

Stredná priemyselná škola elektrotechnická
Hálova 16, 851 01 Bratislava

KlimaSky

KOMPLEXNÁ ODBORNÁ MATURITNÁ PRÁCA

Stredná priemyselná škola elektrotechnická
Hálova 16, 851 01 Bratislava

KlimaSky

KOMPLEXNÁ ODBORNÁ MATURITNÁ PRÁCA

Študijný odbor: 2573M programovanie digitálnych technológií
Konzultant: Ing. Dominik Zatkalík, PhD.

Bratislava, 2025

4.D

Daniel Kliman

Čestné vyhlásenie

Ja, dolupodpísaný, Daniel Kliman, študent 4. D triedy Strednej priemyselnej školy elektrotechnickej Hálova 16, 851 01 Bratislava, týmto vyhlasujem, že som túto prácu vyhotovil sám, s použitím uvedenej literatúry a podľa rád môjho konzultanta.

.....

V Bratislave, 20.2.2025

Daniel Kliman

Pod'akovanie

Rád by som sa touto cestou poďakoval svojmu školiteľovi za prístup a odborné rady. Tiež by som sa rád poďakoval rodičom ktorý ma podporovali pri realizácii praktickej časti mojej práce a OpenLab za poskytnuté odborné vzdelanie a priestor na vzdelávanie.

Abstrakt

Cieľom projektu bolo vytvoriť modernú a spoľahlivú aplikáciu, ktorá poskytuje aktuálne meteorologické údaje na základe geografickej polohy používateľa.

V úvodnej časti sa práca venuje použitým aplikáciám pri vývoji, teoretickému základu predpovedí počasia, princípom získavania a spracovania meteorologických údajov. Následne opisujeme proces vývoja aplikácie a históriu predpovedania počasia

Výsledná aplikácia umožňuje používateľom získať presné informácie o aktuálnom počasí a krátkodobej predpovedi. Okrem základných meteorologických údajov, ako sú teplota, vlhkosť, vietor a zrážky, ponúka aj prehľadné vizualizácie a jednoduché ovládanie.

Kľúčové slová: predpoveď počasia, meteorológia, API, vývoj aplikácií, softvér

Abstract

The goal of the project was to create a modern and reliable application that provides up-to-date meteorological data based on the user's geographic location.

The initial part of the thesis discusses the applications used in the development, the theoretical basis of weather forecasting, and the principles of meteorological data acquisition and processing. We then describe the application development process and the history of weather forecasting

The resulting application allows users to obtain accurate information about the current weather and short-term forecast. In addition to basic meteorological data such as temperature, humidity, wind and precipitation, it also offers clear visualizations and easy operation.

Keywords: weather forecasting, meteorology, API, application development, software

Obsah

Úvod.....	11
1 História predpovedí počasia	12
1.1 Stredovek a renesancia.....	12
1.2 Technologický pokrok	13
1.3 21. storočie: Presnosť a umelá inteligencia	14
1.4 Kľúčové míľniky v predpovedaní počasia.....	14
1.5 Budúcnosť predpovedí počasia.....	15
2 Pracovné programy.....	16
3 Podobné aplikácie.....	18
3.1 VentuSky	18
3.2 PočasieRadar.....	18
3.3 AccuWeather	19
4 O programe	20
5 Vývoj aplikácie.....	22
5.1 Použité technológie a frameworky	22
5.2 Implementácia API rozhraní.....	24
5.3 Nastavenie vývojového prostredia.....	25
5.3.1 Vytvorenie vlastných pluginov a modelov	25
5.4 Vytváranie HTTP ovládačov	25
5.5 Používatelia a autentifikácia	26
5.6 Technická architektúra backendovej časti aplikácie.....	26
5.6.1 Pluginová architektúra	26

5.6.2	Architektúra OCMS aplikácie	27
6	Frontend.....	28
6.1	UI a UX.....	28
6.2	Technológie a úlohy frontendu	28
7	Backend	30
7.1	Historický vývoj backendu	30
7.2	Kľúčové komponenty backendu	30
7.3	Úlohy backendu	31
7.4	Backend v architektúre aplikácie	31
8	Marketingový plán.....	32
9	Záver.....	34
	Použitá literatúra	35

Zoznam skratiek, značiek a symbolov

API	Rozhranie pre programovanie aplikácií (Application Programming Interface)
CMS	System na správu obsahu (Content Management System)
CSS	Kaskádové štýly (Cascading Style Sheets)
HTML	Značkovací jazyk pre hypertext (HyperText Markup Language)
HTTP	Hypertextový prenosový protokol (Hypertext Transfer Protocol)
JavaScript	Skriptovací jazyk pre tvorbu dynamických komponentov
JSON	Formát zápisu objektov v JavaScripte (JavaScript Object Notation)
PHP	Hypertext Preprocessor
REST	Súbor architektonických obmedzení (Representational State Transfer)
UI	Používateľské rozhranie (User Interface)
UX	Používateľská skúsenosť (User Experience)
WWW	Svetová sieť (World Wide Web)

Zoznam tabuliek, grafov a ilustrácií

Obr. 1 – SWOT analýza

Úvod

Každý človek pozná určitú formu predpovedi počasia, či chcete či nie, stretnete sa s ňou, či už v televíznych prijímačoch na mobilnom telefóne v rádio prijímači, alebo kdekoľvek inde kde čítate alebo pozeráte správy. Počasie je neodmysliteľnou súčasťou nášho života. Stretávame sa s ním dennodenne, či už doma keď sa ochladí a treba začať vykurovať miestnosti kvôli nášmu komfortu alebo keď ideme do práce, školy alebo len tak na prechádzku. Počasie sa neprispôsobuje nám, ale my počasiu. Keď je vonku chladno, oblečieme si na seba hrejivé oblečenie aby nám nebola zima, v teple je to naopak, radšej zvolíme tričko alebo kraťasy. Preto máme aj aplikácie na predpoveď počasia. Otázka znie či sú kvalitné a presné? Samozrejme nikdy sa nedá odhadnúť presne aké počasie bude.

Preto sme vytvorili aplikáciu KlimaSky, kde si viete ľahko a pohodlne pozrieť predpoveď počasia. Cieľom práce bolo naprogramovať backend aplikácie a vytvoriť dizajn. Úlohou aplikácie je jednoducho zobrazit' používateľovi predpoveď počasia.

Veríme, že aplikácia, ktorú sme vytvorili splní svoj význam a uľahčí ľuďom si pozrieť predpoveď počasia dopredu.

1 História predpovedí počasia

Prvé zmienky o odhadoch počasia sa datujú už od starovekého grécka. Avšak nebolo jednoduché presvedčiť ľudí a ani Aristotela, ktorý neveril že vietor je vzduch v pohybe, veril v studené vetry prichádzajúce zo západu. V 17. storočí vynášiel taliansky vedec Evangelista Torricelli ortuťový barometer. K vývoju predpovedí prispeli významné pokusy chemikov v 17. a 18. storočí. Začali sa formulovať prvé zákony o teplote, tlaku a hustote plynov Robertom Boylem a Jacquesom Alexandrom Césarom Charlesom.

Predpovedanie počasia začalo ako schopnosť prežiť, keď sa prví ľudia spoliehali na pozorovanie prírody, aby predvídali zmeny počasia.

- Babylončania (okolo roku 650 pred n. l.): Boli jedni z prvých, ktorí systematicky zaznamenávali vývoj počasia a na predpovede využívali pozorovania mrakov a nebeských javov. Ich metódy boli prepojené s astrológiou, pretože verili, že počasie riadia bohovia.
- Starovekí Gréci (okolo roku 340 pred n. l.): Aristoteles v diele Meteorologica položil základy meteorológie ako vedy. Teoretizoval o kolobehu vody, vetre a poveternostných javoch, hoci mnohé z jeho myšlienok boli špekulatívne.
- Staroveká Čína (okolo roku 300 pred n. l.): Číňania vyvinuli podrobné kalendáre a metódy predpovedania počasia na základe astronomických pozorovaní. Na predpovedanie počasia využívali aj prírodné znaky, ako napríklad správanie zvierat a rastlín.

1.1 Stredovek a renesancia

V tomto období sa predpovede počasia naďalej robili prevažne na základe pozorovania, ale vynájdenie prístrojov znamenalo začiatok vedeckejšieho prístupu.

- Stredoveká Európa: V stredoveku sa hojne používali povesti a príslovia o počasí, ako napríklad: „Červená obloha v noci, pastierova radosť; červená obloha ráno, pastierova výstraha.“ Tieto príslovia vychádzali zo stáročných pozorovaní, ale chýbala im vedecká presnosť.
- Renesancia (15. - 16. storočie): Vynález kľúčových prístrojov spôsobil revolúciu v pozorovaní počasia:
- Teplomér (1593): Galileo Galilei vyvinul skorú verziu teplomera, ktorý umožňoval meranie teploty.

- Barometer (1643): Evangelista Torricelli vynášiel barometer na meranie atmosférického tlaku, ktorý je rozhodujúcim faktorom pri predpovedaní počasia.
- Vlhkomer (15. storočie): Leonardo da Vinci vytvoril prvý vlhkomer na meranie vlhkosti.
- 17. - 19. storočie: Zrod modernej meteorológie

V 17. až 19. storočí sa rozvíjalo systematické pozorovanie počasia a vyvíjali sa nástroje, ktoré položili základy modernej meteorológie.

- 17. storočie: Vedci ako Robert Hooke a Blaise Pascal zdokonalili prístroje a začali zaznamenávať údaje o počasi. Pascal dokázal, že atmosférický tlak klesá s nadmorskou výškou, a spojil zmeny tlaku so zmenami počasia.
- 18. storočie: Benjamin Franklin skúmal búrky a zistil, že sa môžu pohybovať naprieč regiónmi, čím spochybnil názor, že počasie je výlučne lokálne.
- 19. storočie: Telegraf spôsobil revolúciu v predpovedaní počasia tým, že umožnil rýchly zber a výmenu údajov o počasi na veľké vzdialenosti.
- 1854: Balaklavská búrka počas krymskej vojny poukázala na potrebu lepšej predpovede počasia. Francúzsky astronóm Urbain Le Verrier navrhol vytvorenie siete meteorologických staníc, čo viedlo k zriadeniu národných meteorologických služieb.
- 1870: V USA bola založená Národná meteorologická služba, čo znamenalo začiatok organizovanej predpovede počasia v Spojených štátoch.

1.2 Technologický pokrok

20. storočie prinieslo nebývalý pokrok v predpovedaní počasia, ktorý bol spôsobený technologickými inováciami a vedeckými objavmi.

- 20. roky 20. storočia: Nórska Bergenská škola pod vedením Vilhelma Bjerknesa a jeho tímu zaviedla koncept poveternostných frontov a cyklón. Táto „teória polárnych frontov“ sa stala základom modernej meteorológie.
- 40. - 50. roky 20. storočia: Druhá svetová vojna urýchlila pokrok v technológii počasia:
- Radar: Radar, pôvodne vyvinutý na vojenské účely, bol prispôbostený na sledovanie zrážok a búrok.
- Meteorologické balóny: Tieto balóny vybavené prístrojmi poskytovali vertikálne profily atmosféry.

- Počítače: Počítače umožnili numerickú predpoveď počasia (NWP), pri ktorej matematické modely simulovali atmosférické procesy.
- 60. Roky 20. storočia: Vypustenie meteorologických satelitov, ako napríklad TIROS-1 v roku 1960, umožnilo globálne pozorovanie poveternostných systémov. To znamenalo prelom v presnosti predpovedí.
- 70. - 80. roky 20. storočia: Dopplerov radar zlepšil sledovanie nepriaznivého počasia, ako sú tornáda a búrky. Zaviedla sa aj skupinová predpoveď, ktorá využíva viacero modelových behov na zohľadnenie neistoty.

1.3 21. storočie: Presnosť a umelá inteligencia

V 21. storočí sa predpovede počasia stali vysoko sofistikovanými a využívajú pokročilé technológie a analýzu údajov.

- Superpočítače: Moderné predpovede sa spoliehajú na superpočítače, ktoré spúšťajú komplexné modely simulujúce atmosféru, oceány a zemský povrch. Tieto modely dokážu predpovedať počasie na niekoľko dní alebo dokonca týždňov dopredu s pozoruhodnou presnosťou.
- Satelity: Satelity ako GOES (Geostationary Operational Environmental Satellite) a JPSS (Joint Polar Satellite System) NOAA poskytujú v reálnom čase údaje o globálnom počasi vrátane hurikánov, lesných požiarov a klimatických zmien.
- Strojové učenie a umelá inteligencia: Umelá inteligencia sa čoraz viac využíva na analýzu obrovského množstva historických údajov a údajov v reálnom čase, čím sa zvyšuje presnosť predpovedí a umožňuje predpovedať extrémne poveternostné udalosti.
- Občianska veda: Mobilné aplikácie a osobné meteorologické stanice umožňujú jednotlivcom prispievať údajmi, čím sa zvyšuje hustota pozorovaní.

1.4 Kľúčové míľniky v predpovedaní počasia

- 1870: Založenie Národnej meteorologickej služby USA.
- 1922: Lewis Fry Richardson navrhol prvý numerický model predpovede počasia, ktorý však nebol praktický, kým neboli k dispozícii počítače.
- 50. roky 20. storočia: Boli vytvorené prvé úspešné počítačové predpovede počasia.
- 1960: Vypustenie TIROS-1, prvej meteorologickej družice.
- 80. roky: Zavedenie Dopplerovho radaru na sledovanie nepriaznivého počasia.

- 90. roky: Vývoj techník skupinovej predpovede.
- 2000: Modely s vysokým rozlíšením a globálne systémy asimilácie údajov sa stali štandardom.

Kultúrny a sociálny vplyv

Predpovede počasia boli vždy prepojené s ľudskou kultúrou a spoločnosťou:

- Poľnohospodárstvo: Poľnohospodári sa už stáročia spoliehajú na predpovede počasia pri plánovaní výsadby a zberu úrody.
- Navigácia: Námorníci používali modely vetra a poveternostné znamenia na bezpečnú navigáciu.
- Pripravenosť na katastrofy: Presné predpovede zachraňujú životy tým, že poskytujú včasné varovanie pred hurikánmi, tornádami, povodňami a inými prírodnými katastrofami.
- Klimatické zmeny: Moderné predpovede zohrávajú kľúčovú úlohu pri pochopení a zmierňovaní dôsledkov klimatických zmien.

1.5 Budúcnosť predpovedí počasia

Budúcnosť predpovedania počasia spočíva v ešte väčšej presnosti a integrácii nových technológií:

- Kvantová výpočtová technika: Mohli by priniesť revolúciu v modelovaní tým, že by riešili zložité rovnice rýchlejšie ako kedykoľvek predtým.
- Zdokonalené satelitné siete: Satelity novej generácie poskytnú ešte podrobnejšie pozorovania.
- Modely riadené umelou inteligenciou: Strojové učenie bude naďalej zvyšovať presnosť predpovedí, najmä v prípade extrémnych poveternostných udalostí.
- Predpovedanie klímy: Dlhodobé klimatické modely pomôžu spoločnostiam prispôbiť sa meniacemu sa počasiu.

Predpovedanie počasia prešlo dlhú cestu od svojich dávnych koreňov a vyvinulo sa do kritickej vedy, ktorá ovplyvňuje takmer každý aspekt moderného života. Jej história je svedectvom ľudskej vynaliezavosti a našej neustálej snahy pochopiť a predpovedať svet prírody.

2 Pracovné programy

Každý programátorský projekt vyžaduje určité programy na kvalitný a pohodlnejší vývoj. Niektorí si možno myslia že si len píšeme nezmyselné riadky do počítača, no tí, ktorí vedia čo obnáša programovanie, vedia že to nie je len tak si niečo písať. Každý programátor má svoj vlastný typ písania kódu, každý kód je niečím výnimočný. Preto sme pre Vás zhrnuli a stručne opísali programy ktoré sme použili pri vývoji KlimaSky.

Visual Studio Code

Jeden z najznámejších editorov zdrojového kódu, od spoločnosti Microsoft. Tento editor podporuje širokú škálu programovacích jazykov ako sú Python, JavaScript, TypeScript, C++, Java a mnoho ďalších. Tento editor používajú začiatčníci ale aj pokročilí dokonca aj profesionáli. Má veľkú výhodu že si môžete pridať ďalšie programovacie jazyky ktoré nie sú zahrnuté hneď po stiahnutí. Pomocou ktorého sme robili Frontend. Tento program je úplne zadarmo [14].

PhpStorm

Ďalší editor zdrojového kódu, ktorý je primárne zameraný na PHP. Takisto podporuje aj iné programovacie jazyky ktoré sa dajú pridať. Tento editor je vytvorený našimi susedmi z Čiech spoločnosťou JetBrains s.r.o. Zatiaľ čo Visual studio code je úplne zadarmo, to sa nedá povedať pri PhpStorm, ktorý je platený. Čo je však veľkou výhodou, študenti si môžu požiadať o študentské predplatné, ktoré je zadarmo ak ste držiteľom študentského preukazu ISIC a máte platný čip na karte ISIC [15].

Figma

Vývojové prostredie ktoré hlavne slúži na vytvorenie WireFrameov a dizajnu bez ktorých sa kvalitné aplikácie nezaobídu. Takisto sa tu dajú vytvárať aj prototypy aplikácií, prezentácie, logá a mnoho ďalšieho. My sme tu vytvorili WireFramy a dizajn aplikácie, podľa ktorého sa následne naprogramoval frontend [16].

Google Sheets

Tabuľky od spoločnosti Microsoft. Excel vo webovej podobe pre jednoduchšie projekty. Možnosť zdieľaného prístupu pre viac používateľov. Dá sa používať aj na mobile po stiahnutí aplikácie Google Sheets. Tu sme si zadefinovali ktorý týždeň budeme na čom pracovať. Netreba platiť Microsoft Office balík.

Google Docs

Dokumenty od spoločnosti Microsoft. Word vo webovej podobe. Možnosť zdieľaného prístupu pre viac používateľov. Netreba platiť Microsoft Office balík. Dá sa používať aj na mobile po stiahnutí aplikácie Google Docs. V Google Docs sme písali funkčnú špecifikáciu, nápady ktoré sme ďalej konzultovali.

Notion

Aplikácia slúži na jednoduché rozdelenie a organizovanie úloh pre jednotlivých členov tímu. Jednoduché organizovanie úloh pomocou drag & drop funkcie. Podpora integrovanej umelej inteligencie. Zdieľanie kalendáru do Apple Calendar a Microsoft Kalendáru. Podpora tmavého a svetlého režimu synchronizovaná so systémovými nastaveniami. Možnosť písania dokumentov [17].

3 Podobné aplikácie

3.1 VentuSky

Aplikácia je založená na mapovo zobrazení. V základnom balíku nájdeme zrážky, ktoré predpovedajú po hodine, teplotu, pocitovú teplotu, satelit, nárazy vetra, kvalitu ovzdušia, polárnu žiaru. V premium balíku nájdeme zrážky, ktoré predpovedajú po 15-tich minútach, rýchlosť vetra, oblačnosť, tlak vzduchu, more, búrky, vlhkosť, rosný bod, snehová pokrývka, nulová izoterma, viditeľnosť. Tento balík je limitovaný na 4 zariadenia, je tu možnosť platiť mesačne 2,99 eur alebo ročne 14,99 eur. Máme tu možnosť predplatného Premium +, ktorý je nacený na 3,99 eur mesačne alebo 19,99 eur ročne, obsahuje všetko z premium balíka, má možnosť až 25 zariadení, prioritnú podporu v prípade akéhokoľvek problému, zobrazenie neobmedzených vrstiev, pokročilé widgety a zasielanie všetkých upozornení ktoré sa dajú nastaviť v nastaveniach. V nastaveniach ďalej nájdeme nastavenie jednotiek, animácie vetra, celkové nastavenie aplikácie, notifikácie, podpora, model a používateľský účet

3.2 PočasieRadar

Zobrazenie počasia na dva dni dopredu, aktuálna lokácia a zobrazenie predpovedi, meteoradar, ktorý zobrazuje aktuálnu predpoveď, predpoveď na aktuálny deň a na nasledujúci deň. Predpoveď na 10 dní dopredu a aj 14 dňovú. Astrologické informácie kde sa nachádzajú informácie o východe a západe slnka a takisto aj východ a západ mesiaca. U index na aktuálny deň. Namerané hodnoty v regionálnych meteorologických staniciach. Stav kvality ovzdušia. Meteorologické výstrahy pred búrkami, dažďom, víchricou a poľadovicou. Teplotný radar, ktorý zobrazuje aktuálnu predpoveď, na aktuálny deň a ďalší deň pomocou mapy. Dažďový radar, ktorý zobrazuje aktuálnu predpoveď, na aktuálny deň a ďalší deň pomocou mapy. Možnosť platby predplatného na odstránenie bannerových, celo obrazových a videových reklám za 1,49 eur mesačne alebo 9,99 eur ročne.

3.3 AccuWeather

Internetový portál na predpoveď počasia.

Zobrazenie aktuálnej predpovedi počasia, teploty, pocitovej teploty, rýchlosti vetra v km/h, rýchlosti nárazu vetra v km/h, kvality ovzdušia, možnosť zobrazenia podrobného počasia. Podrobná predpoveď nám navyše ukazuje UV index, tlak, percentuálnu oblačnosť, percentuálnu vlhkosť, výška oblakov v metroch, dážď a zrážky v milimetroch, v prípade výstrah sa pri vrchu objaví varovanie pred výstrahami. Predpoveď na aktuálny deň s rovnakými ukazovateľmi. Predpoveď na najbližšiu noc s rovnakými ukazovateľmi. Informácie o východe a západe slnka a mesiaca, doba trvania svetla a tmy. História teplôt, predpovedaných aktuálnych a minuloročných, najvyššia a najnižšia teplota. Po návrate na hlavnú kartu vidíme pod predpoveďou mapu s ukazovateľom pohybu oblakov a ukazovateľ teploty. Nižšie sa nachádza hodinová predpoveď na 12 hodín dopredu, kde je predpovedaná teplota a percentuálna šanca zrážok. Nižšie sa nachádza denná predpoveď na 10 dní dopredu. Nižšie sa nachádzajú informácie o východe a západe slnka. Nižšie sa nachádzajú informácie o kvalite ovzdušia. Na konci sa nachádzajú informácie o výskyte najčastejších alergií. Na vrchu sa nachádza lišta s predpoveďou na dnes, každú hodinu, dennú predpoveď, radar zobrazený pomocou interaktívnej mapy. Minútová predpoveď ktorá simuluje pohyb oblakov po 5 minútach. Mesačná predpoveď na 3 mesiace dopredu. Kvalita ovzdušia zobrazená pomocou grafu a interaktívnej mapy. Posledná možnosť nám zobrazí zdravie a aktivity a počasné podmienky na vykonávanie určitých aktivít. Výstrahy pred najčastejšími ochoreniami v aktuálnom období, výstrahy pred alergiami, podmienky na vonkajšie aktivity, podmienky na starostlivosť o záhradu, podmienky na cestovanie, informácie o výskyte hmyzu.

4 O programe

Aplikácia KlimaSky ponúka používateľom intuitívne a priateľské rozhranie na zobrazenie aktuálneho počasia a predpovedí pre rôzne mestá. Po spustení aplikácie sa používateľ nachádza na domovskej karte, ktorá obsahuje tlačidlo **Login**. Tento prvok umožňuje prístup do aplikácie, pokiaľ už má používateľ vytvorený účet. Ak účet ešte nemáte, môžete sa zaregistrovať prostredníctvom tlačidla **Register**, ktoré vás presmeruje na registračný formulár, kde si vytvoríte nový účet.

Po prihlásení

Po úspešnom prihlásení budete presmerovaný na hlavnú obrazovku s názvom **Počasie**. Na tejto karte sa zobrazuje aktuálna predpoveď počasia pre vašu lokalitu, vrátane aktuálnej teploty, pocitovej teploty, rýchlosti vetra a množstva zrážok.

Funkcie hlavnej obrazovky:

Pridať mesto: Pomocou tlačidla + si môžete pridať ďalšie mesto do zoznamu a pozrieť si predpoveď počasia pre konkrétnu lokalitu.

7-dňová predpoveď: Na karte sa nachádza aj predpoveď počasia na nasledujúcich 7 dní.

Navigačná lišta: Na spodnej strane obrazovky je lišta s ponukou na zobrazenie ďalších kariet a nastavení.

Nastavenia aplikácie

Ak chcete upraviť svoje osobné údaje alebo nastavenia aplikácie, môžete kliknúť na tlačidlo

Settings na lište. Otvorí sa vám rozbaľovacia ponuka, ktorá ponúka tri možnosti:

Osobné informácie: Tu si môžete zmeniť svoje používateľské meno a heslo.

Spravovanie predplatného: Umožňuje vám zrušiť svoje predplatné.

Zmena jednotiek: Môžete prepínať medzi metrickými a imperiálnymi jednotkami pre teplotu, vzdialenosti, rýchlosť vetra a iné parametre.

Implementácia API

Aplikácia čerpá dáta o počasí prostredníctvom API z **OpenWeather** portálu, ktorý zabezpečuje aktuálnu a presnú predpoveď. Tento prístup umožňuje aplikácii získavať dáta v reálnom čase, čo poskytuje používateľom relevantné informácie.

Multiplatformová dostupnosť

Aplikácia je navrhnutá ako multiplatformové riešenie, čo znamená, že je dostupná na rôznych zariadeniach a platformách. Cieľom je poskytovať používateľom flexibilitu pri sledovaní predpovedí počasia na platforme, ktorá im vyhovuje najviac.

Tento prístup zabezpečuje, že aplikácia bude užitočná pre široké spektrum používateľov, ktorí môžu používať rôzne zariadenia a operačné systémy.

5 Vývoj aplikácie

5.1 Použité technológie a frameworky

Pri vývoji backendovej časti aplikácie klimaSky sme využil **OCMS** (October CMS), ktorý je postavený na **frameworku Laravel**, v kombinácii s **relačnou databázou MySQL**. Programovanie v prostredí OCMS je intuitívne a efektívne, keďže poskytuje širokú škálu vopred pripravených modulov, ktoré sa dajú ľahko integrovať do projektu.

Pri vývoji front-endovej časti našej aplikácie KlimaSky sme použili aplikáciu **Framer**, čo je skvelý nástroj na vytváranie moderných, interaktívnych a responzívnych aplikácií. Framer nám poskytuje flexibilitu pri dizajne a vývoji našej aplikácie, a tým umožňuje rýchly prototyping a implementáciu dynamických komponentov. Tu je podrobný rozpis, ako nám Framer pomohol pri vývoji našej aplikácie:

Návrh používateľského rozhrania (UI)

Vďaka Frameru sme vytvorili vizuálne atraktívne a intuitívne používateľské rozhranie, čo je pre aplikáciu zameranú na predpoveď počasia veľmi dôležité.

Dizajnovanie obrazoviek: Navrhli sme jednotlivé obrazovky našej aplikácie, ako napríklad domovskú stránku, login kartu, hlavnú obrazovku s počasím a obrazovku nastavení. Vďaka drag-and-drop rozhraniu Frameru sme mohli jednoducho umiestniť prvky ako tlačidlá, textové polia a obrázky.

Interaktívne komponenty: Vytvorili sme interaktívne prvky, ktoré zahŕňajú animácie pri kliknutí na tlačidlá, zobrazenie predpovedí počasia po výbere mesta a dynamické zobrazenie údajov načítaných zo servera (ako teplota, vietor, zrážky).

Responzívny dizajn

Keďže naša aplikácia KlimaSky je multiplatformová, musela byť optimalizovaná pre rôzne zariadenia, ako sú smartfóny, tablety a počítače. Framer nám umožnil ľahko vytvoriť responzívne rozhranie, ktoré sa prispôsobí rôznym veľkostiam obrazoviek.

Flexibilné rozvrhnutie: Nastavili sme komponenty tak, aby sa automaticky prispôsobili rozmerom obrazovky.

Testovanie na rôznych zariadeniach: V Frameri sme mohli testovať náš návrh na simulovaných zariadeniach, čo nám pomohlo zabezpečiť, že aplikácia bude vyzerat' a fungovať skvele na akomkoľvek zariadení.

Integrácia s API a dynamickým obsahom

Framer nám umožnil jednoduchú integráciu s externými API, ako je OpenWeather API, ktoré naša aplikácia využíva na získavanie údajov o počasí.

Dynamické zobrazenie dát: Vytvorili sme dynamické komponenty, ktoré načítavajú dáta z OpenWeather API a zobrazujú aktuálne počasie, predpovede a ďalšie informácie v reálnom čase.

Interakcia s používateľom: Keď používateľ zvolí mesto alebo upraví nastavenia, Framer nám umožní prispôbiť zobrazené údaje podľa jeho volieb, čím môžeme zlepšiť používateľský zážitok.

Prepojenie s backendom a správu stavu aplikácie

Integráciu Frameru s backendovými systémami a správou stavu aplikácie sme dosiahli pomocou October CMS. Pre našu aplikáciu KlimaSky bola dôležitá správa stavu, najmä pre uchovávanie nastavení používateľa, zoznamu miest a údajov o používateľoch.

Synchronizácia s backendom: Pri prihlásení alebo registrácii používateľa sme synchronizovali údaje s backendom, aby sme zabezpečili konzistentnú používateľskú skúsenosť naprieč zariadeniami.

Testovanie a prototypovanie

Framer nám poskytol možnosti testovania a prototypovania, čo bolo veľmi výhodné pre návrh aplikácie ešte pred jeho plným vývojom.

Rýchle prototypovanie: Vďaka Frameru sme mohli vytvoriť interaktívne prototypy, ktoré simulovali fungovanie aplikácie v reálnom prostredí.

Testovanie používateľského rozhrania: Testovali sme naše návrhy a zbierali spätnú väzbu od používateľov, čo nám pomohlo optimalizovať používateľský zážitok a identifikovať prípadné problémy ešte pred finálnym nasadením aplikácie.

Výhody použitia Frameru pre nás

Rýchly vývoj: Framer nám umožnil rýchlo vytvoriť prototypy a priamo nasadiť interaktívne rozhranie.

Škálovateľnosť: Vďaka flexibilitě Frameru sme boli schopní škálovať našu aplikáciu pre rôzne platformy a zariadenia.

Zjednodušené testovanie: Možnosť testovať návrhy v reálnom čase nám umožnilo rýchlo sa prispôbiť potrebám používateľov a získať potrebnú spätnú väzbu.

Použitím Frameru sme vytvorili vizuálne interaktívny a funkčný front-end, ktorý umožňuje našim používateľom jednoducho a efektívne získať predpoveď počasia na akomkoľvek zariadení.

5.2 Implementácia API rozhraní

Jadrom aplikácie je **prehľadná a bohatá databáza miest**. V rámci backendu sme vytvoril **API rozhranie**, ktoré poskytuje dáta vo formáte **JSON** pre frontend aplikácie. Toto API umožňuje **načítanie kompletného zoznamu miest**, ale zároveň aj **pokročilé filtrovanie a vyhľadávanie** na základe parametrov, ako sú: názvu, krajiny.

Každý používateľ má **vlastné API**, ktoré poskytuje **podrobné informácie** o danom používateľovi vrátane: veku, podrobnosti o predplatnom, lokácií.

Používateľ si môže mesto **označiť ako obľúbené** a uložiť si ho pre rýchlejšie a jednoduchšie zobrazenie.

Všetky operácie sú dostupné prostredníctvom **POST requestov** a vyžadujú **validáciu vstupných údajov**, aby sa zabezpečila **konzistencia dát v databáze**.

Súčasťou backendu je aj **komplexný systém používateľov**, ktorý zahŕňa:

- registráciu a prihlasovanie,
- odhlasovanie a obnovu hesla,
- zobrazovanie profilu používateľa.

Použili sme **JWT autentifikáciu**, ktorú sme integrovali cez **vlastný plugin v OCMS**. Zabezpečenie API sme riešili pomocou **middleware-u**, ktorý overuje prihlásenie a oprávnenia používateľa.

5.3 Nastavenie vývojového prostredia

Viacerí z nás používame **MacBook** s **vývojovým prostredím MAMP**. Tento softvér poskytuje **lokálny webový server Apache, MySQL databázu a PHP**. Pre tento projekt sme použil **PHP 8.2**, ktoré je plne kompatibilné s **OCMS 3**.

Kroky prípravy vývojového prostredia:

- **Spustenie MAMP** a prihlásenie do **phpMyAdmin**.
- **Vytvorenie MySQL databázy** `klimasky_local`
- **Konfigurácia pripojenia** v `.env` súbore.
- **Vytvorenie OCMS projektu** cez Composer: `composer create-project october/october KlimaSky`
- **Spustenie migrácií databázy**: `php artisan october:up`

5.3.1 Vytvorenie vlastných pluginov a modelov

V OCMS sú **pluginy kľúčovým prvkom flexibility**. Vytvorili sme niekoľko pluginov, napríklad **AppUser** pre správu používateľov a **AppLocation** pre správu miest a lokácií. Pluginy sa generujú príkazom:

- `php artisan create:plugin Namespace.Name`
Podobne, modely sa vytvárajú pomocou:
- `php artisan create:model Namespace.Name ModelName`

Model **Location** napríklad reprezentuje mesto a lokáciu, ktorý obsahuje **atribúty (title, description, longitude, latitude)**, **relácie (User)** a **metódy na manipuláciu s dátami**.

Backend aplikácie KlimaSky je navrhnutý tak, aby bol **modulárny a efektívny**.

5.4 Vytváranie HTTP ovládačov

Kontrolery, známe aj ako ovládače, zohrávajú kľúčovú úlohu v projekte klimaSky, pretože ich hlavnou úlohou je spracovanie požiadaviek prichádzajúcich z frontendovej časti aplikácie na backend. Ovládače slúžia na efektívne riadenie a spracovanie všetkých akcií, ktoré vznikajú pri volaní jednotlivých endpointov. Každý endpoint predstavuje konkrétnu operáciu, ktorá môže byť vykonávaná prostredníctvom HTTP metód.

Požiadavky, ktoré ovládače spracovávajú, sa delia na dva hlavné typy: GET a POST. GET požiadavky sú určené na získavanie dát, napríklad načítanie detailných informácií o konkrétnom inzeráte alebo výpis zoznamu dostupných vozidiel.

Naopak, POST požiadavky slúžia na vykonávanie zložitejších operácií, akými sú pridávanie nových inzerátov, registrácia nových používateľov alebo iné manipulácie s databázou. POST požiadavky si preto vyžadujú dôslednú validáciu vstupných dát a overenie oprávnení používateľa.

5.5 Používatelia a autentifikácia

Na zabezpečenie autentifikácie používateľov sme sa rozhodli použiť `tymon/jwt-auth`, ktorú sme implementovali prostredníctvom vlastného pluginu `AppUser.UserApi`. Tento plugin zahŕňa kompletnú autentifikačnú logiku aplikácie, čo znamená, že obsahuje aj HTTP kontrolery na vykonávanie operácií, ako je prihlásenie a registrácia nových používateľov. Okrem toho obsahuje aj middleware `Authenticate`, ktorý zabezpečuje, že k určitým endpointom majú prístup iba autentifikovaní používatelia.

5.6 Technická architektúra backendovej časti aplikácie

Počas vývoja aplikácie sme kládli veľký dôraz na navrhnutie dobre premyslenej technickej architektúry, ktorá zabezpečí jej jednoduchú správu, rozšíriteľnosť a udržateľnosť. Tento prístup sa ukázal ako kľúčový najmä v situáciách, keď bolo potrebné implementovať nové požiadavky alebo vykonať rozsiahlejšie zmeny v systéme.

Architektúra backendu nebola statická, ale dynamicky sa prispôbovala meniacim sa požiadavkám projektu. Od počiatočných fáz vývoja sme si uvedomovali, že kvalitne navrhnutá technická architektúra je nevyhnutná na zaistenie dobrého škálovania aplikácie, prehľadnosti zdrojového kódu a efektívnejšej spolupráce pri budúcom rozširovaní aplikácie.

5.6.1 Pluginová architektúra

Na začiatku projektu sme si pripravili pluginovú architektúru, ktorá bola najskôr vizualizovaná v Google Sheets. Tento dokument nám pomohol lepšie pochopiť celé riešenie a slúžil aj na vopred definované rozdelenie funkcionalít do samostatných modulov. Výhodou tohto prístupu je, že každý plugin zabezpečuje špecifickú funkcionalitu a komunikuje s ostatnými modulmi prostredníctvom presne definovaných API rozhraní.

Pod pluginovou architektúrou rozumieme koncept, v ktorom sme jednotlivé časti aplikácie rozdelili do menších, izolovaných modulov. Každý plugin je nezávislý a

zabezpečuje konkrétnu oblasť funkcionality. Táto architektúra zjednodušuje rozširovanie aplikácie a vývojárom poskytuje flexibilitu pri práci na jednotlivých komponentoch.

5.6.2 Architektúra OCMS aplikácie

Keď používateľ odošle požiadavku prostredníctvom webového prehliadača alebo API klienta, prebieha v aplikácii nasledujúci proces:

Prijatie požiadavky – HTTP požiadavka je zachytená hlavným vstupným bodom aplikácie, ktorým je súbor `index.php` v root adresári OCMS aplikácie. Tento súbor inicializuje systémové jadro, zavádza potrebné komponenty a konfiguráciu.

- **Routing** – Požiadavka je presmerovaná na správny route, ktorý je definovaný v routing súboroch v jednotlivých pluginoch.
- **Middleware** – Po spracovaní routingu požiadavka prechádza cez middleware, kde prebieha autentifikácia, validácia a iné procesy.
- **Kontroler** – Po spracovaní middleware sa požiadavka dostane k príslušnému kontroleru, kde sa vykoná potrebná logika.
- **Odpoveď** – Nakoniec sa vygeneruje odpoveď, ktorá je zaslaná používateľovi vo formáte JSON.

6 Frontend

Predstavuje časť aplikácie, ktorú vidí a s ktorou interaguje používateľ. Jedná sa o grafické zobrazenie dát tak, aby používateľ mohol s nimi jednoducho pracovať.

Frontend úzko spolupracuje s backendom, aby spoločne vytvorili plnohodnotnú aplikáciu alebo web. Dáta generované na backende sú odoslané na frontend, kde sa zobrazujú používateľovi. Z tohto dôvodu sa frontend často označuje ako prezentačná vrstva, zatiaľ čo backend je známy ako vrstva pre prístup k dátam. Vo vývojových tímoch sú tieto časti často rozdelené, avšak niektorí programátori sa venujú vývoju oboch, čo je známe ako **full-stack development**.

Pretože frontend tvorí vizuálnu a interaktívnu časť aplikácie, je nevyhnutné, aby bolo rozhranie prehľadné a estetické. Tieto aspekty majú na starosti UI a UX dizajnéri.

6.1 UI a UX

UI (User Interface) – Používateľské rozhranie zahŕňa vizuálny dizajn aplikácie alebo webovej stránky, rozloženie prvkov, ich štýl a interakcie medzi prvkami a používateľom.

UX (User Experience) – Používateľská skúsenosť sa zameriava na to, aké pocity a skúsenosti má používateľ pri používaní aplikácie. Cieľom UX je zabezpečiť, aby bola práca s aplikáciou jednoduchá, rýchla a bezproblémová.

Frontend programátori pretvárajú návrhy UI/UX dizajnérov na funkčné rozhrania. Na tento účel používajú rôzne technológie a nástroje.

6.2 Technológie a úlohy frontendu

Frontend zahŕňa viaceré základné technológie:

HTML (HyperText Markup Language)

HTML je základný jazyk na tvorbu webových stránok. Definuje štruktúru a obsah stránky, ako sú nadpisy, odseky, tabuľky či formuláre. V histórii bolo HTML používané ako protokol vizuálneho štýlu pre tlačiarne.

CSS (Cascading Style Sheets)

CSS slúži na definovanie vzhľadu a štýlu webových stránok. Pomocou CSS je možné upraviť farby, typ písma, rozloženie či responzivitu stránky.

JavaScript

JavaScript je programovací jazyk, ktorý pridáva stránkam interaktivitu a dynamiku. S

jeho pomocou sa vytvárajú animácie, validujú formuláre alebo obnovuje obsah bez potreby načítania celej stránky (napríklad pomocou AJAX).

Frameworky a knižnice

Populárne nástroje ako **React**, **Vue.js**, **Qwik**, **Astro** a ďalšie frameworky zjednodušujú vývoj frontendu. Tieto frameworky ponúkajú hotové komponenty a abstrakcie, ktoré urýchľujú vývoj a uľahčujú správu kódu.

Frontend vývojári sa starajú o rôzne úlohy:

- Zobrazenie dát a obsahu na obrazovke používateľa.
- Interakcia s používateľom cez formuláre a ovládacie prvky.
- Správa a manipulácia dát na strane klienta.
- Reakcia na interakcie používateľa (napr. kliknutia alebo zadanie údajov).
- Optimalizácia výkonu a zabezpečenie používateľského rozhrania.

7 Backend

Backend označuje časť aplikácie, ktorá nie je prístupná používateľom. Ide o serverovú stranu, ktorá je zodpovedná za väčšinu funkcionality aplikácie, ako aj spracovanie webových požiadaviek, práca s dátami a ich ukladanie v databáze.

7.1 Historický vývoj backendu

Backend označuje časť aplikácie, ktorá nie je prístupná používateľom. Ide o serverovú stranu, ktorá zabezpečuje väčšinu funkcionality aplikácie, ako je spracovanie webových požiadaviek, práca s dátami a ich ukladanie v databáze. V začiatkoch webových aplikácií pozostával backend z jednoduchých skriptov, ktoré sa vykonávali na serveri a vracali dáta frontendu. S postupným rozvojom webových technológií sa backendové aplikácie stali sofistikovanejšími a využívajú moderné frameworky na efektívne spracovanie požiadaviek a dynamické generovanie obsahu.

7.2 Kľúčové komponenty backendu

Backend zahŕňa viacero základných súčastí:

Server

Je systém, ktorý prijíma požiadavky od klientov a poskytuje odpovede. Na serveri beží aplikačná logika, ktorá spracováva požiadavky a manipuluje s dátami.

Programovací jazyk

Backend sa implementuje pomocou rôznych programovacích jazykov, ako sú **Java**, **Python**, **Ruby**, **PHP**, **C#** a ďalšie. Tieto jazyky umožňujú vývojárom písať logiku aplikácie a spravovať backend.

Webový server

Webový server, ako je **Apache**, **Nginx** alebo **IIS**, prijíma HTTP požiadavky od klientov, spracováva ich a odosiela odpovede. Je zodpovedný za správne smerovanie požiadaviek na aplikáciu.

Databázový systém

Backend komunikuje s databázou, kde sú uložené dáta aplikácie. Databázy môžu byť:

Relačné (napríklad **MySQL**, **PostgreSQL**, **SQLite**), kde sa dáta ukladajú v tabuľkách.

NoSQL (napríklad **MongoDB**, **Redis**, **Cassandra**), kde sú dáta ukladané v nerelačných formátoch, napríklad ako dokumenty alebo hodnoty.

7.3 Úlohy backendu

Backend je zodpovedný za viacero dôležitých úloh, ako sú spracovanie požiadaviek – prijímanie a odpovedanie na požiadavky od klienta, práca s dátami – ukladanie, aktualizácia a vyhľadávanie údajov v databázach, aplikačná logika – výpočet a spracovanie údajov podľa biznis pravidiel, autentifikácia a autorizácia – zabezpečenie prístupu k citlivým údajom, ochrana pred bezpečnostnými hrozbami, optimalizácia výkonu servera a poskytovanie API pre integrácie s inými systémami a službami.

7.4 Backend v architektúre aplikácie

Backend a frontend tvoria spolu plnohodnotnú aplikáciu. Frontend zabezpečuje zobrazovanie dát a interakciu s používateľom, zatiaľ čo backend sa stará o spracovanie dát a obchodnú logiku. Ich spolupráca je kľúčová pre fungovanie aplikácie – frontend odosiela požiadavky na backend, ktorý ich spracuje a vráti odpoveď. Na komunikáciu medzi frontendom a backendom sa často využívajú REST API alebo GraphQL, ktoré umožňujú efektívnu výmenu dát. Moderné architektúry, ako napríklad microservices alebo serverless computing, umožňujú lepšiu škálovateľnosť a flexibilitu backendových systémov.

8 Marketingový plán

FIRMA Daniel Kliman

ROK 2025

ZHRNUTIE PRODUKTOV A SLUŽIEB vývoj hybridných aplikácií KlimaSky, aplikácia na jednoduché a pohodlné zobrazenie počasia

VÍZIA zviditeľniť sa, pracovať na väčších projektoch, rozšíriť sa do iných krajín.

POSLANIE Sme firma, ktorá sa venuje vývojom hybridných aplikácií zatiaľ len na území SR, momentálne vyvíjame aplikáciu na jednoduché a pohodlné zobrazenie počasia KlimaSky

HLAVNÝ CIEĽ MARKETINGOVÉHO PLÁNU Uvedenie a zviditeľnenie aplikácie KlimaSky na trhu

PRÍLEŽITOSTI Dostali sme sa na slovakiaTech konferenciu.

HROZBY máme veľa konkurencií ktoré sú zabehnuté a dennodenne používané ľuďmi po celom svete, no napriek tomu sa nevzdávame a vyvíjame KlimaSky

CIEĽOVÁ SKUPINA

1. Bežní používatelia

Charakteristika: Ľudia, ktorí si chcú skontrolovať každodenné počasia pre plánovanie svojich aktivít.

Demografia: Široké vekové rozpätie (18–65 rokov), muži aj ženy.

Použitie: Denná predpoveď na cesty, obliekanie, alebo voľnočasové aktivity.

Príklad: Pracujúci ľudia, rodičia plánujúci aktivity pre deti, študenti.

2. Outdooroví nadšenci

Charakteristika: Milovníci outdoorových aktivít ako turistika, bicyklovanie, lyžovanie, plachtenie, alebo kempovanie.

Demografia: Vek 20–45 rokov, aktívni a dobrodružní.

Použitie: Detailné predpovede pre presnú lokalitu, radarové dáta, zrážky, veterné podmienky.

Príklad: Turisti, športovci, cestovatelia.

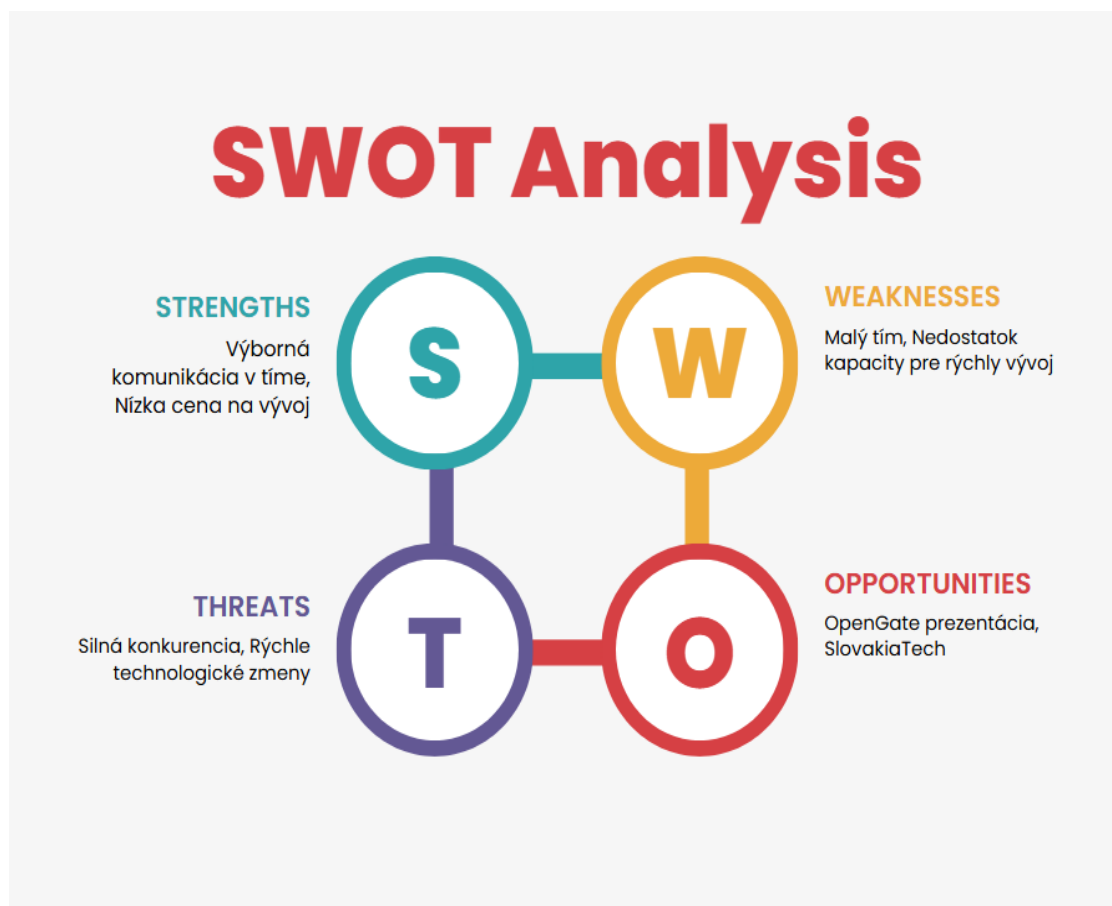
3. Poľnohospodári a pracovníci v poľnohospodárstve

Charakteristika: Ľudia závislí od presných predpovedí počasia pre správne načasovanie práce.

Demografia: Dospelí, prevažne vo vidieckych oblastiach.

Použitie: Zrážky, teplotné výkyvy, údaje o vetre a sezónne predpovede.

Príklad: Farmári, pestovatelia, záhradkári.



Obr. 1 – SWOT analýza

9 Záver

Cieľom tejto maturitnej práce bolo navrhnúť a vytvoriť aplikáciu na predpoveď počasia, ktorá poskytne používateľom presné a aktuálne informácie o meteorologických podmienkach. Počas realizácie sme analyzovali dostupné API služby, zvolili vhodné technológie a implementovali riešenie s dôrazom na používateľskú skúsenosť a funkcionality.

Výsledná aplikácia umožňuje používateľom získať predpoveď počasia pre konkrétnu lokalitu, zobrazovať údaje ako teplota, vlhkosť, rýchlosť vetra či pravdepodobnosť zrážok. Okrem základných funkcií sme sa zamerali aj na prehľadné grafické rozhranie a optimalizáciu výkonu.

Počas vývoja sme čelili viacerým výzvam, najmä pri integrácii externých dátových zdrojov a spracovaní veľkého objemu informácií v reálnom čase. Tieto problémy sme však úspešne vyriešili a aplikácia dosahuje požadovanú funkcionality.

Projekt potvrdil význam moderných technológií pri spracovaní a vizualizácii meteorologických údajov. Do budúcnosti by bolo možné rozšíriť aplikáciu o ďalšie funkcie, ako sú historické dáta, upozornenia na extrémne počasie alebo personalizované predpovede na základe preferencií používateľov.

Táto práca nám poskytla cenné skúsenosti v oblasti programovania, práce s API a vývoja softvérových aplikácií.

Použitá literatura

- [1] CAHIR, J. J., 2025: History of weather forecasting. [online]. [cit. 2025-01-03]. Dostupné na: <<https://www.britannica.com/science/weather-forecasting/History-of-weather-forecasting>>
- [2] BOBKOV, A., 2025: October CMS [online]. [cit. 2025-01-04]. Dostupné na: <<https://docs.octobercms.com/3.x/setup/installation.html>>
- [3] LERDORF, R., 2025: PHP [online]. [cit. 2024-12-04]. Dostupné na: <<https://www.php.net/manual/en/>>
- [4] YOU, E., 2025: Vue.js [online]. [cit. 2025-01-04]. Dostupné na: <<https://vuejs.org/guide/introduction.html>>
- [5] STAFF, C., 2023: What is a Wireframe? [online]. [cit. 2025-01-05]. Dostupné na: <https://www.coursera.org/articles/wireframe?utm_medium=sem&utm_source=gg&utm_campaign=b2c_emea_x_multi_ftcof_career-academy_cx_dr_bau_gg_pmax_gc_sl_en_m_hyb_23-12_x&campaignid=20858198824&adgroupid=&device=c&keyword=&matchtype=&network=x&devicemodel=&creativeid=&assetgroupid=6490027433&targetid=&extensionid=&placement=&gad_source=1&gclid=CjwKCAiAkc28BhB0EiwAM001TTuT1R99vSaso9RoevuaMJLhFB4PwIdef_0QzyZkHo2792FKIyVMMxoCQ00QAvD_BwE>
- [6] MDN, C., 2024: CSS: Cascading Style Sheets [online]. [cit. 2025-01-05]. Dostupné na: <<https://developer.mozilla.org/en-US/docs/Web/CSS>>
- [7] TYMON, S., 2025: *JSON Web Token Authentication for Laravel & Lumen*. [online]. [cit. 2025-01-04]. Dostupné na: <<https://jwt-auth.readthedocs.io/en/develop/>>
- [8] W3SCHOOLS, 2025: *MySQL Tutorial*. [online]. [cit. 2025-01-25]. Dostupné na: <<https://www.w3schools.com/MySQL/default.asp>>
- [9] MARTIN, R.C., 2017. *Clean architecture: A craftsman's guide to software structure and Design*. Boston: Pearson Education. 432 s. ISBN 978-0-13-449416-6.
- [10] GUPTA, L., 2023. „*What is REST?: REST API Tutorial*“. [online]. [cit. 2025-01-25]. Dostupné na: <<https://restfulapi.net/>>

- [11] ASTHANA, A., 2025. *Postman Docs*. [online]. [cit. 2025-01-26]. Dostupné na: <<https://learning.postman.com/docs/introduction/overview/>>
- [12] GEEKSFORGEEKS, 2024. „*Web Technology*“. [online]. [cit. 2025-01-26]. Dostupné na: <<https://www.geeksforgeeks.org/web-technology/>>
- [13] OTWELL, T., 2025. *Laravel Documentation*. [online]. [cit. 2025-02-05]. Dostupné na: <<https://laravel.com/docs/9.x>>
- [14] MICROSOFT., 2022. *Visual studio code tutorial*. [online]. [cit. 2025-02-05]. Dostupné na: <<https://code.visualstudio.com/docs/introvideos/basics>>
- [15] J DMITRIEV, S, SHAFIROV, M, SKRYGAN, K., 2025. *php storm tutorial*. [online]. [cit. 2025-02-05]. Dostupné na: <<https://www.jetbrains.com/help/phpstorm/getting-started.html>>
- [16] WALLACE, E, FIELD, D., 2025. *Design basics*. [online]. [cit. 2025-02-05]. Dostupné na: <<https://www.figma.com/resource-library/design-basics/>>
- [17] ZHAO, I, PRUCHA, C, LAM, J, LAST, S, SCHACHMAN, T., 2025. *Notion guides*. [online]. [cit. 2025-02-05]. Dostupné na: <<https://www.notion.com/help/guides>>