma cez web API (je ale potrebný API kľúč) a tohoto 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=Br-
no,+Czechia&key=KEY`

Google ponúka prístup zdarma k omnoho viacej funkciám a dátam. Pre tento projekt budú
stačiť funkcie geocoding a autocomplete. Funkcia autocomplete[10] napomáha užívateľom
správne zadať názov lokality do vyhľadávania. Využitím tejto funkcie znížime počet chybné
zadaných lokalít a napomôžeme užívateľovi 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.



Obrázok 2.1 - Ukážka funkcie autocomplete od Google

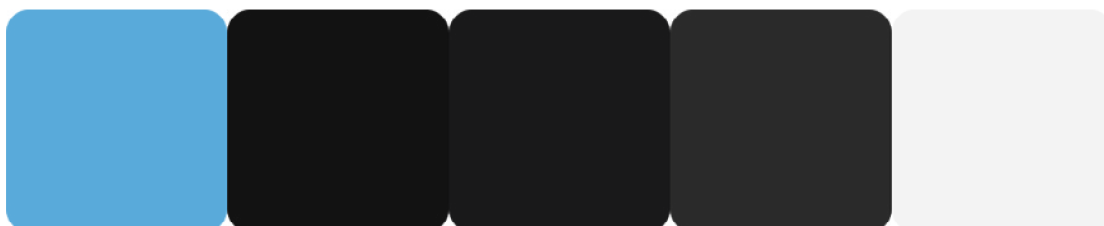
Kapitola 3

Návrh Riešenia

K vypracovaniu tejto práce bude potrebné vhodne zvoliť zdroje dát pre aplikáciu, navrhnuť štruktúru databáze a parsér ktorý bude databázu naplňať dátami a udržiavať ju aktualizovanú. Pre webovú stránku bude potrebné vhodne zvoliť farby, navrhnuť štruktúru stránky tak, aby každý plánovaný prvok mal dostatok miesta a na záver vypracovať grafický návrh všetkých prvkov na základe vybranej palety farieb.

3.1. Grafický Dizajn

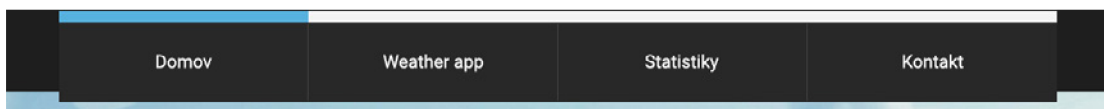
Pre webovú aplikáciu pre predpoveď počasia by som zvolil bledo modrú farbu k téme a niekoľko čiernobielych farieb pre eleganciu webovej stránky.



Obrázok 3.1 - Návrh palety farieb pre webovú aplikáciu

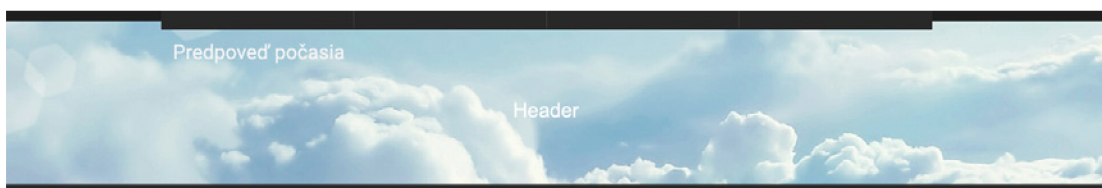
- Modrá #59acde
- Čierna #111111
- Bledo čierna #1a1a1a
- Sivá #2a2a2a
- Biela #f3f3f3

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.

3.2. 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 posielať 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 tlačítek ktoré by menili typ dát ktoré graf zobrazuje. Pre tieto tlačítka by som 5 kategórii:

- teplota
- vietor
- zrážky
- vlhkosť
- oblačnosť

3.3. Voľba lokalít

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 pre začiatok 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 WOEID od Yahoo! a ID lokality od Open Weather Map.

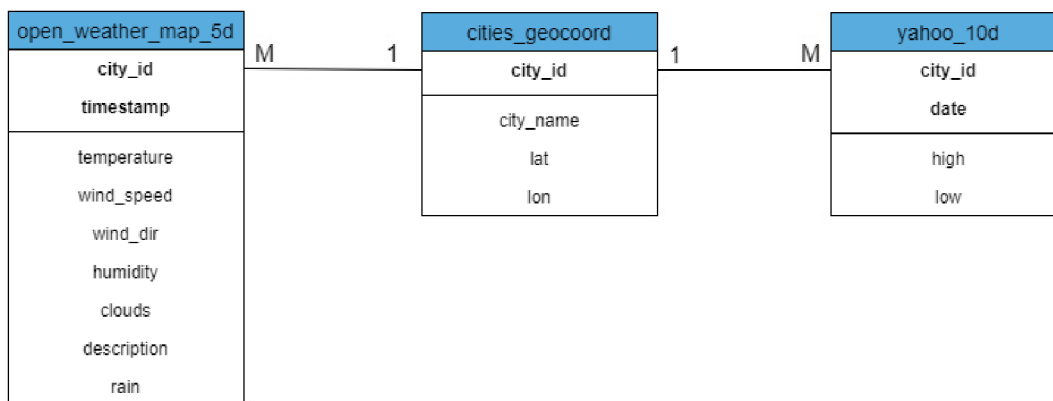
Mesto	WOEID	Open Weather Map ID
Praha	796597	3067695
Ostrava	795610	3068799
Brno	786869	3078610
Ústí nad Labem	800260	3063548
Olomouc	795429	3069011
České Budějovice	787461	3077916
Zlín	802199	3061370
Plzeň	796166	3068160
Hradec Králové	790133	3074967
Jihlava	790803	3074199
Pardubice	795793	3068582
Liberec	792768	3071961
Karlovy Vary	791140	3073803

Tabuľka 3.1: Vybrané lokality

3.4. Databáza

Návrh databáze je celkom jednoduchý. Získavame informácie od zdrojov Yahoo! a Open Weather Map. Každý z týchto zdrojov bude mať vlastnú tabuľku pretože poskytujú rôzne množstvo a typy informácií. Každý záznam predpovede počasia sa ale ešte viaže na nejaké mesto o ktorom potrebujeme mať uložené údaje jeho názvu a jeho zemepisných súradníc. Týmto by sme ale v týchto dvoch tabuľkách mali veľa duplicitných údajov a je preto potrebné vytvoriť tretiu tabuľku pre lokality.

Ako primárne kľúče pre tabuľky s predpovedov počasia som zvolil dvojicu id mesta a časovú značku daného záznamu. Táto dvojica je pre každý záznam unikátna a id mesta sa dá použiť pre spojenie s tabuľkou cities_geocoord pre vyhľadanie názvu mesta a zemepisných súradníc ku ktorým sa daný záznam vzťahuje.



Obrázok 3.5 - ER diagram databáze

3.5. Spracovanie získaných dát

Vďaka vopred urobenému návrhu tabuliek databáze by spracovanie informácií o predpovedi počasia malo byť o dosť jednoduchšie.

Od Yahoo! bude potrebné sťahovať dennú predpoveď počasia na 10 dní, týmto by sme mali vytvoriť 10 záznamov do databáze pre každú lokalitu. Tieto záznamy nebudú ale slúžiť pre hlavný graf, budú použité iba ako rýchle zobrazenie predpovede počasia na základe zistenej lokality návštevníka webovej aplikácie, preto si z nich pri aktualizáciách nemusíme ukladať staršie záznamy a môžeme ich rovno mazať.

U údajov z Open Weather Map ktoré budú zobrazované na hlavnom grafe si ale chceme nechávať staré záznamy a tým si vytvárať históriu počasia pre sledované lokality. Z tohoto zdroja získavame informácie o predpovedi počasia na 5 dní a záznamy sú 3 hodinových intervaloch. Týmto by sme mali získať 40 záznamov pre každú lokalitu a ohľadom histórie počasia musíme počítať s tým že každé 3 hodiny nám vznikne jeden záznam pre každú lokalitu. Týmto by sme za 30 dní mali dostať celkovo až 3120 záznamov o histórii počasia pre našich vybraných 13 lokalít.

Ohľadom zemepisných údajov to bude ešte jednoduchšie. Z Open Weather Map využijeme zemepisné súradnice pre každú z 13 vybraných lokalít a tieto údaje budú tvoriť tabuľku cities_geocoord. Zemepisné súradnice od ip-api.com a google geocoding budú slúžiť na získanie súradníc návštevníka webovej aplikácie a získanie súradníc mesta pre ktoré si praje zobrazíť dáta v grafe, takže tieto údaje budú využité na jednorázove použitie a nebude si ich potrebné ukladať do databáze.

3.6. Vzdialenosť dvoch bodov určených zemepisnou šírkou a dĺžkou

Zo záznamov o predpovedi počasia vieme získať zemepisné súradnice ku ktorým sa vzťahujú. Z tohoto si dokážeme vytvoriť zoznam lokalít pre ktoré máme dostupné dáta. Problém vzniká pokiaľ ale návštevník má záujem o lokalitu pre ktorú nemáme žiadne záznamy v databáze. Za pomoci google api si dokážeme získať jej presné zemepisné súradnice, tým pádom sa dostávame do situácie kedy musíme zistiť vzdialenosť tejto lokality a našich dostupných lokalít na

základe dvoch dvojíc zemepisných súradníc.

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 pólou 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 pólou. Čiže pre výpočet zemepisnej dĺžky by sme mohli použiť vzorec

$$\text{vzdialenosť jedného stupňa zemepisnej dĺžky v kilometroch} = \cos(\text{zemepisná šírka v stupňoch}) * 111.321$$

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

3.7. Návrh mobilnej aplikácie

V dnešnej dobe existujú omnoho výkonnejšie mobilné zariadenia ako niekoľko rokov dozadu. Zvyčajne sa jedná o smartfóny a tablety. Postupom času ako sa zlepšoval jejich hardware tak pribúdali mnohé náročnejšie aplikácie, čiže z prenosného zariadenia ktoré malo budík, stopky a kalendár vzniklo zariadenie so širokou škálou aplikácii ku ktorým je možné si jednoducho stiahnuť a nainštalovať ďalšie. Internetové pripojenie tiež nieje problém, vďaka wifi hotspotom a 3G a 4G sietiam. Takže by bolo možné vytvoriť mobilnú aplikáciu pre predpoveď počasia bez toho aby sme museli byť nejak príliš obmedzený výkonom, inštaláciou alebo pripojiteľnosťou k sieti. Avšak oveľa väčším problémom bude kompatibilita s operačným systémom zariadenia a aktualizácie aplikácie. Preto je potrebné si zvoliť akým spôsobom bude táto aplikácia vyvíjaná.

Natívne mobilné aplikácie sú vyvíjané pre špecifickú platformu a jejich najväčšou výhodou je rýchly prístup k funkciám zariadenia. Čiže dokážu využívať napríklad GPS, akcelerometer, kompas, kameru a všetky ostatné funkcie zariadenia. Avšak je potrebné si ich stiahnuť na mobilné zariadenie a nainštalovať ich. Tieto aplikácie taktiež dokážu pracovať bez prístupu na internet, všetky potrebné údaje môžu mať uložené v databázi a pokiaľ získajú prístup k internetu môžu si dáta v databázi aktualizovať. V našom prípade je ale problém v tom, že máme dostupné iba dáta o predpovedi počasia pre hlavných graf na 5 dní, avšak funkcia GPS by nám pomohla pre zistenie zemepisných súradníc používateľa, lenže aj tá sa dá vypnúť alebo ju zariadenie/platforma nemusí vôbec mať.

Webové mobilné aplikácie sú v podstate webové stránky na ktoré sa dá dostať cez prehliadač mobilného zariadenia. Nie je potrebná žiadna inštalácia a fungujú na všetkých platformách. Jejich nevýhodou je ale to že vyžadujú prístup na internet k tomu aby sa mohli dostať na danú webovú stránku, avšak vďaka tomu budú zobrazené dáta na webovej stránke vždy čerstvé. Pre tvorbu takejto aplikácie je iba potrebné zaručiť že daná webová stránka má zodpovedný dizajn a bude prehľadná aj na zariadeniach s malým displejom.

Hybridné mobilné aplikácie sú kombinácia dvoch predošle zmienených typov. Je potrebné si ich stiahnuť ako natívnu mobilnú aplikáciu a nainštalovať ich na mobilné zariadenie. Taktiež dokážu využívať funkcie mobilného zariadenia ako GPS. Jejich jadro ale tvorí HTML kód, takže je jednoduché ich vytvoriť už z existujúcich stránok a tento kód je prenositeľný na ostatné platformy a operačné systémy, takže taktiež podporujú crossplatformový vývoj.[15][16]

Pre začiatok som sa rozhodol vypracovať zodpovedný dizajn pre webovú stránku a tým vytvoriť webovú mobilnú aplikáciu. Týmto spôsobom tvorby mobilnej aplikácie nie je potrebné ju inštalovať na zariadení a je jednoducho dostupná z akéhokoľvek zariadenia s internetovým pripojením a prehliadačom. Taktiež je zaručené že predpoveď počasia zobrazená na tejto stránke bude vždy aktuálna.

Toto by ale bolo možné ešte rozšíriť o hybridnú mobilnú aplikáciu ktorú by bolo možné si stiahnuť a nainštalovať pre mobilné zariadenie. Pre túto implementáciu by som ale zvolil iba jednu špecifickú platformu pre vývoj a testovanie a možné prenášanie na iné platformy by som nechal na neskôr po kompletnom dokončení aplikácie.

Kapitola 4

Použité technológie

Pred tým než sa dostaneme k samotnej implementácii tohoto projektu rád by som zmienil a vysvetlil základné údaje a princípi o technológiách ktoré boli využité.

4.1. HTML

HTML (skratka pre HyperText Markup Language) je značkový jazyk pre tvorbu webových stránok. Hlavné využitie tohoto 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 `
` 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.

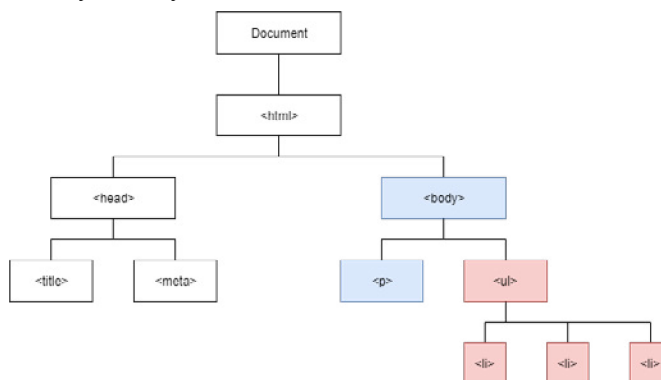
4.2. CSS - Kaskádové štýly

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 zobrazit' 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 X. Aplikáciou modrej farby textu na `<body>`, sa modrá farba neprenesie na ``, pokiaľ aplikujeme červenú farbu priamo na značku ``. [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 myškou. Týmto vznikajú ale nejaké obmedzenia, užívateľovy je vždy poslaný celý zdrojový kód aby si ho mohol spustiť, lenže dá sa do neho ľahko nahliadnuť. Taktiež JavaScript nedokáže pracovať so súborami, 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, relácie. Za pomoci týchto relácií je možné tabuľky spájať, napríklad v našom prípade je možné spojiť každý jeden záznam z `open_weather_map_5d` tabuľky s práve jedným záznamom z `cities_geocoord` a tým k údajom o počasi 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 (skratka pre anglický pojem Structured Query Language) je štruktúrovaný dotazovací jazyk. Tento jazyk je využívaný pre prácu s relačnými databázami. Príkazy SQL sa delia na tri základné skupiny a to:

- správa tabuliek databáze (CREATE, DROP...)
- manipulácia dát (SELECT, UPDATE...)
- správa dát (COMMIT, GRANT)
- 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. Už 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 vykonávajú 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 tohoto 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 zmeniť 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 tohoto nástroja a jejich používanie. [24]

4.7. Chart.js

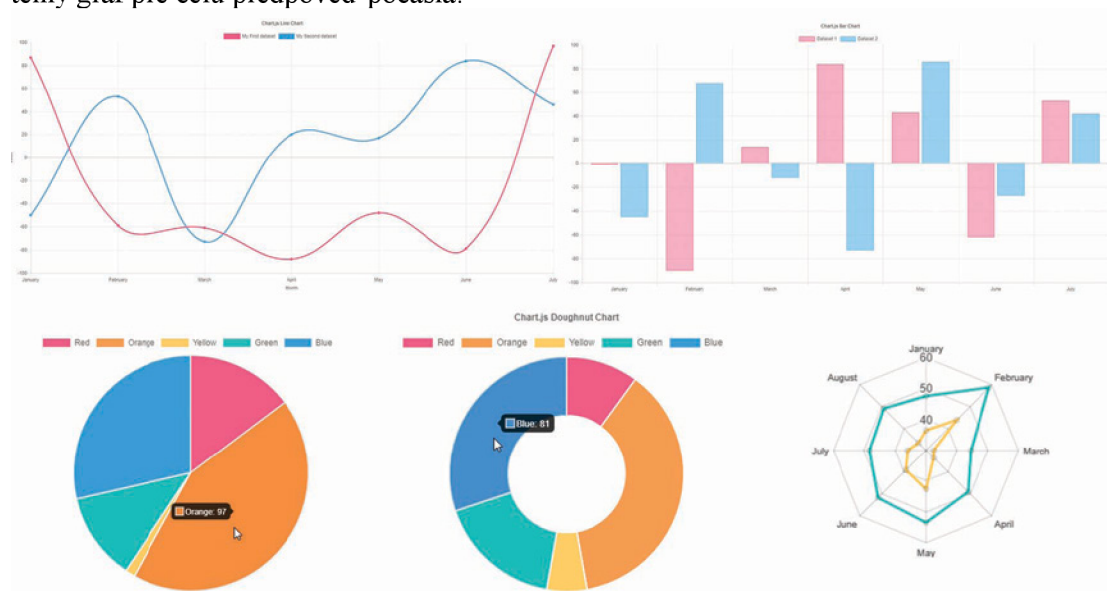
Chart.js je nástroj ktorý uľahčuje tvorbu grafov pre webové stránky. Je napísaný v JavaScripte. Verzia 2.7.2 ktorú budem pre tento projekt používať sa dá velice jednoducho získať kódom:

```
<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.

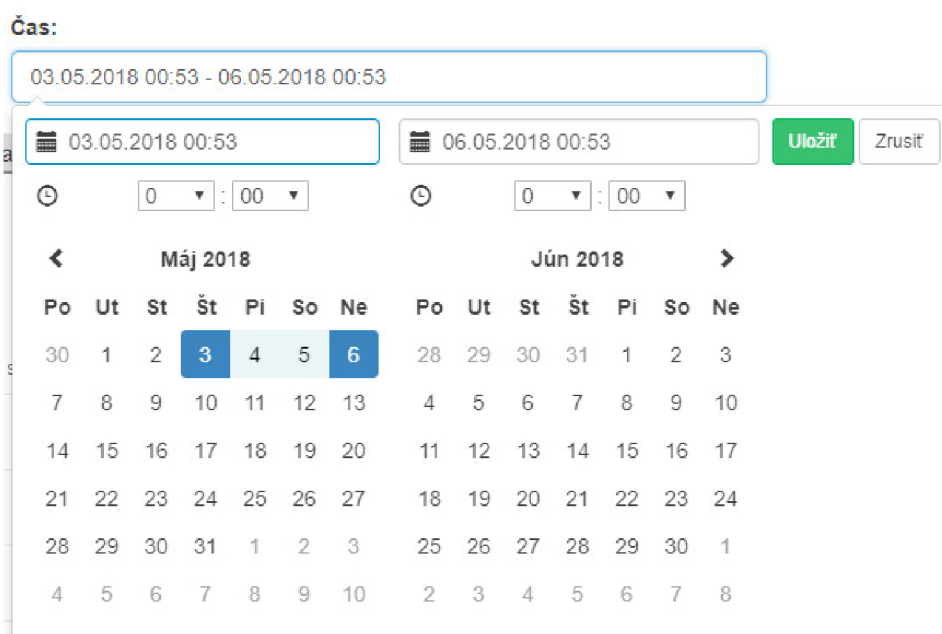


Obrázok 4.2 - Ukážka chart.js grafov

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 tohoto nástroja do slovenského jazyka. V tom prípade by formát dátumu a času bol zapísaný ako DD.MM.YYYY HH:mm.



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. Tohoto 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. Pre vykonanie php skriptu v hodinových intervaloch musíme do tabulky pridať riadok:

```
04 * * * * wget -O /dev/null http://renezis.eu/forecast/db_update.php
```

Týmto by sa *db_update.php* spustil na mojom servery v štvrtú minútu každej hodiny.

Kapitola 5

Implementácia

Implementáciu projektu som začal vytvorením dvoch parserov pre dáta o predpovedi počasia [Kapitola 5.1]. Tieto dva parsery naplňajú a udržuujú databázu aktualizovanú. K týmto parsérom som si vytvoril ešte pomocný php súbor na funkcie kde som taktiež implementoval funkciu na výpočet vzdialenosti na základe haversine formuly [Kapitola 5.2]. Takže týmto som už mal dosť veľkú časť back-endu hotovú a začal som tvorbou webovej stránky [Kapitola 5.3]. Po vypracovaní štruktúry a dizajnu som implementoval aplikáciu pre rýchle zobrazenie predpovede počasia na 10 dní podľa lokácie návštevníka [Kapitola 5.4]. Na záver bolo potrebné pridať hlavný graf do tela stránky, napojiť jeho tlačítka pre výber správnych dát a potom cez php preniesť vybrané dáta do JavaScriptu a v grafe ich zobrazíť [Kapitola 5.5].

5.1. Ukladanie a aktualizácie údajov

Projekt bude využívať dáta z dvoch zdrojov, Yahoo! a Open Weather Map. Tieto dáta bude ale potrebné pravidelne sťahovať, parsovať a ukladať do MySQL databáze. Prácu a uloženie týchto dát som vypracoval v jazyku PHP. Zdroje poskytujú odlišné dáta, preto sú výsledkom dva rôzne php súbory. Pre udržanie databáze aktualizovanej bude potrebné tieto dva súbory spúšťať v pravidelných hodinových intervaloch. K tomuto som využil crontab svojho serveru ktorý tieto dva súbory php pravidelne spúšťa.

Yahoo! dovoľuje stiahnutie viacero lokalít jedným prístupom na web API. Čiže funkciou `file_get_contents("Web API URL")` získavame textový reťazec ktorý obsahuje údaje o predpovedi počasia pre našich 13 lokalít. Tieto údaje sú v JSON štruktúre, čiže funkciou `json_decode(string, true)` získame pole kde už jednoduché v PHP adresovať jednotlivé prvky tohoto pola. Ešte pred uložením týchto údajov som si trochu upravil štruktúru tohoto pola a pozmenil časové značky na Y-m-d formát aby sedeli do MySQL databáze. Kvôli tomu že tabuľky databáze sú navrhnuté tak, že obe predpovede počasia zdieľajú jednu tabuľku `cities_geocoord` s údajom názvu mesta je potrebné zaistiť že názvy miest sa budú zhodovať. Taktiež ich je treba upraviť nech sú všetky v češtine a s diakritikou. Samotné ukladanie dát do databáze som sa rozhodol vykonať tým, že sa obsah celej tabuľky `yahoo_10d` zmaže a to je z toho dôvodu že nemáme záujem si udržiavať staršie záznamy z minulosti. V prípade Yahoo! nás zaujíma iba predpoveď na 10 dní, takže všetky staršie záznamy môžeme zmazať a tabuľku naplniť novými záznamami.

Open Weather Map ponúka zdarma prístup k jednotlivým lokalitám cez web API. Kvôli tomuto bude potrebné mať 13 odkazov na web API a použiť funkcie `file_get_contents()` a `json_decode()` 13 krát. V prípade rozšírenia by bolo vhodnejšie prejsť na spoplatnený "bulk" prenos (viac v kapitole 6). Týmto postupne získame pole dát ktoré obsahujú 40 záznamov pre každú lokalitu. Názvy miest celkom sedia, avšak chýba im diakritika a je ešte potrebné upraviť

názov pre Prahu. Časové značky sú vo vhodnom formáte pre SQL databázu preto ich ani nieje potrebné upravovať. Pri vkladani údajov do databáze ako prvé som implementoval uloženie lokalít do tabuľky cities_geocoord. Pokiaľ mesto s takýmto názvom už nieje v tabuľke tak bude pridané. Týmto dostávame zoznam miest, jejich zemepisné súradnice a vďaka auto_increment funkcií od MySQL získavame aj unikátne ID pre túto lokalitu. Tieto údaje z Open Weather Map sú pre projekt najdôležitejšie, preto sa tabuľka cities_geocoord naplňa na základe týchto záznamov a Yahoo tabuľka sa už podľa názvu lokality naviaže na id týchto lokalít. Pre samotné ukladanie záznamov o predpovedi počasia je potrebné si udržať staršie záznamy, aktualizovať uložené záznamy a pridať nové, pokiaľ nejaké sú. Tohoto som dosiahol využitím SQL príkazu

```
INSERT INTO open_weather_map_5d (...) VALUES (...) ON DUPLICATE
KEY UPDATE ...
```

Z hľadiska testovania boli tieto skripty uložené na mojom servery a boli pravidelne spúšťané crontabom už vyše 20 dní. Do databáze som pravidelne nahliadal a všetky záznamy a dáta boli správne ukladané a aktualizované.

5.2. Implementácia výpočtu vzdialenosti dvoch bodov určených zemepisnou šírkou a dĺžkou v PHP

Cez web API si dokážeme od google získať zemepisné súradnice mnohých miest a lokalít. Avšak pre zistenie najbližšej lokality o ktorej máme dáta je potrebné vypočítať vzdialenosť medzi nimi. Preto by bolo rozumné implementovať funkciu ktorá túto vzdialenosť vypočíta na základe dvojice zemepisných súradníc. Vzdialenosť budem počítať haversine formulou [13], takže ju bude potrebné implementovať v PHP.

$$a = \sin^2(\Delta\text{lat}/2) + \cos(\text{lat}_1) \cdot \cos(\text{lat}_2) \cdot \sin^2(\Delta\text{lon}/2)$$

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

$$d = R \cdot c$$

R je polomer zeme, ktorý je 6 371 kilometrov[27], avšak pre výsledkoch v metroch ho zadáme ako 6 371 000 metrov. Implementácia tejto formule je na základe vzorca už celkom jednoduchá, pretože v PHP už máme k dispozícii funkcie ako sínus, kosínus, atan2 atď...

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 ktorú zistí cez ip-api.com. Tuto sú moje výsledky z lokality Nové Zámky, Slovensko.

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

Tabuľka 5.1: Vybrané lokality

5.3. Web Stránka

Po vypracovaní databáze, parserov a grafického návrhu webovej stránky som začal pracovať na tvorbe front-endu aplikácie. Štruktúra webovej stránky je implementovaná v jazyku HTML. Bola vytvorená domovská stránka a 3 podstránky:

- Domov
- Weather App
- Štatistiky
- Kontakt

Všetky tieto stránky majú rovnakú štruktúru, obsahujú:

- Menu
- Header
- Mini-aplikácia zobrazujúca predpoveď počasia na 10 dní ktorá využíva dáta od Yahoo!
- Telo
- Footer

Obsah týchto prvkov zostáva rovnaký pre všetky podstránky okrem tela. V domovskej stránke je v tele stránky zobrazený hlavný graf pre predpoveď počasia, nad ním sú všetky políčka a tlačítka pre jeho nastavenie. Tento graf zobrazuje dáta od Open Weather Map, čiže predpoveď na 5 dní + uložená história počasia. Podstránka weather app obsahuje údaje o QR kóde ktorý smeruje na domovskú stránku, týmto je možné získať rýchli prístup pre mobilné zariadenia na webovú mobilnú aplikáciu. K tomu je tam pridaný aj odkaz na stiahnutie hybridnej mobilnej aplikácie. V štatistikách sa nachádzajú údaje o návštevníkovi, jeho polohe a jeho vzdialenosti k ostatným mestám/lokalitám pre ktoré má aplikácia dostupné dáta. Podstránka kontakt má v tele skoršie taký placeholder ktorý sa neskôr dá jednoducho nahradiť za normálne údaje.

Pri tvorbe CSS je možné jednoducho použiť už vopred vytvorený css súbor, napríklad z bootstrap. Ja som si už ale v programe Adobe Photoshop vypracoval vzorový dizajn webovej stránky (ukážka v prílohách), ku ktorému sa chcem dopracovať. Implementácia CSS je celkom jednoduchá, avšak dosť časovo náročná. Taktiež je potrebné mať na pamäti že nie každé zariadenie má obrovský monitor s vysokým rozlíšením, preto sa jednotlivé prvky stránky budú musieť vedieť prispôbiť veľkosti monitoru.

5.4. 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 vypísane 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.

5.5. Implementácia hlavného grafu 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.

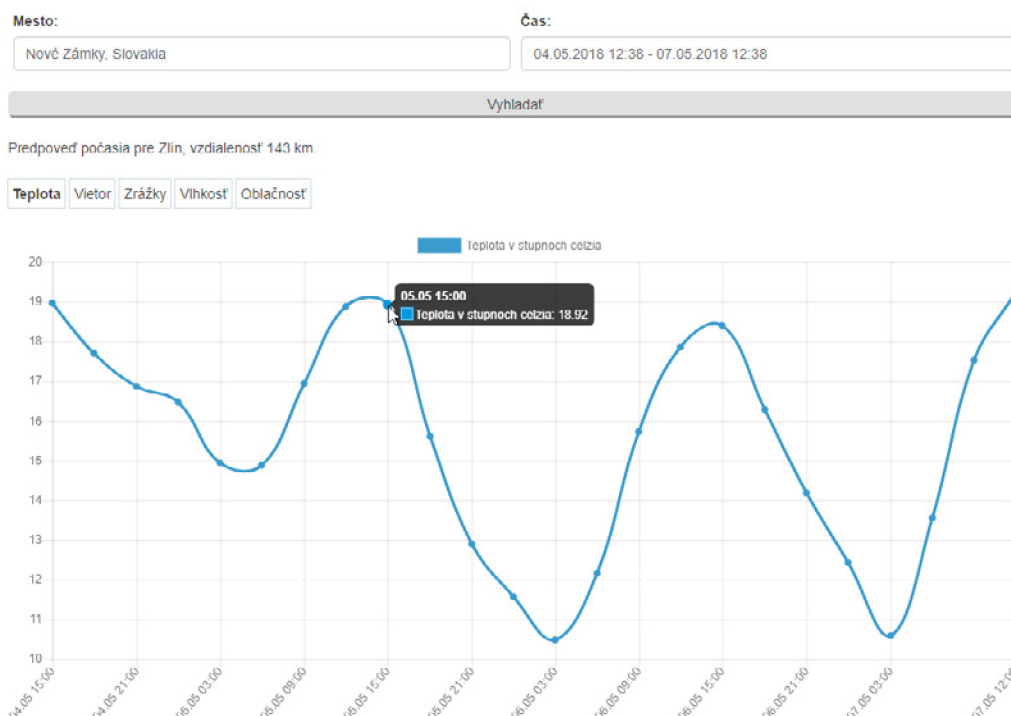
Pre výber správnych dát je potrebné aby užívateľ webovej aplikácie vyplnil dva údaje a to presnú lokalitu a časové rozpätie pre ktoré si chce dané dáta v grafe zobrazit'. Pre výber lokality som sa rozhodol využiť funkcie autocomplete od google. Táto funkcia nadväzuje na `<input type="text">` okienko a pokiaľ užívateľ začne písať nejaký text do tohoto okienka, funkcia mu ponúkne na výber 5 lokalít podľa textu ktorý tam už zadal. Pri výbere časového rozpätia je ale celkom problém s písaním dátumu a času v nejakom určenom formáte. Toto som sa rozhodol zjednodušiť využitím nástroja Date range picker. Pri otvorení okienka s časom sa užívateľovi zobrazí kalendár v ktorom si požadovaný dátum a čas môže naklikať, čo výrazne zjednodušuje zadávanie vstupu pre aplikáciu, ale hlavne pomáha užívateľovi zadať tento vstup v správnom formáte.

Spracovanie 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 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.

Výber dát z databáze bude zabezpečovať server, preto je implementovaný v PHP. Lenže s vybranými dátami už bude potrebné pracovať aj po načítaní stránky. Preto som si tieto dáta preniesol do JavaScriptu ktorý už pracuje na strane klienta. 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 osy 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 5.1 - Nastaviteľný graf predpovede počasia

5.6. Hybridná mobilná aplikácia (Cordova)

Mobilnú aplikáciu som vypracoval cez framework apache cordova[28]. Tento framework dokáže využiť dnešnú technológiu ktorá slúži k tvorbe webových stránok (HTML, CSS a JavaScript) a vytvorí z nej viacero mobilných aplikácií pre rôzne platformy.

Cordova má avšak viacero programov ktoré potrebuje pre vývoj týchto mobilných aplikácií, ako Java Development Kit, Gradle a Android Studio. U všetkých je potrebné taktiež nastaviť premenné prostredia. Takže iba spozrozenie vývojového prostredia nejakú chvíľu zaberie.

Pre vývoj mobilnej aplikácie som si ako cieľ vybral platformu android, a to konkrétne verziu 7.0.0 Cordova ale taktiež umožňuje jednoduché pridanie viacej platforiem ako:

- Amazon Fire OS
- Android
- BlackBerry 10
- Firefox OS
- iOS
- Ubuntu
- Windows Phone 8
- Windows
- Tizen

Celá mobilná aplikácia je vypracovaná ako náhľad webovej stránky. Čiže týmto je webová aplikácia zabalená ako mobilná aplikácia, je potrebné ju nainštalovať a bude ju možné spustiť a pracovať s ňou ako s normálnu mobilnou aplikáciou.

Pre testovanie tejto mobilnej aplikácie som využil android studio emulátor (obrázky v prílohách) v ktorom som emuloval telefón Nexus S s operačným systémom Android 7.0.0. ale taktiež som aplikáciu nainštaloval do mobilu OnePlus 2 s Androidom 6.0.1 kde bola aplikácia taktiež plne funkčná.

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.

U zdroja Yahoo! je rozšírenie o nové lokality a mestá celkom jednoduché, stačí u parséru pozmeniť odkaz na web API a pokiaľ je potrebné pridať korekciu názvu mesta/lokality.

Open Weather Map, pri ktorom využívam iba funkcie ktoré sú dostupné zdarma, má ale dosť veľké obmedzenia. Ako už bolo zmienené, sú tam limity na prístup k dátam a je možné pristupovať iba k jednej lokalite naraz. Tým pádom je potrebné parsér rozšíriť o ďalší odkaz na web API, avšak limit je 60 lokalít. Pre rozšírenie nad tento limit by bolo vhodné prejsť na platený "bulk download" u ktorého je možné získať dáta cez web API pre viacero lokalít naraz a je tam dokonca dostupná predpoveď počasia na 16 dní kde sú záznamy v hodinových intervaloch. Toto by ale vyžadovalo väčšie zásahy do parséru dát, avšak pokiaľ dáta správne naplnia políčka databáze, webová stránka ich už bude schopná správne zobrazit'.

Z pohľadu user experience je webová stránka vypracovaná aby bola prehľadná 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.

Ďalší vývoj tejto aplikácie by som pokračoval prácou na mobilnej aplikácii. Bolo by vhodné ju rozšíriť 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. Avšak momentálne je dostupná iba detailná predpoveď počasia na 5 dní, preto by bolo potrebné najprv prejsť na spoplatnenú verziu sťahovania predpovede počasia.

Kapitola 7

Záver

Cieľom tejto práce bolo navrhnúť a implementovať webovú a mobilnú aplikáciu pre predpoveď počasia. K tomu bolo ale potrebné naštudovať získavanie dát z cudzích zdrojov cez web API a prácu s nimi, správu databáze a vkladanie a výber dát, tvorbu webových a mobilných aplikácií a interaktívne zobrazovanie dát na webovej stránke.

Kľúčovým rozhodnutím pri návrhu tejto práce bolo zvolenie vhodného zdroja informácií o predpovedi počasia. Kvôli detailným a obsiahlym dátam ktoré sú k dispozícii zdarma od Open Weather Map som sa rozhodol využiť tento zdroj pre hlavný graf predpovede počasia. Táto detailná predpoveď počasia je avšak iba na 5 dní dopredu, ale ako základ to stačí. Navyše si z týchto záznamov taktiež ukladám tie staršie a tým si vytváram históriu počasia pre sledované lokality.

Návrh databáze zahŕňa iba tri tabuľky, avšak tento návrh je optimálny. Nevznikajú žiadne duplicitné dáta a k záznamom je velice jednoduchý prístup cez primárny kľúč ktorý sa skladá z ID lokality a časovej známky. Týmto si môžeme jednoducho vybrať záznamy o ktoré máme záujem.

K spracovaniu dát a aktualizácií databáze som vypracoval dva PHP skripty. Tie dokážu stiahnuť dáta o predpovedi počasia od Yahoo! a Open Weather Map, ľahko ich upraviť a správne uložiť do databáze. Momentálne tieto skripty pri spustení sťahujú dáta pre 13 lokalít, ale ako je napísané v kapitole 6, je ich možné rozšíriť.

Grafický návrh webovej stránky som vypracoval v programe Adobe Photoshop a na základe tohoto návrhu som sa snažil stránku implementovať. Takže som k nej vypracoval všetky HTML a CSS súbory. (okrem CSS pre nástroj date range picker)

Na záver bolo potrebné na hlavnom grafe zobrazit' dáta o ktoré ma užívateľ záujem. K tomu som využil JavaScript, pretože pracuje na strane klienta a je možné ho využiť aj po načítaní stránky. Takže na základe vstupu od užívateľa si pri načítaní stránky vyberiem zvolené dáta z databáze a predám ich JavaScriptu. Ten sleduje užívateľové akcie a dokáže prepnúť zobrazené dáta v grafe.

Literatúra

- [1] Web API. 2018, [Online; navštívené 29.04.2018]
URL 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. 2018, [Online; navštívené 28.04.2018]
URL https://en.wikipedia.org/wiki/List_of_Google_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] Earth. 2018, [Online; navštívené 02.05.2018]
URL <https://en.wikipedia.org/wiki/Earth>
- [28] Apache Cordova. 2018, [Online; navštívené 12.05.2018]
URL <https://cordova.apache.org/>

Prílohy

Príloha A

Obsah priloženého CD

- **index.php** - domovská stránka aplikácie
- **app.php** - podstránka pre mobilnú aplikáciu
- **stats.php** - podstránka pre štatistiky o polohe a vzdialenosti
- **kontakt.php** - podstránka pre kontakt
- **main.css** - súbor obsahujúci kaskádové štýly ku všetkým stránkam
- **owm_db_update.php** - aktualizácia databáze podľa dát z Open Weather Map
- **yahoo_db_update.php** - aktualizácia databáze podľa dát z Yahoo!
- **rens_functions.php** - často používané php funkcie
- **pocasio.apk** - inštalačný súbor mobilnej aplikácie
- 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.powerpointintergrund.com cloud-wallpaper-13.jpg** - header
- Mobilna aplikacia
 - **cordova_forecast_app** - zdrojové súbory mobilnej aplikácie
 - **icon1024px.psd** - ikonka pre mobilnú aplikáciu 1024x1024

Príloha B

Náhľad aplikácie (PC)

Domov Weather app Štatistiky Kontakt

Predpoveď počasia.

Predpoveď na 10 dní podľa vašej lokácie, Zlín, 143km

05 Máj	06 Máj	07 Máj	08 Máj	09 Máj	10 Máj	11 Máj	12 Máj	13 Máj	14 Máj
20°/12°	21°/8°	22°/9°	23°/10°	23°/12°	26°/13°	25°/15°	27°/14°	27°/15°	23°/13°

Mesto: Čas:

Vyhľadať

Predpoveď počasia pre Zlín.

Teplota Vietor Zrážky Vlhkosť Oblačnosť

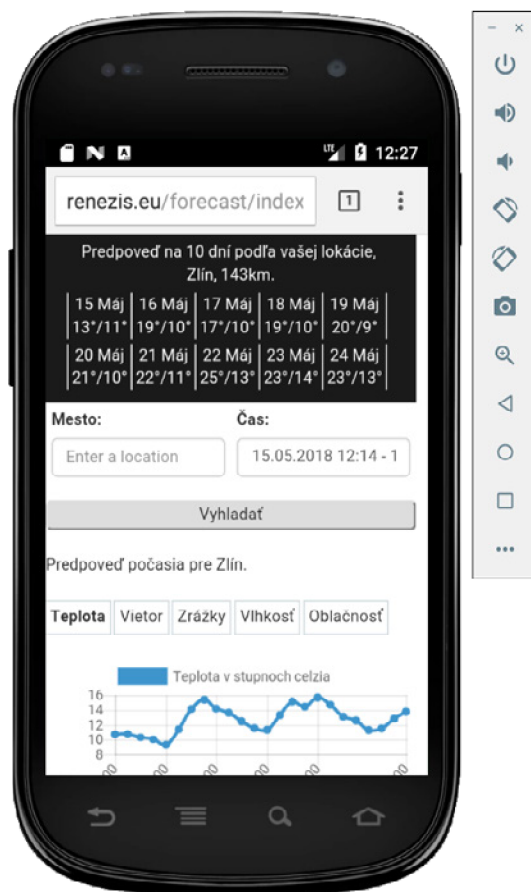
Teplota v stupnoch celzia

Time	Temperature (°C)
05.05.06:00	13.8
05.05.12:00	18.5
05.05.18:00	15.5
06.05.00:00	11.5
06.05.06:00	12.2
06.05.12:00	18.5
06.05.18:00	14.2
07.05.00:00	10.5
07.05.06:00	13.5
07.05.12:00	19.5
07.05.18:00	17.0
08.05.00:00	11.5

Vypracované ako bakalárska práca pre Vysoké Učenie Technické v Brně, Fakulta informačních technologií
2017-2018, Kamil Mucha

Príloha C

Náhľad aplikácie (Mobilné zariadenie - prehliadač)



Príloha D

Náhľad aplikácie (Hybridná Mobiná aplikácia)

