

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Monitorování statistických údajů o nemovitostech



2021

Vedoucí práce:
Mgr. Tomáš Urbanec

Daniel Stonawski

Studijní obor: Aplikovaná informatika,
prezenční forma

Bibliografické údaje

Autor: Daniel Stonawski
Název práce: Monitorování statistických údajů o nemovitostech
Typ práce: bakalářská práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby: 2021
Studijní obor: Aplikovaná informatika, prezenční forma
Vedoucí práce: Mgr. Tomáš Urbanec
Počet stran: 36
Přílohy: 1 CD/DVD
Jazyk práce: český

Bibliographic info

Author: Daniel Stonawski
Title: Monitoring of real estate statistics
Thesis type: bachelor thesis
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense: 2021
Study field: Applied Computer Science, full-time form
Supervisor: Mgr. Tomáš Urbanec
Page count: 36
Supplements: 1 CD/DVD
Thesis language: Czech

Anotace

Tématem mé bakalářské práce je monitorovat statistické údaje nemovitostí a následně je zobrazit ve webové aplikaci, jejímž úkolem je co nejjednodušeji ukázat různá data uživatelům za pomoci grafů a různých výpočtů.

Synopsis

The theme of my bachelor thesis is monitoring of real estate statistics and then view the exact data in a web application, which is designed to show this data, in the easiest way, to the users using graphs and various calculations.

Klíčová slova: webová aplikace; statistické údaje; vizualizace;

Keywords: web application; statistics; visualization;

Děkuji Mgr. Tomáši Urbancovi za každou pomoc a cenné rady při vypracování bakalářské práce.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce

podpis autora

Obsah

1	Úvod	8
2	Získávání dat	9
2.1	Volba monitorovaného území	9
2.2	Volby webových stránek pro web scraping	9
3	Technologie	10
4	Teoretická část	11
4.1	Statistika	11
4.1.1	Použité a různé další statistiky nemovitostí	11
4.2	Webová Aplikace	12
4.2.1	Historie	13
4.2.2	Bezpečnost	13
4.3	Databáze	13
4.3.1	Vrstvy abstrakce dat	13
4.4	Web Scraping	15
4.5	Vizualizace Dat	16
5	Programátorská dokumentace	18
5.1	Struktura aplikace	18
5.2	Chování aplikace	18
5.3	Vytvoření databáze	19
5.4	Scrapování	20
5.4.1	Získávání dat	20
5.4.2	Ukládání dat	22
5.4.3	Automatické scrapování	22
5.4.4	Změna dat	23
5.5	Vzhled Aplikace	23
5.6	Aplikace	23
5.6.1	Plotly Dash	25
6	Uživatelská dokumentace	26
6.1	Grafická podoba	26
6.2	Domovská stránka	26
6.3	Stránky nemovitostí	27
6.4	Login	28
6.5	Register	28
6.6	Účet	29
6.7	Vizualizace Dash	29
6.8	Hlavní statistiky	31
7	Predikce	31
7.1	Problémy s predikcí	31

8 Závěr	33
9 Conclusion	34
10 Obsah přiloženého CD/DVD	35
Bibliografie	36

Seznam obrázků

1	Model databáze	20
2	Vzhled domovské stránky	27
3	Vzhled stránky Houses	28
4	Vzhled stránky Login	29
5	Vzhled stránky Register	30
6	Vizualizace vývoje ceny nemovitostí	30

Seznam tabulek

Seznam vět

Seznam zdrojových kódů

1	SQL vytvoření tabulky	19
2	DataFrame	21
3	Použití urlopen()	21
4	Získání unikátního kódu	21
5	DataFrame	22
6	SQLAlchemy-User	24
7	Formulář WTForms - Login	24
8	Input, Output	25
9	Úprava dat	26

1 Úvod

Výběr a nákup nemovitostí může být velmi zrádný. Ceny jednotlivých domů, bytů či pozemků se stále mění a vyvíjí a pokud si nepíšete poznámky někde do sešitu, předchozí ceny jednotlivých nemovitostí nejsou k dohledání. Účelem mé bakalářské práce bylo navrhnout a implementovat jednoduchou webovou aplikaci, která je schopna zobrazení získaných dat a jejich následné vizualizaci. Vizualizace nám může poskytnout mnohé informace, například vývoj ceny v čase, porovnání ceny s dalšími nemovitostmi v blízkém, či vzdáleném okolí atd.

K vypracování bakalářské práce jsem použil různé technologie. Pro získávání dat z webu jsem využil knihovny BeautifulSoup, která je jednou z nejpoužívanějších v oblasti web scrapingu. V rámci etiky scrapování dat bylo zapotřebí, abych kontaktoval administrátory různých webových stránek a domluvil se s nimi na podmínkách získávání dat z jejich webů. Pro vytvoření aplikace mi posloužil framework Flask, ke kterému je krásně zpracovaná dokumentace a velká komunita. Pro vizualizaci dat jsem použil framework Plotly Dash, který vznikl v roce 2017. Tento framework má pěkně zpracované fórum, na kterém se snaží být pár uživatelů aktivních a pomáhat s různými potížemi, stále jsou však některé otázky nezodpovězeny a při řešení problému je občas složité něco si dohledat.

2 Získávání dat

V této sekci popíši úvodní rozhodování z jak velkého území a z jakých serverů sbírám data.

2.1 Volba monitorovaného území

Mezi hlavní volby na začátku práce patřilo zvolení území, ve kterém bude monitorování nemovitostí probíhat. Po prozkoumání trhu s nemovitostmi jsem zjistil, že v jednotlivých okresech není dostatek nemovitostí pro práci s různými statistikami. Z tohoto důvodu bylo nutné se zaměřit na větší území, v mém případě na celý kraj. Zpočátku jsem se rozhodoval mezi krajem Moravskoslezským a krajem Olomouckým. Jelikož pocházím z kraje Moravskoslezského, byl jsem lehce nakloněn právě k této volbě. V čase výběru byl však v Moravskoslezském kraji nepoměr v počtu nabízených domů, bytů a pozemků. Z tohoto důvodu jsem se rozhodl pro kraj Olomoucký, kde byly počty vyvážené a tudíž jsem mohl v každé kategorii pracovat s dostatečným množstvím dat.

2.2 Volby webových stránek pro web scraping

Nejvíce informací o prodeji nemovitostí jsou na stránkách realitních kanceláří. Jak se později v dokumentu v sekci 3.4 dozvíme, při scrapování bychom správně měli kontaktovat správce stránek o tom, zda-li můžeme scrapování provádět. Po kontaktování několika stránek mi bylo umožněno získávat data ze stránek `realitymix.cz` a `reality.idnes.cz`.

3 Technologie

Na začátku jsem si zvolil programovací jazyk. Rozhodoval jsem se mezi jazykem C# a jazykem Python. Nakonec jsem se rozhodl pro jazyk Python, který patří mezi nejoblíbenější jazyky pro scrapování dat z internetu a má knihovnu BeautifulSoup, se kterou se velmi dobře pracuje. Pro web scraping jsem tedy použil tuto knihovnu. Dále bylo zapotřebí knihovny SQLite3 a Pandas pro vložení daných dat do databáze. Aplikace byla vytvářena využitím webového frameworku Flask. Pro návrh stránky byl použit HTML kód s kaskádovými styly. Celý projekt jsem vypracoval ve vývojovém prostředí PyCharm 2020, se kterým se mi velmi dobře pracovalo a mohu ho doporučit.

Python

Jazyk Python je skriptovací programovací jazyk, který v roce 1991 navrhl Guido van Rossum. Jeho hlavní předností je dynamická kontrola datových typů a podpora různých programovacích paradigmat včetně objektově orientovaného, funkcionálního, atd. Je vyvíjen jako open-source projekt, který je zdarma na většině běžných platform jako jsou MS Windows, macOS, UNIX, ... Ve většině distribucí systému GNU/Linux je dokonce součástí základní instalace. [5]

BeautifulSoup

Tato knihovna byla pojmenována po stejnojmenné básni od Lewise Carrola v pohádce Alenka v říši divů. Stejně jako v té pohádce, BeautifulSoup se snaží vnést trochu smyslu do něčeho nesrozumitelného, čímž kód HTML někdy může být. Pomáhá s formátováním a organizováním webu pomocí opravy špatného HTML a obdarovává nás s jednoduše spustitelnými Python objekty reprezentující XML struktury. Tato knihovna není součástí základní knihovny Pythonu a musí se nainstalovat zvlášť.

Flask

Mikro webový framework, který nevyžaduje konkrétní nástroje ani další vnitřní knihovny. Nemá žádnou vrstvu abstrakce databáze, formuláře ani jiné komponenty poskytující běžné funkce, avšak podporuje rozšíření, která mohou přidávat do aplikace další funkce. [7]

Dash

Dash je produktivní Python framework pro vytváření webových analytických aplikací. Postaven na základech Flasku, Plotly.js a React.js, Dash je ideální nástroj pro vytváření aplikací pro vizualizaci dat s libovolným uživatelským rozhraním v čistém Pythonu.

Přes několik jednoduchých vzorů dokáže abstrahovat všechny technologie a protokoly, které jsou nutné k vytvoření interaktivní webové aplikace. Je

natolik jednoduchý, že je možné navázat uživatelské rozhraní kolem python kódu během jednoho odpoledne.

Jako u Flasku, Dash aplikace jsou renderovány pomocí webových prohlížečů. Je možné umístit aplikaci na server a poté ji sdílet pomocí URL. Díky prohlížení přes webový prohlížeč, Dash aplikace jsou cross-platformové a podporují mobilní zobrazení. [6]

4 Teoretická část

V této části se zaměřím na teoretickou část práce.

4.1 Statistika

Statistika je velmi důležitým vědním oborem, jehož metody jsou aplikovány v mnoha jiných odvětvích. Statistiku můžeme použít hlavně v ekonomických vědách, ale také ve vědách přírodních, technických, ale také i společenských, jako například sociologie.

Pod pojmem statistika si lze představit spoustu významů. Jedním z hlavních významů je činnost, kterou získáváme údaje o obyvatelstvu, jeho struktuře a sociálně ekonomické situaci, stejně jako údaje o hospodaření podniků a o hospodářské situaci rozdílných územních celků a států. Jsou stanoveny různé statistické ukazatele, které se v průběhu času sledují. Zjištěné hodnoty pak mohou být dále porovnávány buď prostorově nebo časově.

Mezi další významy statistiky patří:

1. údaje, nejčastěji souhrnné charakteristiky získané z individuálních dat
2. praktická činnost, kterou jsou získávány a zpracovávány statistické údaje
3. vědní obor zaměřený na metody zjišťování, zpracování a analýzy dat

[2]

4.1.1 Použité a různé další statistiky nemovitostí

Průměrná prodejní cena

Poskytuje uživateli obrázek o tom, jestli je momentální cena nižší nebo vyšší než obvykle a pomůže s rozhodováním, jestli nemovitost koupit nyní nebo si ještě chvíli počkat. Někdy se také používá medián ceny, který není snadno ovlivnitelný nějakou velmi nízkou nebo naopak velmi vysokou cenou. Zároveň pokud mediánem sledujeme rozsáhlejší trh, můžeme porozumět, jak se trh vyvíjí.

Vývoj ceny

Vývoj ceny je měřítkem pohybu cen nemovitostí v určitém časovém úseku, někdy může být prezentován i v procentech. Pokud se cena o dost zvedla od prvního časového bodu k druhému, naznačuje to, že poptávka převyšuje nabídku a ceny rychle rostou.

Prodej nových domů

Statistika, která monitoruje prodeje nově postavených domů po nějakou časovou dobu. Poskytuje představu o aktivitě na trhu. Například pokud se zvedne počet prodaných nových domů, naznačuje to, že se poptávka zvyšuje, protože změny jsou velmi často viděny dříve na prodejích nových domů než na celkovém trhu. Zároveň taky tato statistika může naznačovat ekonomickou stabilitu státu, jelikož prodej nových nemovitostí naznačuje zvýšení sebevědomí a utrácení občanů.

Prodej starších domů

Zaměřuje se na domy, které již byly vlastněny před příchodem na trh. Je to důležitá statistika, protože tyto domy zabírají velké procento prodeje nemovitostí. Narozdíl však od prodeje nových domů, tato statistika shromažďuje již známé trendy na trhu.

Medián dní na trhu

Zachycuje věk zařazení nemovitosti na trhu. Statistika je vypočítána zjištěním, kolik dní uběhlo od té doby, kdy byla nemovitost zařazena na trh, po dobu, kdy byla prodána a následně vytvoření mediánu. Prodejci a agenti realitních kanceláří chtějí mít nemovitost prodanou co nejdříve a tento medián je značkou síly trhu. Pokud medián dní na trhu spadne po dobu několika měsíců, je možné si vyvodit, že poptávka se bude zvyšovat a ceny budou růst. [10]

4.2 Webová Aplikace

Webová aplikace je aplikace poskytovaná přes webový server pomocí počítačové sítě Internet. Kvůli všudepřítomnosti webových prohlížečů se tyto aplikace staly velmi populární. Právě tyto prohlížeče totiž používají jako klienta, přesněji řečeno tenkého klienta, jelikož sám prohlížeč nezná logiku aplikace. Mezi hlavní důvod jejich oblíbenosti patří schopnost aktualizovat a spravovat webové aplikace bez nutnosti šíření a instalování nového softwaru.

Tyto aplikace jsou používány zejména pro implementaci podnikových a jiných informačních systémů, ale také jako webové rozhraní e-mailové schránky. Jedny z nejnámějších aplikací jsou vyvíjeny firmou Google. Dále jsou používány pro internetové obchody, online aukce, diskusní fóra a mnohé další.

4.2.1 Historie

V začátcích této technologie aplikace využívaly typu klient-server, kde každá aplikace měla svůj vlastní klientský program, který musel být nainstalován přímo na počítači uživatele a sloužil jako uživatelské rozhraní. Tohle řešení mělo pár chyb, zejména tedy při aktualizaci serverové části musely být aktualizovány i klientské programy na každé stanici.

4.2.2 Bezpečnost

Z hlediska bezpečnosti je mnohdy lepší při práci s citlivými soubory zvolit lokálně instalovanou aplikaci. Naopak pro příležitostnou práci s daty, které neobsahují žádné důvěryhodné informace a nehrozí jejich zneužití, je webová aplikace vhodným pomocníkem.

4.3 Databáze

Databáze není v podstatě nic víc, než kolekce informací, která existuje po dlouhou dobu, většinou i několik let. V běžném jazyce je termín databáze používán a spojován s kolekcí dat, které jsou řízeny DBMS (Database Management System). Od DBMS se očekává:

1. Umožnění uživatelům vytvořit novou databázi a specifikovat jejich schémata.
2. Poskytnout uživatelům možnost dotázat se na data pomocí query a také upravit data užitím vhodného jazyka.
3. Podpora uložení velkého množství dat (v řádech terabytu) po velmi dlouhou dobu a umožnění přístupu k těmto datům.
4. Umožnit trvanlivost dat v případě vyskytnutí chyby jakéhokoliv druhu nebo v případě zneužití dat.
5. Kontrolovat přístup k datům od několika uživatelů zároveň bez povolení nečekaných interakcí mezi uživateli a bez provedených akcí s daty, které by byly provedeny částečně, ale ne kompletně.

4.3.1 Vrstvy abstrakce dat

Databáze jsou jedinečné v jejich dovednosti ukázat několika uživatelům jejich vlastním rozdílným pohledem data, která jsou uložena pouze jednou. Tyhle pohledy jsou kolektivně nazvány uživatelskými pohledy. Uživatel je v tomhle případě každý člověk nebo aplikace, která se přihlásí do databáze za účelem uložení nebo získání dat.

Když používáme aplikaci jako Microsoft Excel pro více uživatelů najednou, všichni tito uživatelé sdílí stejný pohled, který musí odpovídat tomu, jak jsou data fyzicky uložena v podkladových datových souborech. Pokud jeden z uživatelů provede jakékoliv úpravy a uloží je, další uživatel při otevření souboru uvidí data tak, jak je zanechal předchozí uživatel. Další z možností je, že každý z uživatelů uloží svou kopii v rozdílných fyzických souborech, ale v tomto případě, když jeden z uživatelů uloží svou verzi, verze všech ostatních se stane zastaralou. Databáze tento problém řeší tak, že ukáže všem uživatelům stejná data, ale toto zobrazení může být upraveno pro jednotlivce zvlášť. Protože každé zobrazení ve skutečnosti neukládá žádná data, automaticky tak zobrazuje každé změny dat, které nastanou v podkladových datových souborech. Tohle vše je umožněno díky vrstvám abstrakce.

Fyzická vrstva

Obsahuje datové soubory, které drží všechna data pro databázi. Téměř všechny moderní DBMS umožňují, aby databáze byla uložena v několika datových souborech, které jsou většinou rozšířeny přes několik fyzických datových disků.

Logická vrstva

Logická vrstva existuje pouze jako abstraktní datová struktura sestavena z fyzické vrstvy, pokud je potřeba. DBMS transformuje data v datových souborech do obecné struktury. Této vrstvě se občas říká schéma, termín používaný pro kolekci všech data itemů uložených v dané databázi nebo patřící danému uživateli databáze. S ohledem na DBMS, tato vrstva může obsahovat sadu dvou-dimenzionálních tabulek, hierarchickou strukturu podobnou k firemnímu organizačnímu grafu nebo nějakou jinou strukturu.

Externí vrstva

Externí vrstva nebo také externí model je druhá vrstva abstrakce v databázi. Tato vrstva se skládá z uživatelských zobrazení, které jsou kolektivně nazývány subschéma. V této vrstvě se uživatelé databáze, kteří dostanou přístup, připojí a zaměstnávají databázi dotazy. Ideálně, pouze DBA (administrátor databáze) jedná s fyzickými a logickými vrstvami. DBMS řídí transformaci vybraných složek z jedné nebo více datových struktur v logické vrstvě pro vytvoření každého uživatelského zobrazení. Uživatelské zobrazení může být v této vrstvě přednastaveno a uloženo v databázi pro opakované použití nebo může být dočasným prvkem, který je vytvořen DBMS pro udržení výsledků dotazů, dokud již nejsou pro uživatele podstatné. [3]

4.4 Web Scraping

Web scraping, web harvesting nebo také web data extraction je získávání dat z webových stránek. Software využívaný k této činnosti se může přímo napojit na web pomocí http protokolu a nebo pomocí webového prohlížeče. Důsledkem neustálého generování velkého množství různorodých dat na WWW, web scraping je považován za efektivní techniku pro získávání dat na internetu.

Tato technologie může být použita manuálně uživatelem softwaru, ale typicky se používá pro automatizované procesy implementované pomocí různých botů a nebo takzvaných web crawlerů. Jedná se o formu kopírování, ve kterém jsou požadovaná data nashromážděna a zkopírována přímo z webu. Většinou se tato data kopírují do databáze nebo do tabulek pro pozdější práci s těmito daty nebo jejich analýzou.

Proces scrapování dat se skládá ze dvou fází:

- Získání webových zdrojů
- Extrahování požadovaných informací ze získaných zdrojů

Mezi největší výhody patří skutečnost, že získávat data pomocí web scrapingu může jakýkoliv uživatel. Není zapotřebí žádné speciální vybavení a po lehkém studiu této technologie je schopen si sám naprogramovat svůj kód pro získání dat.

Další důležitou věcí, která je potřeba zmínit je to, že web scraping je naprosto legální, avšak existují pravidla, kterými je potřeba se řídit.

Neuškodit stránce

První pravidlo scrapování je: Neuškodit stránce. Tohle znamená, že frekvence a velikost dotazů, které vytvoříme, by neměla nijak zatěžovat stránku a omezovat její chod.

Toho se dá dosáhnout třemi způsoby:

1. Limitováním počtu požadavků na webovou stránku.
2. Respektováním prodlení, po které crawlery musí čekat mezi požadavky.
3. Pokud je to možné, plánovat scrapování v době nejmenšího zatížení stránky.

Neporušit copyright

Copyright je definován jako exkluzivní právo nad nějakým kouskem práce, například článek, obrázek, film, atd... Takových prací je na internetu velké množství a tudíž je nutné si dávat velký pozor na to, jaká data scrapujeme.

Neporušit GDPR

Zavedení GDPR naprosto změnilo možnosti scrapování. Abychom GDPR neporušili, musíme se vyvarovat scrapování dat občanů jako například jejich jména, emailu, telefonního čísla apod. [1, 11]

Co se týká správného chování při scrapování, je také vhodné kontaktovat administrátora nebo správce daných stránek a sdělit mu, o co se pokoušíte a jestli nějakým způsobem neomezujete chod stránek. Popřípadě je dobré požádat o jeho schválení a také, pokud váš crawler blokuje při větším kopírování dat, je možnost je požádat o nějakou výjimku. Při scrapování ze stránek <https://reality.idnes.cz> se mi vyskytl problém, kdy pokud jsem nenastavil prodlení aspoň na minutu při každém novém dotazu, byl můj crawler zablokovan. S těmito stránkami se mi však nepodařilo navázat kontakt, tudíž scrapování z nich trvá něco okolo tří hodin. Při kontaktování ostatních stránek jsem dostal smíšené odpovědi a u některých jsem se odpovědi ani nedočkal. Ze stránek <https://sreality.cz> mi bylo řečeno, že si nepřejí žádnou takovou činnost na jejich stránkách. Tohle se může stát, ale naštěstí je na internetu plno dalších, kteří byli vstřícnější.

4.5 Vizualizace Dat

Jedná se o proces zkoumání dat a informací, které byly převedeny do grafické podoby. Mezi hlavní cíle patří pochopení zkoumaných jevů a vniknutí do daného problému. Vizualizace dat bývá často zaměňována s prezentační grafikou, která zobrazuje již známé informace, kdežto vizualizace dat je prostředkem k poznávání dosud neznámých skutečností a slouží k jejich pochopení. Slouží také jako prostředek pro usnadnění vědeckých metod. Jako první použil vizualizaci dat John Snow v letech 1854-1855, kdy sestavil mapu rozšíření cholery, díky níž odhalil zdroj nákazy.

Pomocí grafů a různých map může vizualizace dat poskytnout přístupnou cestu k pochopení různých vzorů, trendů či odlehlých hodnot v datech. Ve světě velkého množství dat jsou tyto nástroje a technologie velmi důležité k analýze velkého množství informací a jsou schopné na základě získaných dat dělat důležitá rozhodnutí.

Výhody

Mezi hlavní výhody patří rychlé rozhodování. Shrnutí dat je jednoduché a rychlé, tím pádem se různá data zdají být více jasná, než když je vidíme například v tabulce Excelu. Jednou z výhod je také lepší porozumění problému. Další výhodou je atraktivita problému, čím hezčí znázornění dat je, tím více lidí zaujme. A v neposlední řadě, většina lidí je lepší v porozumění a zapamatování si informací, které jsou zobrazeny vizuálně.

Základy k úspěšné vizualizaci

- Zvolení správné grafiky odvíjející se od cíle vizualizace.
- Potvrzení třetí osoby, že zobrazujeme to co jsme požadovali.
- Zvolení správného designu pro uživatele.

Typy grafů

Pokud zvolíme špatný graf, uživatelé mohou být zmateni nebo si výsledek vyložit špatně, proto je důležité si před vytvořením grafu ujasnit, jaké data chci zobrazit a za jakým účelem.

Čárový

Čárový graf zobrazuje, jak se jedna nebo více proměnných mění přes různé datové body. Tento typ grafu je vhodný pro zobrazení změn dat vzhledem k časové ose.

Sloupcový

Velmi vhodný pro porovnání datových souborů. Horizontální sloupcový graf je většinou používán při porovnávání velkého počtu dat nebo při zobrazení zřetelného rozdílu jednoho z datových souborů. Vertikální sloupcový graf je používán pro vývoj dat v čase.

Koláčový

Zobrazuje podíl každé hodnoty v datovém souboru. Je používán k zobrazení komponent jakýchkoliv dat. Například, kolik procent z celkového výdělku patří určitému produktu. [12]

Problémy s vizualizací

Některá data je však těžké vizualizovat. Například dříve zmíněné statistiky o prodeji nových domů, které jsou sice velmi zajímavé, ale z pohledu mé aplikace se nedají nijak efektivně zobrazit. Tato statistika by se dala zobrazit pouze na grafu poptávky a to ještě ne úplně spolehlivě. Další informace, které z této statistiky získáme, jako například ekonomická stabilita státu, jsou těžko zobrazitelné.

5 Programátorská dokumentace

Následující část je zaměřena na programátorskou část práce. Za cíl bylo získání dat z webu a vytvoření uživatelsky příjemné aplikace pro vizualizaci získaných dat.

5.1 Struktura aplikace

Aplikace je umístěna ve složce „reapp“ a ve spouštěcím souboru „run.py“.

1. reapp - balíček obsahující dvě složky a dále soubory jako například forms.py, models.py, atd.
 - (a) plotlydash - složka, ve které jsou všechny potřebné soubory pro vizualizaci dat
 - (b) static - balíček obsahující veškeré statické soubory(css, img, ...)
 - (c) templates - balíček obsahující veškeré html soubory
2. run.py

5.2 Chování aplikace

Aplikace při prvním otevření zobrazí hlavní stránku, na které se uživateli zobrazí tlačítko pro přechod k vizualizacím. Zároveň můžeme na horním okraji stránky najít navigaci, ve které jsou odkazy na různé stránky aplikace.

- Houses - Při otevření této stránky se vygeneruje list nabízených domů či vil v Olomouckém kraji. Každý takto vygenerovaný prvek listu obsahuje jednoduché informace o dané nemovitosti s odkazem na původní stránku, ze které jsou informace převzaty. Pokud však není stránka dostupná, tento odkaz se nezobrazí. Uprostřed nahoře je taky zobrazeno datum vystavení inzerátu a datum jeho zaniknutí. Na těchto stránkách jsou však zobrazeny všechny inzeráty, aby měl uživatel možnost porovnat aktuální inzeráty s inzeráty v minulosti.
- Flats - Při otevření této stránky se vygeneruje seznam nabízených bytů v Olomouckém kraji.
- Lands - Při otevření této stránky se vygeneruje seznam nabízených pozemků v Olomouckém kraji.

- Login - Po kliknutí na záložku login nás aplikace přesměruje na formulář k přihlášení. Pro přihlášení je nejprve zapotřebí účet vytvořit a zaregistrovat se. K tomu nám poslouží odkaz v pravém horním rohu nebo pod přihlašovací formulářem. Pokud již vytvořený účet máme, můžeme se do něj přihlásit pomocí emailu a hesla. Po přihlášení se změní navigace, ubydnou záložky login a register, které nahradí přezdívkou přihlášeného a také v levé části navigace přibude položka MyRE.
- Register - Jak už jsem zmínil u předchozí záložky, přes tuhle záložku se dostaneme k formuláři k registraci. Po vyplnění formuláře a úspěšném zaregistrování nás stránka přesměruje na stránku k přihlášení.
- Účet - Zobrazí přehled uživatelského účtu, dovoluje změnit potřebné informace k přihlášení a profilový obrázek.
- MyRE - Slouží jako uživatelská schránka, do které si může ukládat nemovitosti, které se mu zalíbily, aby pak později nemusel opakovaně hledat v dlouhém seznamu inzerátů. Po otevření se uživateli zobrazí seznam všech uložených nemovitostí bez ohledu na to, jaký je to druh nemovitosti.

5.3 Vytvoření databáze

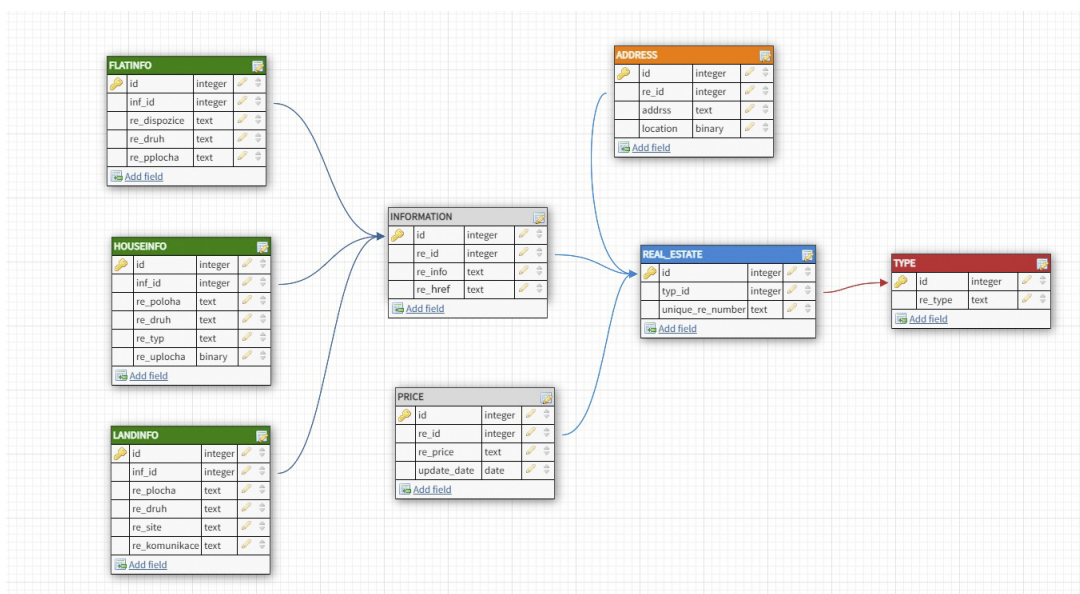
Pro vytvoření databáze jsem použil knihovnu v pythonu SQLite3. Je to velmi používaná knihovna s rozsáhlou dokumentací. Nejprve bylo zapotřebí si ujasnit, jaká data budu chtít ukládat a jak budou muset být jednotlivé tabulky propojeny. Nakonec jsem se rozhodl pro schéma, které má jednu hlavní tabulku `REAL_ESTATE`, ve které jsou všechny nemovitosti se svým unikátním klíčem a na její id jsou napojeny další tabulky, jako např. adresa, cena, atd. Tabulku `REAL_ESTATE` jsem vytvořil takto:

```

1 CREATE TABLE IF NOT EXISTS REAL_ESTATE
2 (
3 ID INTEGER PRIMARY KEY,
4 UNIQUE_RE_NUMBER TEXT,
5 TYP_ID INT,
6 FOREIGN KEY (TYP_ID) REFERENCES TYPE (ID)
7 )

```

Zdrojový kód 1: SQL vytvoření tabulky



Obrázek 1: Model databáze

5.4 Scrapování

Získávání dat z internetu je rozsáhlý proces, který je potřeba si předem rozvrhnout. Začal jsem vybráním webových stránek, které by byly vhodné pro získání potřebných dat. Pro analýzu těchto stránek je možné použít různých webových prohlížečů, které disponují funkcí, která umožňuje prohlížet jednotlivé prvky stránky. Já používal webový prohlížeč Chrome, se kterým se mi velmi dobře pracuje. Stránky se většinou hodně liší a data, která bych chtěl získat, jsou uložena v různých elementech. Vybíral jsem takové stránky, které byly přehledné a obsahovaly nějakou jednotnou strukturu. Dalším krokem je zjištění, ve kterých prvcích jsou data uložena a přes jaké třídy se k nim lze dostat. Až potom je vhodné začít psát kód.

Pro scrapování dat v pythonu jsem zvolil knihovnu BeautifulSoup, která je srozumitelná a existuje pro ni rozsáhlá dokumentace.

5.4.1 Získávání dat

V první řadě je zapotřebí pomoci funkce `urlopen()` nastavit, jakou stránku budeme scrapovat. Jelikož stránky realitních kanceláří jsou většinou rozsáhlé a mají několik stránek, je zapotřebí také zajistit, aby se web crawler dostal na všechny tyto další stránky. Většina stránek má na konci stránky prvky s odkazy na navazující stránky očíslované od první stránky po poslední. Po prohlédnutí stránek jsem zjistil, že v každém osmém prvku se nachází odkaz na poslední stránku. Z tohoto posledního prvku jsem tedy získal číslo poslední stránky, které jsem použil v cyklu a také při odkazu pro crawler, který jako argument bere url adresu stránky.

```

lastPageStr1 = allPages[8].find('a').get_text()
lastPageInt1 = int(lastPageStr1)

for page in range(1, lastPageInt1):
if page == 1:
    html = urlopen
        ('https://realitymix.cz/vyhledavani/
            olomoucky/prodej-bytu.html')
else:
    html = urlopen
        ('https://realitymix.cz/vyhledavani/
            olomoucky/prodej-bytu.html?stranka='
            + str(page))

```

Zdrojový kód 2: DataFrame

Jednotlivé nemovitosti se většinou vyskytují v ucelených prvcích stránek pod názvem určité třídy. Všechny prvky této třídy je potřeba najít pomocí příkazu `.find_all()`. Od tohoto momentu se u všech stránek začne kód většinou lišit, data jsou uložena v rozdílných elementech stránky a pod různými názvy tříd. Z hlavních stránek, kde máme uloženy všechny nemovitosti, lze většinou zjistit cena, adresa a krátké informace u jednotlivé nemovitosti. Pro důkladnější informace je zapotřebí otevřít nový web crawler, který rozklikne odkaz nemovitosti a dostane se na stránku, kde je vše vypsáno.

```

1 href = 'https://reality.idnes.cz/' + item.a['href']
2 html2 = urlopen(href)
3 bs2 = BeautifulSoup(html2.read(), 'html.parser')

```

Zdrojový kód 3: Použití urlopen()

Tento druhý crawler se opět orientuje podle tříd nebo elementů, ve kterých jsou potřebná data uložena, např. pro získání unikátního kódu jsem musel najít veškeré elementy `dt` a potom v cyklu zjišťovat, který element obsahuje požadovanou informaci.

```

details = bs2.find(class_='b-definition-columns')
labels = details.find_all('dt')
for label in labels:
    if label.get_text() == 'Cislo zakazky':
        unique_code =
            unique_code + [table[labelNumber].get_text()]

```

Zdrojový kód 4: Získání unikátního kódu

Nemohl jsem použít pevný index elementu, jelikož se unikátní kód nacházel u každé nemovitosti na jiné pozici.

5.4.2 Ukládání dat

Vyscrapovaná data jsem ukládal do unikátního pole pro každou rozdílnou informaci. Tato pole jsem následně vkládal do dataframu z knihovny Pandas. Tato knihovna skvěle spolupracuje s knihovnou BeautifulSoup a také SQLite3, kterou jsem používal pro vytvoření databáze a následně i pro ukládání dat do databáze.

Při ukládání dat do dataframu je zapotřebí, aby všechna vkládaná pole měla stejný počet prvků. S touto podmínkou jsem měl často problém. Při scrapování dat je někdy možné, že někdy zvolíte špatný element, ve kterém nejsou žádná data. V tomto případě se nemusí několik záznamů do pole nahrát a při následném ukládání polí do dataframu nám program spadne. Vzhledem k tomu, že scrapování je někdy velmi časově náročné, takovéto chyby mohou značně prodloužit vývoj celého projektu.

Pokud jsou data v pořádku, vytvoření je pak už jednoduché a jednotlivá pole jsou uloženy pod svůj vlastní label.

```
lastPageStr1 = allPages[8].find('a').get_text()
lastPageInt1 = int(lastPageStr1)
```

```
Price_frame = pd.DataFrame(
    {
        'ID': id_arr,
        'RE_ID': re_id,
        'RE_PRICE': price,
        'UPDATE_DATE': re_date
    }
)
```

Zdrojový kód 5: DataFrame

Pro uložení daného dataframu do databáze pak stačí použít funkci `.to_sql()` s parametry Jméno tabulky a připojení databázi

5.4.3 Automatické scrapování

Jelikož scrapování některých stránek trvá i několik hodin, bylo zapotřebí tento proces zautomatizovat. Z tohoto důvodu jsem požádal o vytvoření studentského serveru, na který jsem nahrál svůj kód a pomocí Windows plánovače úloh nastavil každodenní spuštění různých částí kódu během celého dne. Takto jsem byl schopen získat velké množství dat každý den a nezatěžoval jsem svůj počítač.

Nastavení úlohy je jednoduché, při rozkliknutí plánovače klikneme na vytvořit úlohu..., kde na nás vyskočí okénko se spoustou nastavení. V obecném nastavení je důležité zaškrtnout políčko Spustit nezávisle na přihlášení uživatele. V sekci Aktivační události pak nastavíme frekvenci jak často a v jakou hodinu chceme program spustit. V sekci Akce si navolíme, jaký program chceme spustit.

5.4.4 Změna dat

Po nějakém uplynulém čase může nastat situace, kdy se stránky, ze kterých data bereme, změní. V důsledku toho se nám přestanou data stahovat, jelikož námi napsaný kód již nadále neodpovídá názvům elementů na upravených stránkách. V takovém případě je zapotřebí nastudovat, jak velké změny na stránkách nastaly a jestli je zapotřebí změnit celý kód nebo jen upravit názvy proměnných v kódu stávajícím.

Během vypracování práce se mi stalo, že se změnila struktura stránek `idnes.cz` a to zejména v sekci s výpisem nemovitostí, která pro mne byla podstatná. Díky těmto změnám již nebyl můj kód aktuální a tudíž jsem nezískával nová data. Nejprve jsem musel zjistit, jestli jsou pro mě důležitá data uložena ve stejných elementech a pod stejným názvem tříd. Pozice mnou chtěných dat byla většinou stejná, došlo však ke změnám v názvech tříd, jako například u ceny nemovitosti, kde se název změnil z `c-item-products__price` na `c-products__price`.

5.5 Vzhled Aplikace

Jelikož Flask dovoluje použít html template, vzhled a rozvržení stránek jsem vypracoval za užití právě html a css. Velkým pomocníkem byla i knihovna bootstrap, která ulehčila práci s responzivitou.

Stěžejní prvek aplikace představuje menu, které je prakticky stejné na každé stránce. Pro tento účel jsem vytvořil vzorový soubor `layout.html`, který pomocí příkazu `extends "layout.html"` aplikujeme na všechny ostatní html stránky.

5.6 Aplikace

Sekce obsahující popis hlavních částí aplikace.

run.py

Soubor pro spuštění celé aplikace. Lze nastavit možnost debugování.

__init__.py

Slouží jako tvůrce aplikace a zároveň také říká Pythonu, aby zacházel s `reapp` (složka, ve které je soubor umístěn) jako s balíčkem. Pro vytvoření instance flasku slouží příkaz `app = Flask(__name__)`. V tomhle souboru je také zapotřebí implementovat jakékoliv přidané funkcionality aplikace, jako např. `db = SQLAlchemy(app)` pro databázi, která ukládá uživatele aplikace.

routes.py

Funkce `route()` je používána k propojení funkce s URL. Ve výsledku, vždy když je URL zmíněno v prohlížeči, funkce je provedena a vrací výsledek. Pomocí funkce `redirect()` lze přesměrovat uživatele na jinou stránku za použití routování tak, že do výstupní hodnoty funkce zadáme novou URL adresu. Pro posílání proměnných mezi jednotlivými stránkami je nutno

použít funkce `url_for()`, její první argument je jméno funkce a druhý argument jsou posílaná data.

models.py

Pro ukládání uživatelů používám knihovnu SQLAlchemy. Tato knihovna pracuje velmi dobře s knihovnou Flask, dokonce mají i své rozšíření, Flask-SQLAlchemy, které však v mém případě nebylo zapotřebí.

```
class User(db.Model, UserMixin):
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(20),
        unique=True, nullable=False)
    email = db.Column(db.String(150),
        unique=True, nullable=False)
    profilePic = db.Column(db.String(20),
        nullable=False, default='default.jpg')
    password = db.Column(db.String(60),
        nullable=False)
    saved = db.relationship('Saved', lazy=True)
```

Zdrojový kód 6: SQLAlchemy-User

K vytvoření tabulky se používá instance třídy a pro vytvoření jednotlivých sloupců tabulky použijeme funkci `Column()`. Při správném vyplnění registračního formuláře se uloží záznam uživatele do tabulky a při přihlášení se kontroluje, jestli v tabulce uživatel existuje.

forms.py

Formuláře poskytují nejvyšší formu rozhraní (API) v knihovně WTForms. Obsahují definice polí, delegují validaci, berou vstupy, shrnují chyby a celkově fungují jako lepidlo, které drží vše pospolu. V mé aplikaci byly tyto formuláře použity pro přihlášení, registraci a změnu dat uživatele. Pro definování formuláře je zapotřebí vytvořit podtřídu třídy `Form` a definovat pole jako atributy třídy.

```
class loginForm(FlaskForm):
    email = StringField('Email',
        validators=[DataRequired(), Email()])
    password = PasswordField('Password',
        validators=[DataRequired()])
    rememberMe = BooleanField('Remember me')
    submit = SubmitField('Login')
```

Zdrojový kód 7: Formulář WTForms - Login

Těmto polím jsou také přiřazeny popisky a validátory, které zaručují, že budou vloženy správná data. Například validátor `DataRequired()` zaručuje, že formulář nebude možno odeslat, dokud toto pole nebude vyplněno.

5.6.1 Plotly Dash

Při odkázání na stránku s vizualizací se spustí samostatná aplikace Dash, která je uvnitř hlavní Flask aplikace. Aplikace Dash se dělí na dvě části.

Tou první částí je rozložení stránky, které určuje, jak bude aplikace vypadat. V mém případě jsem chtěl replikovat vzhled všech ostatních stránek aplikace a k tomu jsem použil funkce ze souboru `layout.py`, která přepisuje rozložení stránky aplikace Dash. Do této funkce jsem nahrál stejný kód jako u rozložení ostatních html stránek. Do rozložení patří také prvky, které jsou součástí knihoven `dash_core_components` a `dash_html_components`. Jelikož můj vlastní layout obsahuje pouze menu, do obsahu stránky přijdou právě tyto komponenty.

Druhá část popisuje interaktivitu aplikace a skládá se ze zpětných volání, které upravují data podle toho, jak uživatel s aplikací pracuje.

Dash HTML a Core komponenty

Hlavní komponent je `dcc.Graph`, který je pro mou vizualizaci klíčový. Tento prvek jsem umístil jako první. Pod něj jsem umístil potřebné rozbalovací nabídky pro různá data. Mezi další z hlavních komponent patří `dcc.RadioItems`, který tvoří seznam zaškrťovacích políček. Tato políčka používám k vybrání, jaký typ nemovitostí chce uživatel zobrazit. Všechny tyto komponenty jsou děti html komponentu `div`, který pak v css stylizují, aby stránka vypadala podle mých představ. Všem těmto komponentům je zapotřebí přiřadit id pro identifikaci ve zpětném volání.

Zpětná volání

Zpětná volání akceptují dva druhy argumentů, vstupy a výstupy, se dvěma parametry. Jeden parametr je id komponentu, se kterým chceme pracovat a druhý parametr je hodnota komponentu, kterou bude zpětné volání měnit. Jako třetí prvek zpětného volání je samotná funkce, která akceptuje tolik parametrů, kolik má volání vstupů a vrací tolik výsledků, kolik má volání výstupů.

```
Output('drpdwn-1', 'disabled'),  
Input('radio-types', 'value')
```

Zdrojový kód 8: Input, Output

V mé aplikaci používám několik různých volání pro různé výstupy, Dash totiž nedovoluje upravovat jeden výstup dvěma nebo více zpětnými voláními. Hlavním voláním je úprava grafu, ve kterém беру vstupy ze všech vstupních komponentů (rozbalovací nabídky, zaškrťovací políčka) a přenáším tato nastavení na graf za použití podmínek a úpravy dat, která graf zobrazuje.

```

if loc != 'None' or len(loc) == 0:
    dff = dff[df['LOCATION'].isin(list(loc))]
    dfPrice = dfPrice[dfPrice['LOCATION'].isin(list(loc))]
    dfAvg = dfAvg[dfAvg['LOCATION'].isin(list(loc))]

```

Zdrojový kód 9: Úprava dat

Data jsou opět uložena v Pandas DataFrame, ve kterém lze data filtrovat pomocí syntaxe `dff = dff[df['LOCATION'].isin(list(loc))]`, která pokud se lokace nachází v dataframe a v seznamu `loc`, zůstane, jinak ji dataframe vypustí. Data lze takto také porovnávat za pomoci různých matematických operátorů.

Pro načtení některých komponentů, které se mění při otevření vizualizace přes jednotlivé nemovitosti z jedné ze stránek Houses, Flats nebo Lands používám další volání. Tohle tentokrát spolupracuje s komponentem `dcc.Location`, který reprezentuje umístění nebo adresový řádek ve vašem prohlížeči. Díky hodnotě „search“ daného komponentu jsme schopni zjistit, na jaké url adrese se zrovna nacházíme. Při zobrazení grafu přes nemovitost se url adresa liší a díky tomuto prvku jsme schopni určit, jak má zpětné volání nastavit jednotlivé komponenty.

6 Uživatelská dokumentace

Následující text bude zaměřen na objasnění užití aplikace. Mým cílem bylo vytvořit jednoduché rozhraní pro širokou škálu uživatelů.

6.1 Grafická podoba

Snažil jsem se zvolit co nejjednodušší design, který bude pro běžného uživatele příjemný na pohled a zároveň nijak nezhoršuje přehlednost stránek. K tomu mi posloužila světle fialová barva pro menu a některé prvky aplikace, například okraje tlačítek. Šedá pro objekty jednotlivých nemovitostí a bílá pro pozadí a text psaný fontem Open-Sans.

6.2 Domovská stránka

Po spuštění aplikace se uživateli zobrazí jednoduchá domovská stránka rozdělená na dvě hlavní části.

První částí je menu, které obsahuje odkazy na další stránky aplikace a je rozděleno na levou a pravou stranu. Na levé straně jsou odkazy na stránky s přehledem nemovitostí a také domeček, kterým se dostaneme zpět na hlavní stránku. Tato strana zůstává v celé aplikaci neměnná. Na pravé straně můžeme najít odkazy na formuláře pro přihlášení a registraci uživatele. Tato část se změní, pokud se uživatel přihlásí. Po najetí kurzoru myši na položku menu dojde ke změně barvy textu na šedou.

Welcome To The Real Estate App

To see visualisation of all the data gathered here, click down below.

DASHAPP

Obrázek 2: Vzhled domovské stránky

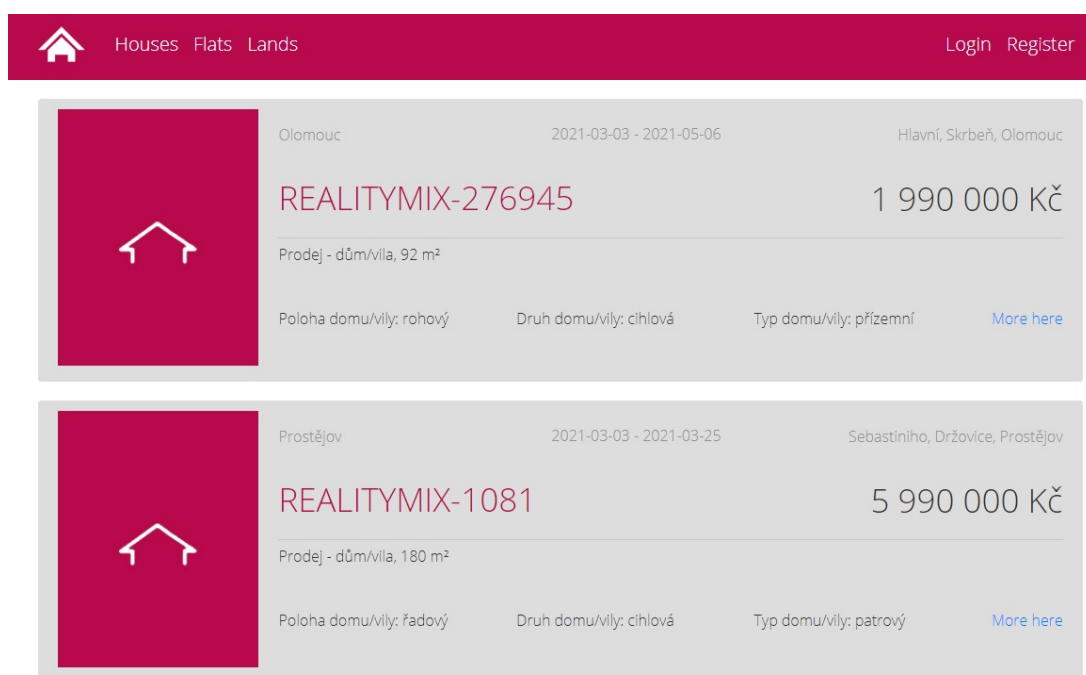
Druhou část tvoří obsah stránky, ve kterém se nachází nadpis a také tlačítko s odkazem na nejdůležitější funkci aplikace, kterou je vizualizace.

6.3 Stránky nemovitostí

Při kliknutí na jeden ze tří prvků menu (Houses, Flats, Lands) aplikace otevře stránku s přehledem všech dostupných nemovitostí v aplikaci podle zvoleného typu. Každá nemovitost má svůj vlastní objekt, ve kterém se nachází její unikátní kód, cena, okres a adresa.

Také jsou v nich další podrobnější informace o dané nemovitosti, u domů je to například poloha domu (rohový, řadový, ...) nebo také druh domu (cihlový, smíšený, ...). U každé nemovitosti je také odkaz na původní stránky, kde si o nemovitosti může uživatel najít podrobnější informace a také více obrázků pro lepší představu, o jakou nemovitost se jedná.

Další informací, kterou každý prvek listu obsahuje, je datum vystavení inzerátu a datum poslední aktualizace inzerátu. V aplikaci se nacházejí i takové inzeráty, které nebyly aktualizovány delší dobu. Je to z toho důvodu, aby si mohl uživatel takový inzerát dohledat a případně porovnat s nějakým aktuálním inzerátem, jestli se cena v dané lokalitě nějak změnila podle období nebo jestli je nezměněná.



Obrázek 3: Vzhled stránky Houses

6.4 Login

Pokud v navigaci klikneme na „Login”, dostaneme se na stránku přihlášení. Na této stránce najdeme formulář pro vyplnění přihlašovacích údajů. Do první kolonky je zapotřebí vyplnit email a do druhé heslo použité při registraci. Máme také na výběr zaškrtnout políčko „Remember Me”, které si bude pamatovat uživatelské údaje.

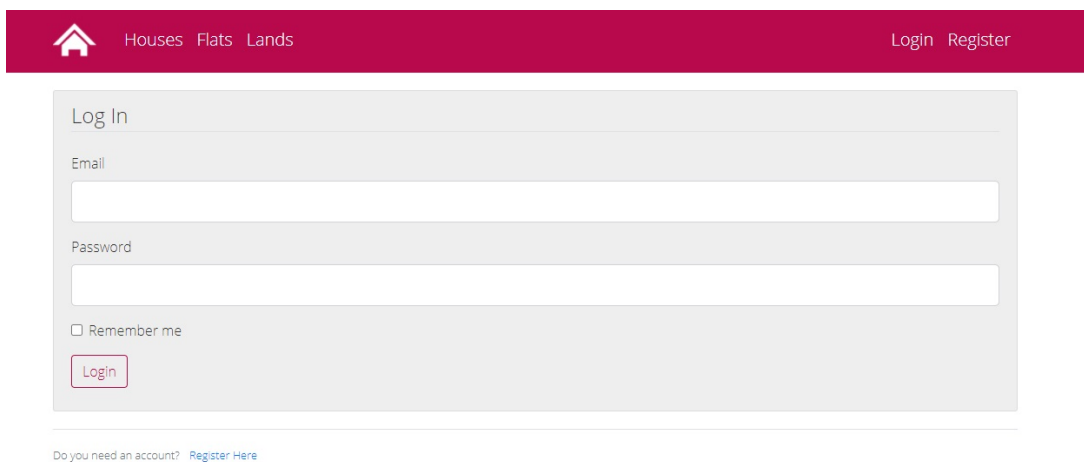
Pokud ještě nemáme vytvořený účet, dole pod formulářem je malý odkaz „Register Here”, který nás přesměruje na registrační formulář.

6.5 Register

Ať už po kliknutí na odkaz v menu, nebo po přesunutí z přihlašovacího formuláře, máme možnost po otevření této stránky vytvořit nového uživatele pomocí formuláře pro registrování. Je nutno si zvolit uživatelské jméno, které dosud nebylo použito. Pokud už zvolené jméno existuje, formulář o tom uživatele informuje. Dále je zapotřebí zvolit email, díky kterému se může uživatel následně přihlásit. U emailu se kontroluje, jestli je validní (např. user@gmail.com je validní, ale např. user.cz nebo user@user jsou nevalidní) a jestli již daný email nebyl použit. Mezi poslední část registrace patří heslo a potvrzení zadaného hesla, tyto dvě kolonky musí být vyplněny stejnými znaky.

Po úspěšné registraci je uživatel přesměrován na stránku s přihlášením, pokud se však uživatel dostal na stránku registrace omylem a chce se přihlásit, k tomu

slouží malý odkaz pod formulářem „Sign In”



The image shows a web application's login interface. At the top, there is a dark red navigation bar containing a home icon, the text 'Houses Flats Lands', and links for 'Login' and 'Register'. Below this is a light gray login form with the following elements: a title 'Log In', an 'Email' input field, a 'Password' input field, a checkbox labeled 'Remember me', and a 'Login' button. At the bottom of the form, there is a link that reads 'Do you need an account? Register Here'.

Obrázek 4: Vzhled stránky Login

6.6 Účet

Po přihlášení se vzhled menu lehce změní. Na místě, kde se předtím nacházely odkazy „Login” a „Register”, se nyní zobrazí avatar uživatele s jeho uživatelským jménem. Po kliknutí na uživatelské jméno se uživatel dostane na stránku s nastavením svého účtu. Zde si může změnit své uživatelské jméno, email, ale také i svůj avatar. Je také možné se zde ze svého účtu odhlásit po kliknutí na tlačítko „Log Out”.

6.7 Vizualizace Dash

Na úvodní stránce se nachází tlačítko „DASHAPP”. Toto tlačítko nás přesune na stránku aplikace, kde se provádí vizualizace dat. Po načtení stránky se jako první zobrazí graf. V základním nastavení je tento graf sloupcový a zobrazuje aktuální cenu jednotlivých nemovitostí.

Pod tímto grafem je sekce s nastavením daného grafu. Mezi hlavní úpravy grafu patří nastavení na prvním řádku. Tím je zvolení statistiky, která se v grafu zobrazí. Další nastavení pracují s těmito zvolenými statistikami. Graf lze upravit pomocí nastavení lokality, ze které má brát nemovitosti nebo lze přímo zvolit dané nemovitosti, které má zobrazit. Takto můžeme například nastavit, aby se vykreslil vývoj ceny u takto zvolených nemovitostí v průběhu monitorování. Součástí nastavení je také i možnost zvolit typ nemovitosti, které se mají zobrazit. V základu je graf nastaven tak, aby zobrazoval všechny. Při změně nastavení typu nemovitosti se změní poslední čtyři rozbalovací nabídky podle toho, jaký typ uživatel zvolil a tato nastavení rozšiřují možnosti vizualizace.

[Home](#)
[Houses](#)
[Flats](#)
[Lands](#)

[Login](#)
[Register](#)

Join Today

Username

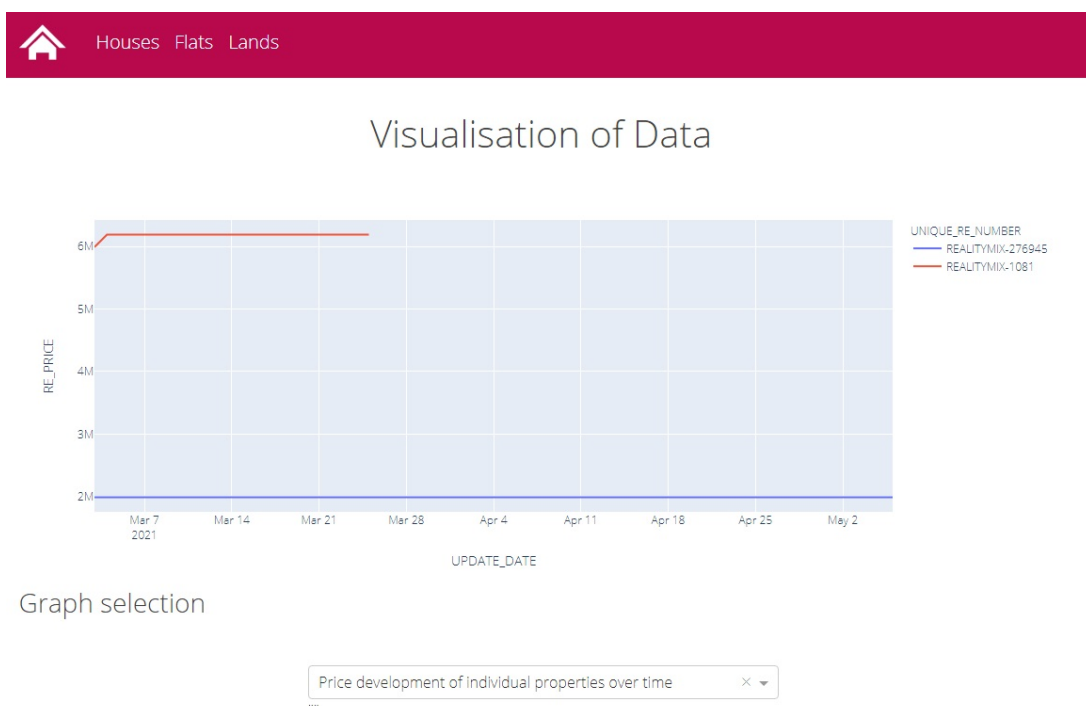
Email

Password

Confirm Password

Already Have An Account? [Sign In](#)

Obrázek 5: Vzhled stránky Register



Obrázek 6: Vizualizace vývoje ceny nemovitostí

6.8 Hlavní statistiky

Development of the price of real estate in the location

Vývoj celkové ceny v jednotlivých lokacích v čase.

Average price development of real estate in the location

Vývoj průměrné ceny v jednotlivých lokacích v čase.

Price development of individual properties over time

Vývoj ceny jednotlivých nemovitostí v čase. Pro fungování této statistiky je potřeba navolit jednotlivé nemovitosti ze seznamu níže.

Price of all properties in the location

Celková cena všech nemovitostí v dané lokaci.

Average price of real estate in certain area

Průměrná cena nemovitostí v dané lokaci.

Price of individual real estate

Zobrazení ceny všech dostupných nemovitostí pomocí sloupcového grafu.

Average price of individual real estate

Zobrazení průměrné ceny všech dostupných nemovitostí pomocí sloupcového grafu.

Visualisation of new posted real estate in certain date

Zobrazení, jak často se data pro nemovitosti doplňují. Lze vyčíst taky doba vytvoření a zániku inzerátu.

7 Predikce

Predikce je proces nacházení zajímavých vzorů v datech. Čerpá z několika příbuzných odvětví, které jsou někdy používány déle jak 100 let, jako například statistika.

Mezi hlavní rozdíly od ostatních analytik patří to, že predikce je poháněna daty, což znamená, že algoritmus si odvodí klíčové charakteristiky přímo z dat na rozdíl od předpokladů vytvořených nějakým analytikem. Dalším rozdílem je automatizace procesu nalézání vzorců v datech.

7.1 Problémy s predikcí

Predikce mohou generovat nezměrné vylepšení v účinnosti, rozhodování a také například v návratu investic. Ovšem nejsou vždy úspěšné a ve většině případů jsou predikátové modely nepoužitelné. Mezi nejčastější problémy patří:

Problémy v datech

Predikátové modely potřebují data ve formě jednoduchých tabulek obsahující řádky a sloupce. Pokud jsou data uložena v transakčních databázích, klíče musí být identifikovány pro připojení dat ze zdrojů dat do podoby jednoduchých tabulek. Projekty mohou selhat ještě před jejich začátkem, pokud klíče neexistují v tabulkách potřebných pro vybudování dat.

Problémy v nasazení

Predikátové modelové projekty mohou selhat kvůli překážkám v části nasazení modelu. Modely samotné nejsou výpočetně velmi komplikované, často potřebují pouze stovky nebo tisíce násobků či součtů, které jsou lehkou zvládnutelné dnešními servery. Avšak na základní úrovni musí být modelu umožněno být dotazován operačním systémem a vydávat predikce konzistentní s tímto systémem. Tohle typicky znamená, že model musí být zakódován v programovacím jazyce, který může systém zavolat (SQL, C++, ...). Pokud takto model nemůže být přeložen nebo se přeloží špatně, model je nepoužitelný.

Po prostudování tohoto tématu jsem se o implementaci nepokusil, jelikož se mi během doby scrapování vyskytly problémy v databázi s uloženými daty. Při opravě jednoho z kódů jsem si nevšiml chyby v ukládání dat do databáze a při každodenní aktualizaci se mi data přepisovaly. To způsobilo nepořádek v ID záznamů a některá data odkazovala na špatné nemovitosti. Tento problém se mi podařilo vyřešit, v databázi však ale momentálně nemám dostatečný počet záznamů pro predikce. Po delším sbírání dat a jejich studování by bylo možné do aplikace tuto predikci implementovat. Uvažoval jsem například nad predikcí vývoje ceny nemovitostí, který by byla zajímavým obohacením mé aplikace.

[13]

8 Závěr

Výsledkem mé bakalářské práce je funkční webová aplikace schopna analýzy nahromaděných dat a následné vizualizace. Uživateli je umožněno zobrazit si potřebná data a zjistit různé informace. Při správném použití může aplikace pomoci uživateli koupit nemovitost ve správnou dobu za nejnižší cenu. Zároveň také umožňuje uživateli získat přehled o situaci na trhu nemovitostí v Olomouckém kraji.

Do budoucna bych rád obohatil aplikaci o predikce vývoje ceny nemovitostí a také bych rád doplnil více statistik. Také by bylo možné rozšířit seznam scrapovaných stránek a zvětšit tak množství dat, se kterými aplikace pracuje.

Při vypracování této práce jsem se naučil používat spoustu nových technologií a nabyl jsem cenné zkušenosti, které budu moci aplikovat v budoucnu.

9 Conclusion

The result of my bachelor thesis is working web application capable of analysing stored data and subsequent visualisation. User can view needed data and find out different sorts of information. When correctly used the application can help user with buying a real estate at the right time for the lowest price. At the same time it allows user to learn about real estate market in region of Olomouc.

In the future I would like to add predictions of real estate price and also add more statistics.

During elaboration of this thesis I learned how to use new technologies and gained valuable experience that I can apply in the future.

10 Obsah přiloženého CD/DVD

Zde je uveden stručný popis

doc/

Text práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech příloh, a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu textu (v ZIP archivu), tj. zdrojový text textu, vložené obrázky, apod.

src/

Kompletní zdrojové texty webové aplikace se všemi potřebnými zdrojovými texty, knihovnamí a dalšími soubory potřebnými pro bezproblémové vytvoření adresářové struktury pro zkopírování na webový server.

readme.txt

Instrukce pro nasazení webové aplikace na webový server, včetně všech požadavků pro její bezproblémový provoz, a webová adresa, na které je aplikace nasazena pro účel testování při tvorbě posudků práce a pro účel obhajoby práce.

Bibliografie

- [1] MITCHELL, Ryan E. Web scraping with Python: collecting more data from the modern web. Second edition. Sebastopol, CA: O'Reilly Media, 2018. ISBN 1491985577.
- [2] ŘEZANKOVÁ, Hana, Tomáš LÖSTER a Zdeněk ŠULC. Úvod do statistiky. Vydání 2. přepracované. Praha: Oeconomica, nakladatelství VŠE, 2019. ISBN 978-80-245-2301-9.
- [3] GARCIA-MOLINA, Hector, Jeffrey D. ULLMAN a Jennifer WIDOM. Database systems: the complete book. 2nd ed. Upper Saddle River, N.J.: Pearson Prentice Hall, c2009. ISBN 978-0131873254.
- [4] AMOS, David, Dan BADER, Joanna JABLONSKI a Fletcher HEISLER. Python Basics: A Practical Introduction to Python 3: Real Python. 4th edition. -: Real Python, 2021. ISBN 1775093328.
- [5] Our Documentation | Python.org. Welcome to Python.org [online]. Copyright ©2001 [cit. 24.05.2021]. Dostupné z: <https://www.python.org/doc/>
- [6] Dash Documentation & User Guide | Plotly. Dash Documentation & User Guide | Plotly [online]. Dostupné z: <https://dash.plotly.com/>
- [7] Welcome to Flask — Flask Documentation (2.0.x). [online]. Dostupné z: <https://flask.palletsprojects.com/en/2.0.x/>
- [8] Forms — WTForms Documentation (2.3.x). [online]. Dostupné z: <https://wtforms.readthedocs.io/en/2.3.x/forms/>
- [9] (PDF) Web Scraping. ResearchGate | Find and share research [online]. Copyright © 2008 [cit. 24.05.2021]. Dostupné z: https://www.researchgate.net/publication/317177787_Web_Scraping
- [10] 27 Real Estate Statistics You Should Know and Understand. WordPress Real Estate - IDX, RETS, CRM, DDF and VOW solutions [online]. Copyright © 2021 Realtyna [cit. 25.05.2021]. Dostupné z: <https://realtyna.com/blog/27-real-estate-statistics-know-understand/>
- [11] Best Practices For Web Scraping. Zyte (formerly Scrapinghub) #1 Web Scraping Service [online]. Copyright © Zyte Limited [cit. 25.05.2021]. Dostupné z: <https://www.zyte.com/learn/web-scraping-best-practices/>
- [12] What is data visualization? A definition, examples, and resources. Business Intelligence and Analytics Software [online]. Copyright © 2003 [cit. 25.05.2021]. Dostupné z: <https://www.tableau.com/learn/articles/data-visualization>
- [13] ABBOTT, Dean. Applied predictive analytics: principles and techniques for the professional data analyst. Hoboken, NJ: Wiley, [2014]. ISBN 978-1-118-72796-6.