

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství (PEF)



Bakalářská práce

Webová aplikace na výpočet ceny nemovitosti

Paul Pešel

© 2023 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Paul Pešel

Informatika

Název práce

Webová aplikace na výpočet ceny nemovitosti

Název anglicky

Web Application for Real Estate Price Calculation

Cíle práce

Cílem bakalářské práce je vytvoření webové aplikace, která bude umět vypočítat cenu nemovitosti v ČR pomocí API od Valuo a ze zadaných údajů, kterými jsou povinné a upřesňující údaje, vrátí cenu nemovitosti. Za účelem splnění stanoveného cíle bude vytvořen design, frontend a backend aplikace, s využitím PHP frameworku Symfony.

Metodika

Provedte analýzu dostupných zdrojů z problematiky odhadu ceny nemovitostí a zpracujte formou literární rešerše.

Na základě výsledků analýzy vytvořte webovou aplikaci pro výpočet ceny nemovitostí s využitím aplikačního rozhraní firmy Valuo.

Doporučený rozsah práce

30 – 40 stran

Klíčová slova

cena nemovitostí, Symfony, PHP, API, webová aplikace

Doporučené zdroje informací

KOSEK, J. *PHP – tvorba interaktivních internetových aplikací : podrobný průvodce*. Praha: Grada, 1999. ISBN 80-7169-373-1.

MICHÁLEK, M. *CSS: Moderní layout*. Martin Michálek – VzhůruDolů.cz, 2022. ISBN 978-80-88253-07-5.

Předběžný termín obhajoby

2022/23 LS – PEF

Vedoucí práce

Ing. David Buchtela, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 7. 3. 2023

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 13. 3. 2023

doc. Ing. Tomáš Šubrt, Ph.D.

Děkan

V Praze dne 13. 03. 2023

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Webová aplikace na výpočet ceny nemovitosti" jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 15.3.2023

Poděkování

Rád bych touto cestou poděkoval Ing. Davidu Buchtelovi, Ph.D., za věcné připomínky, odborné rady a pomoc během zpracování bakalářské práce. Velké poděkování patří především Ing. Radku Šiterovi za poskytnutí API k aplikaci Valuo.

Webová aplikace na výpočet ceny nemovitosti

Abstrakt

Autorova bakalářská práce se zabývá vytvořením webové aplikace pro výpočet ceny nemovitosti v České republice. Aplikace umožňuje uživatelům zadat informace o nemovitosti, kterými jsou například velikost, umístění, stav a na základě těchto dat vypočítává cenu prodeje či pronájmu dané nemovitosti. V teoretické části je popsána analýza a problematika webových aplikací zabývajících se výpočtem ceny nemovitostí. Dále teoretická část obsahuje seznámení s technologiemi, které byly pro vývoj a implementaci této aplikace použity, přičemž bude kladen důraz na API od firmy Valuo, přes kterou je výpočet ceny nemovitosti realizován. Výsledná práce slouží široké veřejnosti se zájmem o realty hledající rychlý a spolehlivý způsob k získání přibližné ceny prodeje nebo pronájmu nemovitosti. Aplikace umožňuje uživatelům snadno a intuitivně zadat informace o nemovitosti, a následně zobrazí vypočtenou cenu. Pro uživatele, kteří mají další dotazy, je k dispozici kontaktní formulář umožňující snadné spojení s autorem aplikace. Tato bakalářská práce přináší užitečný nástroj pro širokou veřejnost, který umožňuje snadno a rychle získat přibližnou cenu nemovitosti na základě vstupních údajů.

Klíčová slova: kalkulátor nemovitostí, cena nemovitostí, webová aplikace, Symfony, PHP, API, Twig, CSS, Webpack

Web Application for Real Estate Price Calculation

Abstract

The author's bachelor thesis deals with the development of a web application for calculating the price of real estate in the Czech Republic. The application allows users to enter information about the property, such as size, location, condition and based on this data calculates the price of sale or rent of the property. The theoretical part describes the analysis and problems of web applications dealing with the calculation of real estate prices. Furthermore, the theoretical part includes an introduction to the technologies that were used for the development and implementation of this application, with emphasis on the API from Valuo, through which the calculation of the property price is implemented. The resulting work serves the general public interested in real estate looking for a fast and reliable way to obtain an approximate price for selling or renting a property. The application allows users to easily and intuitively enter information about the property, and then displays the calculated price. For users who have further questions, there is a contact form that allows easy connection with the author of the application. This bachelor thesis provides a useful tool for the general public to easily and quickly obtain the approximate price of a property based on the input data.

Keywords: real estate calculator, property price, web application, Symfony, PHP, API, Twig, CSS, Webpack

Obsah

1 Úvod	11
2 Cíl práce a metodika	12
2.1 Cíl práce.....	12
2.2 Metodika.....	12
3 Teoretická východiska	13
3.1 Kalkulátor nemovitosti.....	13
3.1.1 Výpočet ceny.....	13
3.1.2 Využití pro realitní kanceláře.....	14
3.2 Webová aplikace.....	15
3.3 Front-end a back-end.....	16
3.4 Front-end.....	16
3.4.1 Twig.....	16
3.4.2 SCSS.....	17
3.4.3 JavaScript.....	18
3.4.4 Webpack.....	18
3.5 Back-end.....	19
3.5.1 PHP.....	19
3.5.2 Symfony framework.....	20
3.5.3 API.....	20
3.6 Persony.....	21
3.7 Uživatelské cíle.....	22
3.8 Případy užití.....	22
3.9 Scénář.....	23
3.10 Drátěný model.....	23
4 Vlastní práce	24
4.1 Persony.....	24
4.1.1 Persona typu A.....	24
4.1.2 Persona typu B.....	25
4.1.3 Persona typu C.....	25
4.2 Uživatelské cíle.....	26
4.3 Tvorba případů užití, scénářů a drátěných modelu.....	27
4.3.1 Záhloví.....	27
4.3.2 Úvodní obrazovka.....	28
4.3.3 O kalkulátoru.....	29
4.3.4 Kalkulátor.....	30
4.3.5 Modální okno.....	32

4.3.6	Kontakt	34
4.4	Úvod do webové aplikace	35
4.4.1	Valuo API	36
4.4.2	Google Maps.....	38
4.5	Zpracování formulářů	39
5	Výsledky a diskuse	41
5.1	Spuštění aplikace.....	41
5.2	Testování	41
6	Závěr	42
7	Seznam použitých zdrojů	44
8	Seznam obrázků a zkratk.....	48
8.1	Seznam obrázků	48
8.2	Seznam použitých zkratk	48
9	Přílohy	50
9.1	Příloha 1 – Drátěný model a prototyp webové aplikace	50

1 Úvod

V dnešní době každý v určité fázi svého života přijde do styku s nemovitostí, ať už při řešení nákupu, prodeje nebo jejího pronájmu. Sám jsem se ve svém okolí setkal s názory o nedostatku povědomí o reálné ceně nemovitosti nebo pronájmu, kdy je třeba zvážit závislost na různých faktorech. Z tohoto důvodu jsem se rozhodl tento stav změnit a vytvořit webovou aplikaci na výpočet ceny nemovitosti. Dalším z důvodů, který mě na toto téma přivedl je fakt, že v digitální agentuře, kterou vlastním, jsme pro klienta z oboru realit podobný kalkulátor vyvíjeli. Nicméně uživatel byl v posledním kroku klientovy aplikace nucen vyplnit kontaktní údaje ke zjištění cenového odhadu nemovitosti, a to z toho důvodu, že nástroj sloužil především k získávání potencionálních zákazníků, jež byli následně kontaktováni realitní kanceláří.

Vzhledem k tomu, že mě osobně reality zajímají a sám nerad zveřejňuji své osobní údaje na internetu, mrzelo mě nedostatečné množství kalkulátorů nevyžadujících zadávání osobních údajů v České republice. Takto vznikla prvotní myšlenka vytvořit webovou aplikaci na výpočet ceny nemovitosti zdarma a bez nutnosti vyplňovat osobní údaje, kterou jsem následně rozpracoval jako zadání své bakalářské práce.

K úspěšnému fungování webové aplikace bylo zapotřebí získání API klíče umožňujícího provádění výpočtu, a proto jsem po dokončení klientova kalkulátoru kontaktoval majitele firmy Valuo zpracovávající odhad ceny nemovitostí. Majitel společnosti Valuo, jehož jsem požádal o vygenerování API klíče pro mou bakalářskou práci, s mým požadavkem souhlasil a díky tomu jsem mohl začít vytvářet webovou aplikaci, která na základě několika údajů o nemovitosti dokáže spočítat průměrnou cenu prodeje nebo pronájmu předmětné nemovitosti, a to zadarmo bez nutnosti vytváření uživatelského účtu.

Webová aplikace na výpočet ceny nemovitosti pomůže široké veřejnosti při styku s nemovitostí ke správnému rozhodnutí, ať už jím je prodej, nákup, pronájem nebo pouze rozšíření informovanosti o hodnotě dané nemovitosti.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem bakalářské práce je vytvořit webovou aplikaci, která bude sloužit pro výpočet ceny nemovitosti v České republice. Tento výpočet bude založen na údajích získaných z API od společnosti Valuo, která poskytuje aktuální tržní ceny nemovitostí. Aplikace bude umožňovat zadání povinných údajů, jako jsou například druh nemovitosti, adresa nemovitosti, plocha, stav nemovitosti a další. Výsledkem práce bude plně funkční webová aplikace, která bude umět zadané údaje zpracovat a vrátit uživateli výslednou cenu pronájmu či prodeje nemovitosti. Pro dosažení tohoto cíle bude vytvořen návrh aplikace, který zahrnuje prototyp v aplikaci Figma, front-end a back-end aplikace, a to s využitím PHP frameworku Symfony. K dokončení projektu autor využije i technologie Webpack, SCSS, Twig a jQuery. Dalším cílem práce bude snadná použitelnost a uživatelská přívětivost aplikace a nemálo důležité bude také zajistit její správnou funkčnost.

V souhrnu je cílem bakalářské práce vytvoření webové aplikace, jež bude schopna poskytovat uživatelům spolehlivé a přesné informace o ceně nemovitosti v České republice, a to prostřednictvím využití API od společnosti Valuo a vlastních programátorských schopností autora.

2.2 Metodika

Metodika bakalářské spočívá v zaměření se na analýzu dostupných zdrojů z problematiky odhadu ceny nemovitostí v případě první, teoretické, části bakalářské práce. Získané informace budou zpracovány formou literární rešerše. Na základě výsledků analýzy bude vytvořena webová aplikace pro výpočet ceny nemovitostí s využitím aplikačního rozhraní firmy Valuo.

Druhá, praktická, část práce, bude postavena na poznacích získaných v teoretické části. Nejprve bude přiblížena tvorba drátěného modelu (skica webu) v programu Figma, ze kterého se dále pokračuje na prototyp a samotný vývoj celé aplikace postavený na PHP frameworku Symfony. Front-end bude tvořen z Twig šablon a stylování pomocí SCSS, které dále bude zpracovávat Webpack modul. Skripty budou psány v jQuery.

V závěru autor provede zhodnocení naplnění cíle bakalářské práce.

3 Teoretická východiska

3.1 Kalkulátor nemovitosti

On-line kalkulátor nemovitosti je zpravidla zdarma poskytovaná webová aplikace, která umožňuje zjistit cenu prodeje či pronájmu nemovitosti z údajů poskytnutých uživatelem. V praxi se jedná především o parametry typu, kterými jsou byt nebo rodinný dům, lokality, stavu a velikosti nemovitosti. (1)

Je nutné vzít v potaz skutečnost, že výsledné hodnoty získané z kalkulátoru jsou pouze orientační. Takto získané hodnoty slouží především uživatelům k vytvoření představy o ceně prodeje či pronájmu nemovitosti. Před samotným prodejem případně pronájemem je však obecně doporučováno cenu a podmínky prodeje či pronájmu konzultovat s odborníky a profesionály, kteří cenu dokáží určit přesněji, čímž dojde k zamezení možného nechtěného podhodnocení nemovitosti. V opačném případě může nastat situace, kdy budou mít uživatelé nereálné představy o vysoké hodnotě, jelikož kalkulátor vypočítá výslednou hodnotu jako nepřiměřeně vysokou. Zveřejněním nabídky vycházející ze zjištěné nepřiměřeně vysoké ceny může dojít k nedostatečné poptávce. (2)

3.1.1 Výpočet ceny

Prvotní výpočet je u kalkulátorů velmi podobný. Webové aplikace na výpočet ceny nemovitosti poskytují možnost vypočítat obecnou cenu, jejíž vypočtení umožňuje i autorova webová aplikace, závislou na základních proměnných, a některé webové aplikace pro zpřesnění výsledné hodnoty uživatelům umožňují vypočítat i cenu blíže specifikovanou, která je závislá na větším množství proměnných. Cena, ať už obecná nebo blíže specifikovaná, je vypočítána z reálných cen prodejů evidovaných v katastru nemovitostí, cenových mapách a údajů z inzerčních portálů či umělé inteligence. (2)

Evidence ceny prodejů jednotlivých nemovitostí nebyly vždy shromažďovány a zveřejňovány v katastru nemovitostí, stalo se tak až účinností zákona č. 256/2013 Sb., zákon o katastru nemovitostí (katastrální zákon) ke dni 1. 1. 2014. Cena se do katastru nemovitostí vkládá dle ceny uvedené v kupní smlouvě. (3)

Kupní ceny jednotlivých pozemků je možné z katastru nemovitostí získat vyplněním elektronické žádosti, což je uživateli umožněno až po přihlášení prostřednictvím například

Identity občana. V případě, že má uživatel zájem žádat o výpis pod názvem „*Údaje o dosažených cenách nemovitostí podle nemovitostí*“ či o výpis „*Údaje o dosažených cenách nemovitostí podle řízení*“, je získání těchto výpisů, prostřednictvím kterých uživatel získá potřebné informace, zpoplatněno 50,- Kč. Pokud by uživatel zvolil jiný způsob podání žádosti, kterým je podání přímo na katastrálním úřadu nebo na CzechPointu, jsou ceny vyšší. (4) Vyšší cena je zejména dána administrativní zátěží, která je spojena s přímým kontaktem správního orgánu a nutností vyplnění žádostí a jiných potřebných dokumentů.

Další možností výpočtu odhadu ceny nemovitosti jsou cenové mapy, jež vycházejí z cen vložených do katastru nemovitostí, zmíněných výše. Výhodou cenové mapy je její přehlednost zobrazování formou běžné mapy ve spojení s uváděním ceny za m². V této souvislosti stojí za zmínku služba Reas.cz spočívající mimo jiné v možnosti zobrazení atlasu cen, cenové mapy, jež má dle slov služby Reas.cz zůstat trvale bez poplatků. Služba Reas.cz bezpochyby usnadní mnohým, kteří o prodeji nemovitosti přemýšlejí, proces cenového odhadu a mohlo by tak dojít k nárůstu prodejů či pronájmů nemovitostí na českém trhu. (5)

3.1.2 Využití pro realitní kanceláře

Možnost vložení on-line kalkulátoru na web je pro realitní kanceláře a realitní makléře skvělým marketingovým nástrojem, který může při správném nastavení přinést realitní kanceláři mnoho klientů majících vážný zájem o koupi nebo pronájmů nemovitostí. (6)

Zpracovávání odhadu ceny nemovitostí je realitními makléři nabízeno s garancí profesionality v jejich portfoliu služeb, a to ve většině případů jako jedna z doplňkových služeb. Vzhledem k aktuálním cenám na trhu je tato doplňková služba realitními makléři běžně zpoplatněna mezi 4 až 5 tisíci Kč (7), v případě, že má klient zájem o ohodnocení nemovitosti izolovaně bez návaznosti na následný prodej, pronájem či nákup. Pokud se zájemce rozhodne využít podpory realitní kanceláře v průběhu celého procesu prodeje, pronájmů či nákupu, je ve většině případech tato zpoplatněná služba zcela zdarma. (8)

3.2 Webová aplikace

Webová aplikace je taková aplikace, která je zprostředkována internetovým prohlížečem. Výhodou oproti desktopového softwaru je skutečnost, že ji není potřeba interně instalovat do počítače, a proto je možné webové aplikace spustit z jakéhokoliv zařízení. U uživatelů dále benefitují z toho, že nezabírají žádné místo na disku. Další přednost, asi tu nejvýznamnější, spatřují uživatelé ve snadném přístupu k webové aplikaci, jelikož postačí pouze připojení k internetu a jakýkoliv webový prohlížeč. (9)

Najisto je potřeba postavit rozdíl mezi webovými aplikacemi a webovými stránkami, se kterými jsou často zaměňovány a jejichž rozdíly jsou zásadní. Jedním z prvních rozdílů je lišící se účel. Webové stránky mají především za účel propagaci firem, produktů či služeb, proto jsou také často veřejností známy pod pojmem prezentační stránky. Další rozlišení je možné spatřovat v míře interaktivity. Webová stránky mají zpravidla statický obsah a nízkou interaktivitu s uživatelem. Naproti tomu webové aplikace jsou stránky s vysokou mírou interaktivity, jelikož slouží uživateli ke zprostředkování chtěné akce. Příkladem takových webových aplikací jsou chatovací aplikace (Slack, Messenger), aplikace na sdílení dokumentů (Google Dokument, Microsoft Word), aplikace na řízení projektů (Asana, Jira), ale můžou jimi být i jednoduché stránky s kontaktním formulářem a další. (10)

Vývoj webových stránek a webových aplikací naopak příliš rozdílný není, protože je možné použít podobné technologie a v mnoha případech tak mají jako společného jmenovatele technologie HTML, CSS a JavaScript. Nicméně webové aplikace jsou v některých případech rozšířené o technologii PHP, díky níž je možné provádět složitější úlohy nebo využívat databázi k práci s daty. (11) Z tohoto důvodu je vývoj webových aplikací náročnější než tvorba webových stránek, a to nejen z hlediska času a financí, ale i programátorských dovedností. Při vývoji webových aplikací je významný prvek bezpečnosti, na který se klade větší důraz, a to především u aplikací pracujících s osobními a důvěryhodnými daty. (10)

Náročnost vývoje je kompenzována možností využití napojení na další externí systémy. Není tak potřeba vytvářet například vlastní fakturační systémy, ale v mnoha případech se využívá již hotová webová aplikace zprostředkovávající danou činnost, jež je na aplikaci napojována pomocí API. (10)

3.3 Front-end a back-end

Vývoj aplikací se dělí na front-end a back-end, což jsou pojmy patřící mezi ty nejnámější a nejpoužívanější v oblasti vývoje webových aplikací potažmo i webových stránek. Hlavní rozdíl mezi front-endovými a back-endovými vývojáři spočívá především v používaných technologiích. (12)

3.4 Front-end

Front-end je část webové aplikace či stránky viditelná běžným uživatelům, kteří skrze ni interagují. Někdy je front-end znám i pod dalšími názvy jako je například veřejně přístupná část nebo klientská strana aplikace. Tato část vývoje webové aplikace či stránky se zaměřuje především na kódování, vytváření prvků a funkcí webové aplikace či stránky. (12) Front-end má během vývoje na starosti front-endový vývojář, jehož cílem je dát prvkům webové aplikace či stránky vizuální podobu. Mezi jeho další úkoly patří zajištění hladkého fungování webové aplikace či stránky, takovým způsobem, aby byla komunikace s uživateli, pokud možno bezproblémová. Je třeba poukázat i na úkoly, které již nejsou v gesci front-endového vývojáře, ale úzce s jeho prací souvisí, jako je vytváření samotné grafiky, kterou má na starosti UI a UX designér. Spolupráce UI a UX designéra a front-end vývojáře je velmi důležitá a klíčová, jelikož front-endový vývojář postupuje dle vytvořené grafiky a na základě ní vytváří a styluje prvky, čímž dává samotnému webu život. (13)

Základními technologiemi pro front-end jsou především HTML, CSS a JavaScript. Z těchto technologií vychází další frameworky a knihovny, se kterými se v mnoha případech pracuje jednodušeji. Ukázkový výčet odvozených technologií s jejich bližší charakteristikou poskytuje autor níže. (12)

3.4.1 Twig

Twig je flexibilní, rychlý a bezpečný šablonový systém pro technologii PHP. Tento šablonový systém je využíván v mnoha Open-Source projektech jako například Symfony, Drupal8, Matomo a mnoho dalších frameworků ho podporuje – například Laravel či Yii. (14).

Podstatou Twigu je mimo jiné vytváření šablon, které vývojářům usnadňují práci, zejména zjednodušují orientaci v kódu webové aplikace či stránky. Šablony vytvořené

v Twigu mají koncovku `.twig`. Pro maximální využití potenciálu technologie Twig je obecně doporučováno do editoru kódu nainstalovat rozšíření podporující tuto technologii.

Vývojářům pro práci v Twigu zásadně stačí znalost pouze 3 základních syntaktických konstrukcí:

- `{{ ... }}` – používá se například pro vykreslení proměnné
- `{% ... %}` – používá se v případě aplikace logiky, kterou je například definování proměnné, zápis podmínky, vytvoření *for* cyklu nebo importování další šablony
- `{# ... #}` – používá se pro komentování kódu (15)

Instalace Twigu je doporučována přes Composer, nástroj pro správu knihoven a jiných zdrojů v PHP, za použití příkazu `composer require twig/twig`, který nainstaluje nejaktuálnější bezpečnou verzi. (14)

3.4.2 SCSS

SCSS je často představováno jako vylepšené a pokročilejší CSS, které jsou dále známé pod označením kaskádové styly. Hlavním využitím kaskádových stylů je stylování vzhledu webové aplikace či stránky. Dále se také kaskádovými styly upravuje responzivita webu, nebo-li způsob zobrazování webové aplikace či stránky v závislosti na typu a velikosti zařízení uživatele. V praxi se pro responzivitě webových aplikací či stránek používá koncept Mobile First, který vývojářům stanovuje způsob stylování webové aplikace či stránky s upřednostněním mobilního zařízení, jež je dále z mobilních zařízení rozšířeno na desktopovou verzi. (16)

Sass/SCSS a LESS je preprocesor jazyka kaskádových stylů CSS. Preprocesor je mezikód následně překládaný pomocí Compileru do cílového jazyka, kterým je v tomto případě jazyk kaskádových stylů CSS. Překlad do kaskádových stylů CSS se uskutečňuje, jelikož webové prohlížeče umí zpracovávat pouze styly v syntaxi CSS. (17) Mezi hlavní výhody využití preprocesorů, vzhledem k relevanci tématu zmiňuje autor preprocesor SCSS, neboli Sassy CSS (18), patří výrazné urychlení vývoje a celková přehlednost kódu. Preprocesory umožňují zejména definování všech klíčových vlastností pro určité třídy jazyka CSS na jediném místě, čímž dojde k výraznému zkrácení šablony stylů. Dalšími

benefity jsou například zanořování (Nesting), vytváření funkcí, početních operací a podmínek. (17)

3.4.3 JavaScript

JavaScript patří mezi tři nejpoužívanější technologie, HTML, CSS, JS, které slouží vývojářům pro tvorbu webových aplikací či stránek. Frekvence používání je dána především jejich dostupností na téměř všech počítačích po celém světě. Jedná se o skriptovací jazyk pro webové aplikace či stránky, jež si časem nacházel uplatnění ve validaci formulářů a manipulaci s obsahem prvků na stránce. Dnes se v něm dají vytvářet i pokročilé a složité dynamické webové aplikace, které jsou náročné na vývoj. (19)

Programům, které jsou psané v JavaScriptu se říká skripty a vkládají se přímo do HTML kódu, což dříve prohlížeče nepodporovaly. Proto se v minulosti, kdy byl ještě trendem Adobe Flash, JavaScript potýkal s nedostatečnou kompatibilitou webových prohlížečů, což mělo za následek, že se webové stránky nezobrazovaly správně. (20)

Nyní je však situace rozdílná a JavaScript je na vzestupu. Aktuálně je již součástí většiny webů a jeho popularita stoupá čím dál více především díky frameworkům, jako je Next.js, React, Vue nebo Angular, které z JavaScriptu vychází. (20)

3.4.4 Webpack

Webpack je v poslední době často používanou technologií, která se ne nadarmo srovnává s nástroji jako je Gulp či Grunt. (21) Je to nástroj pro zpracovávání souborů usnadňující vývojářům práci. Webpack používá JavaScriptové moduly a vytváří balíčky pro prohlížeče, což je blíže vysvětleno níže. Umí však pracovat i s obrázky či styly, a to díky pluginům, které se do Webpacku dají importovat. (22)

Modulem se ve Webpacku rozumí kus kódu. V případě, že jde o JavaScript, tak dokáže Webpack z vícero skriptů udělat jeden samotný balíček (Bundle), který je v základním nastavení pojmenovaný jako App.js. (21)

Jak již bylo zmíněno, tak pomocí Webpacku je možné zpracovávat či kompilovat i styly z preprocesoru SCSS, a to díky dostupnosti velkého množství pluginů pro tuto moderní technologii. (21)

3.5 Back-end

Back-end je na rozdíl od front-endu webovou částí, která pro běžné uživatele není vidět, a je zpracovávána serverem nikoliv klientem, jako je tomu u front-endu. Tato část webové aplikace je zodpovědná za zpracovávání dat a zajišťuje celkovou logiku aplikace. (23)

V praxi je důležité, aby obě části vývoje, front-end a back-end, spolu správně komunikovaly. Pro představu je možné využití back-endu ilustrovat například na e-shopu. E-shop je složitější dynamická webová aplikace, která s back-endem musí pracovat, jinak by se na webu uživateli nezobrazovaly produkty, jejich ceny, název a další informace o produktech, jež jsou na webu zobrazovány z databáze. (24)

Back-end vývojář má tedy na starosti kompletní zpracovávání dat, práci s databází a celou část webové aplikace, kde není statický obsah (neměnný). V mnoha případech je tato práce náročnější nejen z pohledu potřebného většího množství znalostí, ale i z časového hlediska. Tyto důvody vedou k lepšímu finančnímu ohodnocení vývojářům působících na pozici back-end vývojáře. Nejčastěji se v praxi setkáme s tím, že jsou front-end a back-end vývojáři specializovaní pouze na jedno z těchto dvou odvětví a v něm se poté více realizují. Profesionálům, kteří rozumí oběma částem vývoje, front-endu i back-endu, se říká Full-stack vývojáři. (25) Co se týče jazyků, s nimiž se Back-end vývojáři setkávají, jedná se o jazyky jako je Ruby, Python, Java nebo PHP. (26)

3.5.1 PHP

PHP je jednou z nejvíce používaných technologií pro webové aplikace, což je viditelné na míře používání, kdy s ní pracuje více než 70 % webů po celém světě. Za poslední léta se PHP drží i mezi nejvíce používanými programovacími jazyky. Velká obliba této technologie je dána především jejím využitím pro tvorbu webových aplikací, její rychlostí a bezpečností. Mimo jiné se PHP řadí mezi jeden z nejsnazších programovacích jazyků, se kterými se dá v oblasti IT začít. Přesto, že je PHP doporučováno začátečníkům, je třeba myslet na to, že profesionálové obecně radí mít základy HTML. (27)

Programovací jazyk PHP je využíván pro vývoj v back-endové části webové aplikace. Tento jazyk je používán v tak vysoké míře především díky tomu, že v něm jde naprogramovat takřka cokoliv a díky dalším výhodám zmíněným výše. PHP dokáže

zpracovávat data z formulářů, vygenerovat dynamický obsah pro stránku, pracovat s databází, odesílat e-maily nebo také šifrovat hesla. Právě díky jeho velké flexibilitě se stal populárním programovacím jazykem napříč celým světem. (28)

Oblíbenost a frekvenci využití PHP dokazuje i jeho použití těmi největšími technologickými giganty, kterými jsou například firmy Facebook, Slack nebo Lyft. Dále je také využíván v CMS (Systém pro správu obsahu), z nichž jsou pravděpodobně nejznámější WordPress nebo Drupal. (29)

3.5.2 Symfony framework

Symfony je kompletní PHP framework pro webové aplikace, který je tvořen sadou PHP komponent, jež tvorbu webových aplikací výrazně zrychlují. (30) Frameworky jsou ucelené soubory do sebe pasujících knihoven. Nutno říci, že se u větších projektů programovaných v PHP vývojář bez kvalitních knihoven neobejde. Důvodů, proč tomu tak je, je několik. Ačkoliv je PHP velmi pokročilý jazyk, přesto se v něm najdou nedostatky spočívající zejména v absenci mnoha důležitých funkcí nebo se s nimi špatně pracuje. Díky kvalitním komponentám z frameworku se pak webová aplikace dá napsat výrazně rychleji a přehledněji, především nemusí vývojář opakovaně a zbytečně řešit rutinní záležitosti a vyvíjená aplikace pak bude i lépe udržovatelná. (31)

Symfony framework je jeden z nejznámějších a nejvíce používaných PHP frameworků. Výběr kvalitního frameworku pro vývoj aplikace je zásadní, a proto tento framework používá již více než tisíce webových aplikací. (30)

3.5.3 API

API je zkratka Application Programming Interface, která se v překladu do češtiny dá nazvat jako Rozhraní pro programování aplikací. API se používá velmi často a je běžnou součástí mnoha webových aplikací. Příkladem její implementace jsou například webové aplikace, na kterých se dá zjistit počasí nebo měnový kurz. (32)

API se dá nazvat jako prostředník, který pracuje s daty z jiných aplikací. (32) Obecně se však API vysvětluje pouze s obtížemi, ale Petr Smejkal ve svém článku *Co je API a k čemu slouží?* z 25. 4. 2021 rozumně a ilustrativně vysvětluje, jak API funguje: „...*tak si představte API jako číšníka. Vy jako návštěvník webu budete jako zákazník restaurace a server bude kuchyně. Na webových stránkách budete chtít třeba zjistit jaké je počasí a proto*

si zavoláte číšníka. Řeknete mu, že si přejete zjistit jaké je počasí v Brně. To, že mu to řeknete tak vytvoříte požadavek neboli v angličtině "request". Číšník s tímto požadavkem půjde do kuchyně (server) a získá počasí pro oblast Brno neboli "response" a tu vám donese a díky tomu víte, že bude slunečno.“ (33)

Vývojář k ověření funkčnosti API často provádí její testování. Výsledky vrácené z API jsou testovány nejčastěji prostřednictvím aplikace Postman. Díky této aplikaci se dají vytvářet různé sady s metodami a je možné testovat, zda API vrací očekávaný výsledek. Všechna API mají však své vlastní dokumentace, kterými se vývojář musí řídit a najde v nich další užitečné informace, zejména třeba metody nebo hodnoty, které by API měla vrátet. (33)

3.6 Persony

Persony jsou fiktivní postavy, které se vytváří na základě průzkumu. Persony reprezentují uživatele, kteří by mohli daný produkt či službu potencionálně využívat, proto je jejich správné profilování klíčové. Návrh person slouží a pomáhá především návrhářům a marketérům porozumět potřebám, názorům, chováním a cílům daných uživatelů. Na základě vytvořených person potom návrhář vytváří ideální uživatelský požitek pro danou cílovou skupinu. (34)

K vytváření person je především potřeba znát zákazníky a jejich potřeby. Bližší specifikaci o reálných zákaznících si opatří návrhář a marketéři z dat získaných během uživatelských rozhovorů, dotazníkových šetření či pozorování nebo využitím některých dalších statistických metod. (35)

Při tvorbě person se využívají tři typy a to:

- **Persona A:** Jedná se o typického a hlavního uživatele, který bude využívat všechny funkcionality a systém je pro něj musí implementovat. Zejména pro tento typ persony bude vznikat i prototyp systému.
- **Persona B:** Tento typ uživatele není primární a jedná se tedy o raritního uživatele, který službu využívá spíše ze zájmu. Ačkoliv systém není navržený pro tuto personu, tak by i přesto měl zajistit možnost naplnění jeho potřeb a očekávání.

- **Persona C:** Jde o naprostý opak primárního uživatele. Jedná se o uživatele, který systém nebude nikdy využívat, proto ho marketéři a návrháři při realizaci systému neberou v potaz. (35)

3.7 Uživatelské cíle

Uživatelské cíle, neboli User goals, vznikají na základě požadavků, jež mají uživatelé, kteří budou aplikaci využívat, na celkové fungování aplikace. Uživatelé jsou pak blíže definovány personami, které jsou autorem popisovány výše. (35)

Především je při optimalizaci uživatelského cíle důležité, aby uživatelé dosáhli daného cíle co nejjednodušším a nejrychlejším možným způsobem. (35)

3.8 Případy užití

Případ užití, anglicky se používá termín Use case, je metoda popisující sekvenci kroků vykonávaných uživatelem při využívání interagujícího systému nebo při plnění konkrétních úkolů. V případě této bakalářské práce se zejména jedná o využívání webové aplikace. Využití případů užití je nejvíce frekventováno především v oborech jako HCI (známe pod českým názvem Interakce člověka s počítačem) či interakční prototyp. (36)

Jak již bylo zmíněno výše, případ užití je složen z posloupnosti kroků, jež uživatel vykonává ke splnění daného úkolu. Pro popis těchto kroků je využíván specializovaný jazyk UML (Unified Modeling Language), který popisuje jednotlivé akce uživatele a na druhé straně reakce systému, a to formou diagramu. I když je popis jazykem UML obecně doporučován je možné celou interakci uživatele se systémem popisovat i slovně. (36)

Pro slovní popis zmiňovaných interakcí se v českém jazyce využívají slova jako očekává či požaduje. Příkladem takového slovního popisu je, že uživatel očekává přihlášení pomocí uživatelského jména nebo uživatel požaduje možnost změny hesla. (35)

První zmínky metody případů užití a její specifikace je možné pozorovat již roku 1986, kdy švédský výpočetní inženýr Ivar Jacobson, jako první, formuloval textové, strukturální a vizuální modely technik pro specifikování případů užití. Další významný moment pro Ivara Jacobsona nastal v roce 1992, kdy se stal spoluautorem knihy Object-Oriented Software Engineering. Tato kniha významně pomohla zviditelnit a zpropagovat

techniky, které se používají pro zjištění funkčních požadavků na vytvoření případů užití. (35)

3.9 Scénář

Za obecnou definici scénáře je veřejností přijímáno vysvětlení, že jde o odpověď systému na očekávané a požadované prvky uživatele, které byly definovány v případech užití. V HCI je ustálenou praxí psát, jak systém interaguje s prvky, které uživatel očekává či požaduje. Výsledná interakce s prvky je psána z pohledu systému a k popisu se používají slova jako zobrazí nebo vypíše. Příkladem popsané interakce může být, že systém zobrazí vstupní pole pro zadání uživatelského jména. (35)

3.10 Drátěný model

Drátěný model představuje základní kostru webové stránky či aplikace a je znám i pod anglickým názvem Wireframe. Úlohou drátěného modelu je naznačení struktury webu rozložením jejích prvků a dále rozdělení stránky wireframem na jednotlivé kontejnery neboli sekce jednotlivých stránek. (37)

S tvorbou wireframu a prototypu se často pojí pojmy UI a UX designer. Je však třeba tyto dvě profese přísně odlišovat, jelikož má každá z nich jinou úlohu. Tvorbou wireframu se má správně zabývat UX designer, který připravuje výše zmiňovanou strukturu webu tak, aby web obsahoval všechny požadované a očekávané prvky uživatelem. Následně drátěný model převezme UI designer, který na jeho základě připraví grafickou podobu. (37)

4 Vlastní práce

Na základě poznatků získaných v teoretické části může autor přejít k plnění hlavního cíle této práce, čímž je vývoj webové aplikace na výpočet ceny nemovitosti vycházející z analýzy dostupných zdrojů z problematiky odhadu cen nemovitostí. Mimo vývoj webové aplikace na výpočet ceny nemovitosti je cílem také vytvořit drátěný model a prototyp v aplikaci Figma. Druhý cíl je vytvořený dle požadovaných uživatelských cílů a person, které tuto aplikaci budou využívat.

4.1 Persony

4.1.1 Persona typu A

Jméno: František Franc

Věk: 40

Pohlaví: muž

Zájmy: cestování, jídlo, jízda na kole, běh, péče o zdraví

Typický den: František vstává přibližně v 7 hodin a jde si zaběhat trasu dlouho 5 km. Po ranním běhu se osprchuje a nasnídá. Následně se připraví na odjezd do práce, kam běžně dorazí kolem deváté hodiny. Jelikož pracuje na telefonní podpoře v IT společnosti, má psychicky velmi náročnou práci, neboť náplň jeho pracovní pozice spočívá v komunikaci s mnoha zákazníky. Obden po práci jezdí za svou maminkou, která uvažuje o přestěhování se do menšího bytu, a tak spolu hledají možné varianty nového bydlení. Večer se vrací domů vyčerpaný a tráví čas se svou rodinou v domku za Prahou.

Krátká historie: František vystudoval ekonomickou střední i vysokou školu, ačkoliv se po dokončení školy věnoval pouze IT. Celý život bydlel v Praze v domě se svou rodinou. Bohužel mu během studia na střední škole zemřel otec, tudíž vyrůstal dále pouze s maminkou. Ve 32 letech našel svou životní partnerku, se kterou po pár letech měli první miminko. Maminka Františka však zůstala bydlet sama ve velkém domě v Praze a přemýšlí, že by se přestěhovala do malého bytu, aby se ve stáří nemusela o celý dům starat sama. Uvažují tak nad prodejem domu a koupí nového bytu v Praze.

4.1.2 Persona typu B

Jméno: Vladimír Žitný

Věk: 50

Pohlaví: muž

Zájmy: matematika, čtení, kutilství, zvířata

Typický den: Vladimír vstává brzy ráno kolem 6. hodiny. Ráno chodí vyvenčit pejska, nasnídá se a připraví se na cestu do školy, ve které pracuje. Vladimír je učitelem matematiky na střední škole, kde tráví velkou část každého dne vždy přibližně do tří hodin odpoledne. Poté se vrací domů, kde se věnuje přípravě testů a materiálů pro své žáky, tráví čas se svou rodinou a stará se o domácnost. Večer chodí brzy spát, aby byl plný energie na další den.

Krátká historie: Vladimír vyrůstal v rodině učitelů, a tak se celý život profiloval, aby se jím stal i on. Vladimír má dvě děti a manželku, se kterými žije na okraji Prahy v bytě. Vladimír, protože má rád matematiku a čísla, je velmi systematický člověk, a proto se v rodině stará i o finance. Každý rok si tak dělá součet rodinného majetku, aby zjistil, zda si jako rodina od předešlého roku polepšily či nikoliv. Vladimír přemýšlí o koupi rodinného domu v přírodě a klidu za Prahou, kam by se v důchodu s manželkou a pejskem odstěhovali.

4.1.3 Persona typu C

Jméno: Ivana Housková

Věk: 18

Pohlaví: žena

Zájmy: škola, zábava, cestování, noční život

Typický den: Ivana vstává brzy ráno, upraví se, ale nestíhá se nasnídat, a proto ve spěchu odchází na zastávku, odkud jede městskou hromadnou dopravou do školy. Ve škole je každý den do odpoledne a po škole chodí se spolužačkami na kávu. Poté se vrací domů, kde se učí nebo dělá úkoly do školy, aby měla volný večer a mohla jít do centra Prahy, kde se většinou potká s kamarády a vyrazí za zábavou, nebo mohla sledovat oblíbené filmy.

Krátká historie: Ivana vyrůstala v Praze a později se s rodinou přestěhovala do domu kousek za Prahou, kde nyní žije i se svou malou sestrou. V současné době studuje ekonomickou střední školu, hlásí se na vysokou školu a také se připravuje na maturitní zkoušku. Ivana má v plánu během studií na vysoké škole setrvat bydlet u rodičů, aby ušetřila peníze a nemusela se starat o vlastní domácnost.

4.2 Uživatelské cíle

Na základě analýzy problematiky odhadu cen nemovitostí provedené v teoretické části této bakalářské práce a vytyčených typů person v úvodu sekce vlastní práce byly autorem vytvořeny tyto uživatelské cíle:

- možnost se v navigaci proklikat na jednotlivé sekce stránky
- na úvodní obrazovce mít možnost rychlého najetí kalkulátoru odhadu ceny nemovitosti
- zjistit informace o kalkulátoru
- zjistit informace o tom, jak kalkulátor funguje
- možnost si přečíst často kladené dotazy ke kalkulátoru ceny nemovitosti
- umožněné odeslání dotazu skrz kontaktní formulář

V kalkulátoru:

- spočítat cenu prodeje domu
- spočítat cenu pronájmu domu
- spočítat cenu prodeje bytu
- spočítat cenu pronájmu bytu
- možnost předvyplnit adresu během zadávání adresy nemovitosti

4.3 Tvorba případů užití, scénářů a drátěných modelu

Na základě uživatelských cílů byly specifikovány případy užití a scénáře, dle kterých byly navrženy drátěné modely a prototyp, jež autor bakalářské práce zhotovil v aplikaci Figma. Celý drátěný model a prototyp je přílohou bakalářské práce níže.

4.3.1 Záhlaví

Případ užití:

V záhlaví uživatel očekává:

- možnost dostat se na sekci O Kalkulátoru
- možnost dostat se na sekci Kontakt

V záhlaví uživatel požaduje:

- možnost dostat se na sekci Kalkulátoru

Scénář:

System v záhlaví vypíše všechny prvky, které uživatel očekává a požaduje. Tyto prvky vypíše horizontálně v pravé straně záhlaví. Samotné záhlaví je rozděleno do dvou částí, levé a pravé. V levé části systém zobrazí logo webové aplikace na kalkulaci nemovitosti. Toto logo zároveň slouží jako odkaz na domovskou stránku webové aplikace.

Záhlaví není statické a pro uživatele využívajícího zařízení od tabletů níže se změní, aby bylo uživatelsky přívětivé. Prvky, které uživatel očekává a požaduje budou zobrazeny až po kliknutí na tzv. hamburger tlačítko, které zobrazí všechny uživatelem očekávané a požadované prvky přehledně pod sebou v samostatné navigaci.

Drátěný model:



Obrázek 1 - Záhlaví webové aplikace

Prototyp:

KALKULÁTOR
NEMOVITOSTI

O KALKULÁTORU

KONTAKT

SPOČÍTAT CENU

Obrázek 2 - Záhleví webové aplikace

4.3.2 Úvodní obrazovka

Případ užití:

Na úvodní obrazovce uživatel očekává:

- krátký a výstižný nadpis webové aplikace

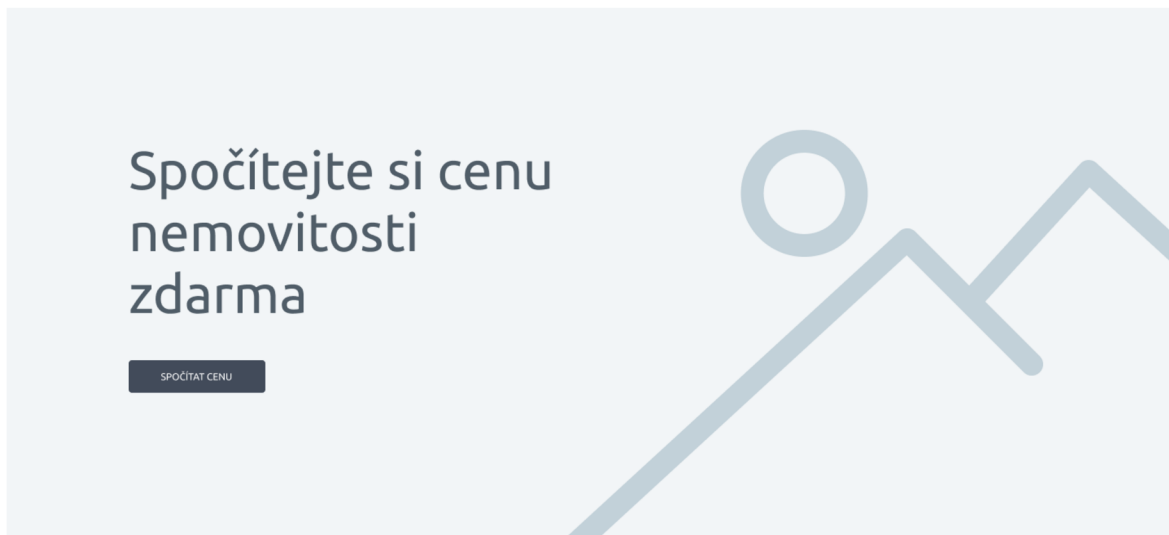
Na úvodní obrazovce uživatel požaduje:

- možnost dostat se na sekci Kalkulátoru

Scénář:

System na úvodní obrazovce pod sebou zobrazí prvky očekávané a požadované uživatelem. V pozadí úvodní obrazovky systém zobrazí fotografii překrytou barvou, aby byly uživatelem očekávané a požadované prvky dostatečně čitelné a viditelné.

Drátěný model:



Obrázek 3 - Úvodní obrazovka webové aplikace

Prototyp:



Obrázek 4 - Úvodní obrazovka webové aplikace

4.3.3 O kalkulátoru

Případ užití:

V sekci O kalkulátoru uživatel očekává:

- krátký popis o kalkulátoru
- často kladené dotazy a odpovědi na tyto dotazy

Scénář:

System v sekci o nás zobrazí všechny očekávané prvky. Sekce je rozdělená na dvě části, a to levou a pravou. V levé straně systém zobrazí nadpis a krátký popis o kalkulátoru. Zatímco na pravé straně systém zobrazí často kladené dotazy. Na menších zařízeních se tyto dvě části poskládají pod sebe. Odpovědi na jednotlivé dotazy se zobrazí až po kliknutí na dotaz a stejným způsobem, druhým kliknutím, může uživatel odpověď zavřít.

Drátěný model:

O kalkulátoru

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. incididunt ut labore.

Is this thing on? ▼

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. m dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor inc.

Is this thing on? >

Is this thing on? >

Is this thing on? >

Obrázek 5 - Sekce O kalkulátoru

Prototyp:

O KALKULÁTORU

Lorem ipsum dolor sit amet, consectetur adipiscing elit. In convallis. Quisque porta. Curabitur vitae diam non enim vestibulum interdum. Nam sed tellus id magna elementum tincidunt. Curabitur bibendum justo non orci. Morbi imperdiet, mauris ac auctor dictum, nisl ligula egestas nulla, et sollicitudin sem purus in lacus.

SPOČÍTAT CENU

— Jak funguje kalkulátor? ▲

Lorem ipsum dolor sit amet, consectetur adipiscing elit. In convallis. Quisque porta. Curabitur vitae diam non enim vestibulum interdum. Nam sed tellus id magna elementum tincidunt.

— Podle čeho je cena vypočítaná? ▼

— Podle čeho je cena vypočítaná? ▼

— Podle čeho je cena vypočítaná? ▼

— Podle čeho je cena vypočítaná? ▼

Obrázek 6 - Sekce O kalkulátoru

4.3.4 Kalkulátor

Případ užití:

V Kalkulátoru uživatel očekává:

- možnost vložení podlahové plochy nemovitosti
- možnost vložení plochy pozemku nemovitosti
- možnost výběru stavu nemovitosti
- možnost vložení počtu parkovacích míst
- možnost vložení ostatních výměr (například sklep)

V Kalkulátoru uživatel požaduje:

- možnost výběru druhu nemovitosti (dům / byt)
- možnost výběru prodeje či pronájmu nemovitosti
- možnost vložení adresy nemovitosti
- výpočet ceny prodeje či pronájmu nemovitosti na základě vložených údajů

Scénář:

Systém v části kalkulace ceny nemovitosti zobrazí na základě požadavků uživatele přepínací tlačítka (radio-button) umožňující výběr druhu nemovitosti tedy, zda se jedná o dům či byt.

Systém dále na základě požadavků uživatele zobrazí přepínací tlačítka s výběrem, zda má uživatel zájem o prodej či pronájem nemovitosti.

Systém na základě požadavků uživatele zobrazí vstupní pole s možností zadání adresy nemovitosti. Systém u tohoto pole umožňuje předvyplnění adresy během zadávání adresy uživatelem tak, že mu podle vložených písmen začne nabízet relevantní adresy a uživatel si ze zobrazeného seznamu má možnost vybrat.

Systém na základě očekávání uživatele blíže specifikovat vložené údaje zobrazí vstupní pole pro možnost vložení podlahové plochy, plochy, počtu parkovacích míst a ostatních výměr nemovitosti.

Dále systém dle očekávání uživatele zobrazí výběrové pole na výběr stavu nemovitosti.

Systém také zobrazí tlačítko, které po kliknutí a zadání všech údajů zobrazí modální okno s odhadovanou cenou nemovitosti. Popis modálního okna je sepsán níže.

Drátěný model:

The wireframe shows a dark blue background with the title "Kalkulátor nemovitostí" in white. Below the title are three columns of three white placeholder boxes, each with a small downward arrow on the right. At the bottom center is a light blue button labeled "ODESLAT".

Obrázek 7 - Kalkulátor nemovitosti

Prototyp:

The prototype features a dark blue header with the title "KALKULÁTOR NEMOVITOSTI" in white. The form is organized into eight numbered steps:

- 1. DRUH NEMOVITOSTI**: Radio buttons for "Dům" and "Byt".
- 2. PRODEJ NEBO PRONÁJEM**: Radio buttons for "Prodej" and "Pronájem".
- 3. ADRESA NEMOVITOSTI**: A text input field with the placeholder "Zadejte adresu nemovitosti".
- 4. PODLAHOVÁ PLOCHA**: A text input field with the placeholder "Zadejte plochu v m²".
- 5. PLOCHA POZEMKU**: A text input field with the placeholder "Zadejte plochu v m²".
- 6. STAV NEMOVITOSTI**: A dropdown menu with the placeholder "Vyberte stav nemovitosti".
- 7. POČET PARKOVACÍCH MÍST**: A text input field with the placeholder "Zadejte počet".
- 8. OSTATNÍ VÝMĚRY**: A text input field with the placeholder "Např. sklep, zadejte v m²".

At the bottom center is a dark blue button labeled "ZOBRAZIT CENU". Below it is a red error message: "Ve formuláři se nachází chyby".

Obrázek 8 - Kalkulátor nemovitosti

4.3.5 Modální okno

Případ užití:

V Modálním okně uživatel očekává:

- krátký a výstižný nadpis a popis k cenovému odhadu nemovitosti

V Modálním okně uživatel požaduje:

- zobrazit odhadovanou cenu nemovitosti na základě zadaných údajů v kalkulátoru

Scénář:

System po kliknutí na tlačítko pro výpočet ceny v sekci Kalkulátoru zobrazí modální okno, ve kterém systém zobrazí uživatelem očekávané a požadované prvky. System tedy zobrazí nadpis, krátký popis o odhadované ceně a samotnou odhadovanou cenu nemovitosti v českých korunách. System také umožňuje modální okno zavřít pomocí křížku v pravém horním rohu modálního okna.

Drátěný model:



Obrázek 9 - Modální okno s odhadovanou cenou nemovitosti

Prototyp:



Obrázek 10 - Modální okno s odhadovanou cenou nemovitosti

4.3.6 Kontakt

Případ užití:

V sekci Kontaktů uživatel očekává:

- možnost vyplnit jméno a příjmení
- možnost vyplnit e-mail
- možnost vyplnit telefonní číslo

V sekci Kontaktů uživatel požaduje:

- možnost napsat zprávu
- možnost odeslat formulář

Scénář:

System zobrazí uživatelem očekávané a požadované prvky. V sekci Kontaktů zobrazí nadpis a krátký popis, pod kterým systém zobrazí formulář. Na prvním řádku formuláře zobrazí vstupní pole pro jméno a příjmení. Na druhém řádku zobrazí dvě vstupní pole, jedno pole pro e-mail a druhé pro vyplnění telefonního čísla, označené zkratkou telefon. Na posledním, třetím, řádku systém zobrazí vstupní pole pro zprávu. Následně pod formulářem zobrazí tlačítko pro odeslání kontaktního formuláře.

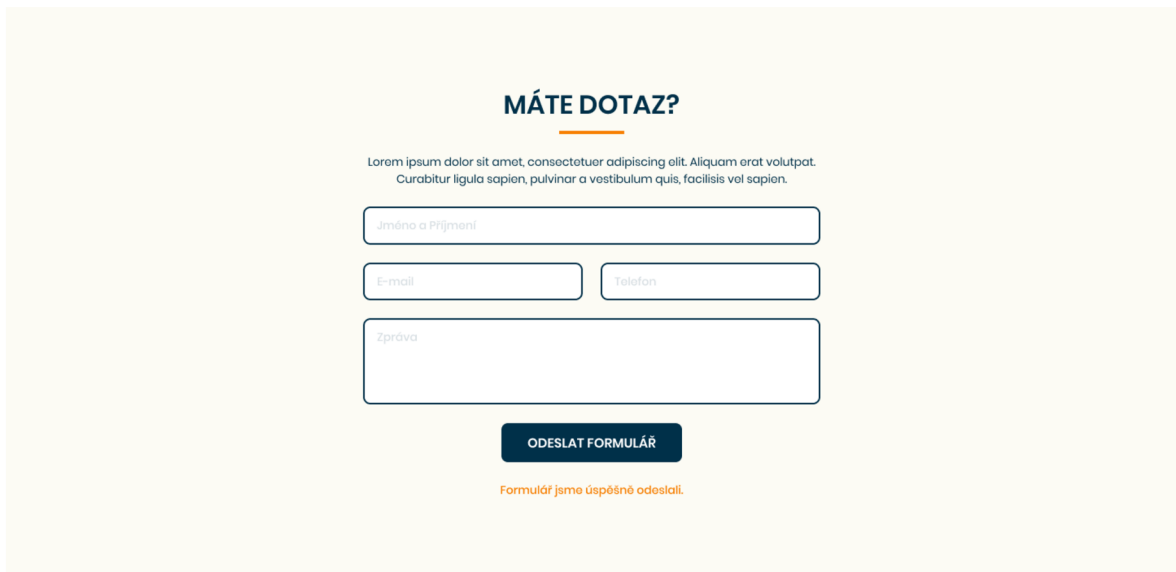
Drátěný model:

The wireframe shows a contact form titled "Kontaktujte nás". Below the title is a placeholder text: "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim". The form consists of the following elements:

- A single input field for "Jméno a příjmení".
- Two input fields: "E-mail" and "Telefon".
- A larger text area for "Zpráva".
- A dark button labeled "ODESLAT" at the bottom.

Obrázek 11 - Sekce Kontaktů

Prototyp:



The image shows a contact form prototype on a light yellow background. At the top, the heading "MÁTE DOTAZ?" is centered in bold black text, with a short orange horizontal line underneath. Below the heading is a paragraph of placeholder text: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam erat volutpat. Curabitur ligula sapien, pulvinar a vestibulum quis, facilisis vel sapien." The form consists of four input fields: a single-line text box labeled "Jméno a Příjmení", two side-by-side single-line text boxes labeled "E-mail" and "Telefon", and a larger multi-line text area labeled "Zpráva". Below these fields is a dark blue button with the white text "ODESLAT FORMULÁŘ". Underneath the button, a success message "Formulář jsme úspěšně odeslali." is displayed in a small orange font.

Obrázek 12 - Sekce Kontaktů

4.4 Úvod do webové aplikace

Webová aplikace na výpočet cenového odhadu nemovitosti byla vytvořena v programovacím jazyce PHP za použití Symfony frameworku, který je blíže popsán autorem v teoretické části bakalářské práce. Pro vývoj webové aplikace byla použita zejména vývojáři webových aplikací populární architektura MVC (model-view-controller), která odděluje řídicí logiku aplikace od datového modelu a uživatelského rozhraní. (38)

Vzhledem k tomu, že je webová aplikace jednostránkovou webovou aplikací, byl vytvořen pouze HomeController, který má na starosti požadavky na zpracování logiky, jež probíhá v odpovídajících Services. V Controlleru konkrétně probíhá nastavování Route a dále jsou zde vytvořené funkce *index()*, *ajaxContact()* a *ajaxCalculator()*:

- funkce *index()* vykresluje celou úvodní stránku *Homepage*
- funkce *ajaxContact()* zpracovává požadavek na odeslání kontaktního formuláře
- funkce *ajaxCalculator()* zpracovává požadavek na výpočet ceny nemovitosti

Symfony je skvělým PHP frameworkem i z pohledu možnosti vytváření formulářů. Webová aplikace se skládá ze dvou formulářů, které jsou specifikovány v *CalculatorType* a

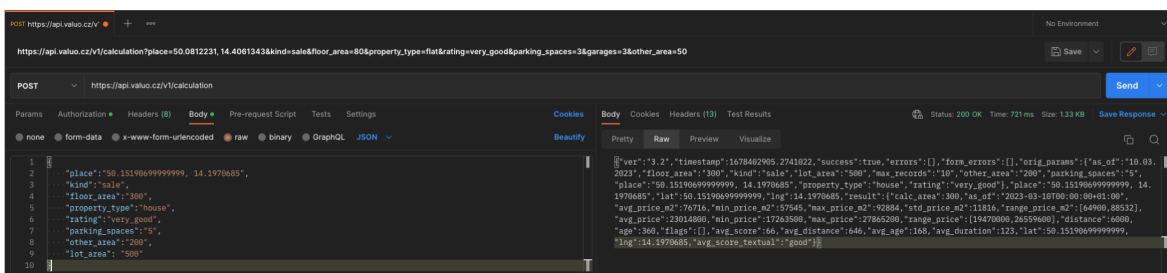
ContactType, kde se definují všechna pole a prvky samotného formuláře. Formuláře jsou následně vypsané v Twig šabloně. V Services, konkrétně *ApiService* a *MailService*, tedy probíhá práce s daty a zpracování jednotlivých formulářů.

Webová aplikace pracuje se dvěma entitami, a to *Calculator* a *Contact*, ve kterých jsou specifikované všechny potřebné vlastnosti samotných entit a potřebné *get()*, *set()* metody.

Co se týče front-endové části, tak webová aplikace pracuje s preprocesorem SCSS, ve kterém jsou definované styly webové aplikace. Dále pracuje s jQuery, JavaScriptovou knihovnou a Twig šablonami, kde je specifikovaná HTML struktura aplikace.

4.4.1 Valuo API

Pro ověření fungování API a pro ověření vrácených výsledků byla použita aplikace Postman, která je jednou z nejpoužívanějších aplikací využívaných na práci s API. Pro získání odpovědi z API je použita metoda Post.



Obrázek 13 - Testování API v aplikaci Postman

Implementace napojení na Valuo API ve webové aplikaci se provádí v *ApiService*, a to ve funkci *getValuoData*. Pro umožnění práce s API je potřeba získat tzv. Bearer token, který je v případě této bakalářské práce vygenerován ze strany firmy Valuo. Výsledkem, který je získán odesláním požadavku na API, jsou data ve formátu JSON, s kterými se dále pracuje.


```

private function getValuoData(array $data): array
{
    $response = $this->client->request(
        method: 'POST',
        url: 'https://api.valuo.cz/v1/calculation',
        [
            'auth_bearer' => $this->valuoApiKey,
            'json' => $data,
        ]
    );

    return $response->toArray();
}

```

Obrázek 14 - Implementace Valuo API v kódu

Odesílání dat na API se provádí v *ApiService*, a to ve funkci *setApiData()*. Vzhledem k naplnění uživatelských cílů vývojářům stačí ze získaných dat z API pouze průměrná cena, a proto je jedinou vrácenou hodnotou funkce.

```

public function setApiData(Calculator $calculator): string
{
    $apiData = [];
    $apiData['property_type'] = $calculator->getPropertyType();
    $apiData['kind'] = $calculator->getSellRent();
    $apiData['place'] = $calculator->getCoordinates();
    $apiData['floor_area'] = $calculator->getFloorArea();
    $apiData['lot_area'] = $calculator->getLandArea();
    $apiData['rating'] = $calculator->getPropertyCondition();
    $apiData['parking_spaces'] = $calculator->getParkingSpaces();
    $apiData['other_area'] = $calculator->getOtherDimensions();

    $date = new DateTime();
    $date = $date->format( format: 'd.m.Y');
    $apiData['as_of'] = $date;

    $response = $this->getValuoData($apiData);

    if ($response["success"] === true){
        return $response["result"]["avg_price"];
    }

    return "Nebyla vyhodnocena žádná cena.";
}
}

```

Obrázek 15 - Zaslání dat na Valuo API

4.4.2 Google Maps

Google Maps se v projektu této bakalářské práce využívá pro automatické doplnění zadávané adresy ve vstupním poli pro lokaci. Tato API také umožňuje danou lokaci převést na souřadnice, s kterými dále Valuo API na výpočet ceny nemovitosti pracuje.

Pro práci nejen s touto, ale i s dalšími službami od Googlu přes API, je potřeba mít založený Google Cloud Console účet, ve kterém je dále nutné nastavit fakturační údaje, přidat platební prostředek a vytvořit první projekt. Následně je ve vytvořeném projektu potřeba povolit služby, jež budou využity a následně vygenerovat API klíč, který je obecně doporučováno omezit pouze na danou doménu, aby nemohl být využit i jinou, nepovolanou, osobou.

Pro samotnou implementaci služeb Google poskytuje Google detailní dokumentaci, dle které je doporučováno postupovat. Pro potřeby projektu této bakalářské práce je konkrétně využívána služba Maps JavaScript API, která mimo možnosti zmíněné výše umožňuje na webové aplikaci či stránce například i zobrazení mapy. Prvně je potřeba do webové aplikace implementovat skript umožňující práci s funkcemi, které tato služba od Googlu přináší. Do zdroje s URL se také přidává API klíč vygenerovaný v Google Cloud Console.

```
<script async=""
  src="https://maps.googleapis.com/maps/api/js?key={{ google_maps_api }}&region=CZ&language=cz&libraries=places&callback=initMap"
  type="text/javascript"></script>
<script>
```

Obrázek 16 - Vložení skriptu pro Google Maps API

Dalším nutným krokem je vytvoření skriptu, který tzv. Autocomplete nastaví a dosadí pro dané vstupní pole. Následně je po události, která ve vstupním poli zaznamenává změny *place_changed*, potřeba získat souřadnice vybrané lokace. Protože není potřeba získané souřadnice koncovému uživateli ukazovat, jsou dosazeny ve formuláři do neviditelného vstupního pole, což umožňuje jejich odeslání na Valuo API a získání ceny pro vybranou lokalitu.

```

<script>
function initMap() {
  const addressInput = document.getElementById("calculator_propertyAddress");
  const options = {
    componentRestrictions: {country: ["cz"]},
    language: ['cs'],
  }
  const autocomplete = new google.maps.places.Autocomplete(addressInput, options);

  autocomplete.addListener("place_changed", () => {
    const place = autocomplete.getPlace();
    if (!place.geometry) {
      console.log("No geometry found for selected place.");
      return;
    }
    const latitude = place.geometry.location.lat();
    const longitude = place.geometry.location.lng();
    document.getElementById("calculator_coordinates").value = `${latitude}, ${longitude}`;
  });
}
</script>

```

Obrázek 17 - Skript na vytvoření Autocomplete do vstupního pole a vrácení souřadnic vybrané lokace

4.5 Zpracování formulářů

Ve webové aplikaci se nachází dva formuláře, jeden kontaktní a druhý na výpočet ceny nemovitosti. Oba formuláře jsou zpracovány pomocí technologie Ajax, která mimo jiné umožňuje odeslání formuláře bez znovunačtení stránky, což zjednodušuje uživatelský požitek. Pro tento účel byla použita i v této webové aplikaci.

Kontaktní formulář je odesílán pomocí SMTP (Simple Mail Transfer Protocol), který je určený pro odesílání elektronické pošty, čili e-mailů. Posílání formuláře touto metodou je běžnou praxí a používá se především z důvodu zamezení spamu, což je jedním z důvodů jeho implementace i v této webové aplikaci. Pro uplatnění této metody je potřeba vyplnit e-mailovou adresu, z které se bude formulář odesílat, a heslo k ní v kódu webové aplikace.

```

$( 'form[name="contact"]' ).submit(function (e) {
    e.preventDefault(); // avoid to execute the actual submit of the form.

    var form = $(this);
    var message = form.find(".message");

    $.ajax({
        type: "POST",
        url: "{{ path('ajax_contact') }}",
        data: form.serialize(), // serializes the form's elements.
        success: function (data) {
            form[0].reset();
            message.css("display", "flex");
            message.text(data); // show response from the php script.
        },
        fail: function (data) {
            message.css("display", "flex");
            message.text(data); // show response from the php script.
        }
    });
});

```

Obrázek 18 - Ajax zpracování kontaktního formuláře

Formulář kalkulátoru, jak již bylo zmíněno výše, odesílá data ze vstupních polí na Valuo API, která následně vrací data o nemovitosti. Pro účely této webové aplikace však autor pracuje pouze s průměrnou cenou dané nemovitosti, která se poté pouze zformátuje na evropské standardy a vypíše v modálním okně.

```

$( 'form[name="calculator"]' ).submit(function (e) {
    e.preventDefault(); // avoid to execute the actual submit of the form.

    var form = $(this);
    var message = form.find(".message");
    var calculation = $(".kalkulace");

    $.ajax({
        type: "POST",
        url: "{{ path('ajax_calculator') }}",
        data: form.serialize(), // serializes the form's elements.
        success: function (data) {
            calculation.css("display", "flex");
            calculation.find(".kalkulace__price").text(new Intl.NumberFormat("de-DE").format(data) + " Kč");
            message.text(data); // show response from the php script.
        },
        fail: function (data) {
            message.css("display", "flex");
            message.text(data); // show response from the php script.
        }
    });
});

```

Obrázek 19 - Ajax zpracování formuláře kalkulátoru

5 Výsledky a diskuse

5.1 Spuštění aplikace

Webová aplikace je spuštěna na dedikovaném serveru u společnosti Hetzner Online GmbH, na kterém běží v kontejneru na virtualizační technologii Docker na PHP verzi 8.2. Verzování webové aplikace je nastaveno na platformě GitHub, v němž je nastavený také deploy celé webové aplikace. Výhodou deploye je pro vývojáře skutečnost, že není potřeba pokaždé nahrávat novou verzi na server manuálně, ale stačí pouze tzv. pushnout změny na GitHub do větve (anglicky branch) *master* a nová verze webové aplikace se následně projeví i na serveru. Na server s webovou aplikací je nasměrovaná subdoména `kalkulator.pesel.cz`, pod kterou je možné kalkulačtor nemovitosti nalézt širokou veřejností.

5.2 Testování

V průběhu vývoje bylo testování webové aplikace prováděno autorem této bakalářské práce. Díky tomu autor předešel možným kritickým chybám, které by bývaly mohly požitkem z používání kalkulačtoru nemovitosti u koncových uživatelů znepříjemnit. Během samotného testování autor také nacházel možná zlepšení, která v průběhu vývoje implementoval a zároveň v průběhu testování odstraňoval nalezené chyby.

Následně byla webová aplikace otestována v ostrém provozu uživateli odpovídajícími různým typů person. Obecně se dá konstatovat spokojenost uživatelů s fungováním aplikace, avšak se autor několikrát setkal s názory, že je výsledná odhadovaná cena nemovitosti nižší, než je tomu v realitě.

6 Závěr

Cílem této bakalářské práce bylo vytvořit funkční webovou aplikaci, která dokáže odhadnout cenu prodeje či pronájmu nemovitosti na základě vstupních údajů zadaných uživateli.

V první, teoretické části, této práce bylo mým cílem čtenáře seznámit s problematikou a využitím on-line kalkulátorů nemovitostí. Následovalo bližší seznámení čtenáře s použitými technologiemi na tvorbu webové aplikace pro výpočet odhadu ceny nemovitosti a některými pojmy, které se hojně objevují v praktické části mé práce, a to proto, aby byla bakalářská práce srozumitelná pro široký okruh veřejnosti a nedocházelo k používání pro čtenáře neznámých pojmů.

V druhé, praktické části, popisují vytvoření finální podoby webové aplikace krok za krokem. Nejdříve bylo za potřebí si vytyčit konkrétní osoby a jejich uživatelské cíle, což umožnilo vytvoření drátěného modelu a na základě jeho zpracování jsem připravil interaktivní prototyp v aplikaci Figma. Po zhotovení této části návrhu webové aplikace je čtenářům bližší přiblížen samotný vývoj webové aplikace, pro který jsem zvolil PHP framework Symfony, jelikož mám s tímto frameworkem již zkušenosti a také ho považuji za jednu z nejlepších technologií na vývoj webových aplikací. Na základě dříve zhotoveného prototypu byla vytvořena struktura stránky, aneb front-endová část, za použití Twig šablon a následně nastýlována opět dle připraveného prototypu. Následně jsem tuto část otestoval na několika zařízeních k zajištění jistoty, že nakódovaná stránka odpovídá vytvořenému prototypu. Bližší je v praktické části popsána druhá část vývoje spočívající v naprogramování back-endové části webové aplikace, ve které jsem zprovoznil kontaktní formulář a formulář na výpočet ceny nemovitosti.

Po úspěšném otestování širším okruhem uživatelů považuji tuto webovou aplikaci na odhad ceny nemovitosti za funkční a úspěšnou. Nicméně se nabízí možná vylepšení v oblasti konkretizaci údajů o nemovitosti, aby výsledný odhad více odpovídal skutečnému stavu, a tím byla i cena přesnější. Obecně však kalkulátor na odhad ceny nemovitosti splňuje uživatelské cíle a zejména je uživatelsky přívětivý, tím pádem je způsobilý být využíván širokým okruhem lidí, a to především z důvodu jeho snadné interakce. Dále se nabízí více propracovat design webové aplikace například přidáním načítacího okna při generování odhadu ceny nebo skrývání nejčastěji kladených otázek při rozkliknutí další otázky.

Navzdory možnému vylepšení webové aplikace, je již v aktuální podobě způsobilá pomoci širokému okruhu uživatelů. Došlo tedy k naplnění cílů bakalářské práce.

7 Seznam použitých zdrojů

1. Valuo Profi. *Valuo*. [Online] Valuo Technologies s. r. o., 2017. [Citace: 24. Únor 2023.] <https://profi.valuo.cz/calculation/>.
2. Odhad Zdarma. *Odhad Zdarma*. [Online] ODHAD ZDARMA s.r.o., 2016. [Citace: 24. Únor 2023.] <https://www.odhad-zdarma.cz/>.
3. Dohnal, Ing. Václav. Za kolik prodal soused nemovitost? *dohnalvaclav.cz*. [Online] 13. Duben 2016. [Citace: 24. Únor 2023.] <https://dohnalvaclav.cz/za-kolik-prodal-soused-nemovitost/>.
4. Michal Bureš. Jak si vyřídít výpis z katastru nemovitostí online? *Finance.cz*. [Online] 10. Květen 2022. [Citace: 24. Únor 2023.] <https://www.finance.cz/517482-vypis-z-katastru-nemovitosti-online/>.
5. Cenová mapa nemovitostí celé ČR - Atlas cen. *Reas.cz*. [Online] Nexter Company s.r.o., 2016. [Citace: 24. Únor 2023.] <https://www.reas.cz/cenova-mapa#co-to-umi>.
6. Makléři, hledáte přesný online nástroj pro úspěšný prodej? *DataZtrhu.cz*. [Online] 2021. [Citace: 24. Únor 2023.] <https://www.dataztrhu.cz/pro-maklere>.
7. Odhad ceny nemovitostí. *Kotula.cz*. [Online] 25. Listopad 2021. [Citace: 24. Únor 2023.] <https://www.kotula.cz/odhad-ceny-nemovitosti/>.
8. Kalkulačka Odhad-zdarma.cz letos vypracovala už víc než 250.000 odhadů. *Médiář*. [Online] 1. Listopad 2021. [Citace: 24. Únor 2023.] <https://www.mediar.cz/kalkulacka-odhad-zdarma-cz-letos-vypracovala-uz-vic-nez-250-000-odhadu/>.
9. Webová aplikace (Web Application). *ManagementMania.com*. [Online] 18. Říjen 2018. [Citace: 25. Únor 2023.] <https://managementmania.com/cs/webova-aplikace-web-application>.
10. Web, webová stránka a webová aplikace, v čem je rozdíl? *Rascasone*. [Online] 22. Červen 2021. [Citace: 25. Únor 2023.] <https://www.rascasone.com/cs/blog/web-webova-aplikace-rozdil>.

11. Co jsou to webové aplikace? Zdroj: <https://pr.denik.cz/doporucujeme/co-jsou-to-webove-aplikace-20220404.html>. *Denik.cz*. [Online] 3. Duben 2022. [Citace: 25. Únor 2023.] <https://pr.denik.cz/doporucujeme/co-jsou-to-webove-aplikace-20220404.html>.
12. Frontend vs Backend. *GeeksforGeeks*. [Online] 6. Červen 2023. [Citace: 25. Únor 2023.] <https://www.geeksforgeeks.org/frontend-vs-backend/>.
13. Vernerová, Sára. Front end vs. Back end - jaký je mezi nimi rozdíl? *ApiTree*. [Online] 9. Březen 2021. [Citace: 25. Únor 2023.] <https://www.apitree.cz/blog/front-end-vs-back-end-jaky-je-mezi-nimi-rozdil>.
14. The flexible, fast, and secure template engine for PHP. *Twig*. [Online] [Citace: 25. Únor 2023.] <https://twig.symfony.com/doc/3.x/intro.html>.
15. Symfony po krůčkách – Twig. *Zdrojak.cz*. [Online] 8. Březen 2016. [Citace: 25. Únor 2023.] <https://zdrojak.cz/clanky/symfony-po-kruckach-twig/>.
16. Sharkie, Craig a Fisher, Andrew. *Responzivní webdesgin Okamžitě*. Brno : Albatros Media a. s., 2015. ISBN 978-80-251-4384-1.
17. Bittner, Honza. Lekce 2 - Úvod do CSS preprocesoru Sass Zdroj: <https://www.itnetwork.cz/html-css/webove-portfolio/tutorial-moderni-webove-portfolio-sass>. *itnetwork.cz*. [Online] [Citace: 25. Únor 2023.] <https://www.itnetwork.cz/html-css/webove-portfolio/tutorial-moderni-webove-portfolio-sass>.
18. Gosselin, Jeff. Sassy CSS (SCSS) Quick Guide. *Medium.com*. [Online] 20. Říjen 2019. [Citace: 25. Únor 2023.] <https://medium.com/@jgoz/sassy-css-scss-quick-guide-b38f51c6868a>.
19. Pehlivanian, Ara a Nguyen, Don. *JavaScript Okamžitě*. Brno : Albatros Media a. s., 2014. ISBN 978-80-251-4163-2.
20. Kod'ousková, Barbora. JavaScript pro začátečníky: co to je a jak funguje. *Rascasone*. [Online] 28. Leden 2022. [Citace: 25. Únor 2023.] <https://www.rascasone.com/cs/blog/co-je-javascript-pro-zacatecniky>.

21. Michálek, Martin. Webpack: Úplné základy a tutoriál k tomu. *Vzhůru dolů*. [Online] 9. Říjen 2019. [Citace: 25. Únor 2023.] <https://www.vzhurudolu.cz/prirucka/webpack>.
22. Webpack. *Ackee*. [Online] [Citace: 25. Únor 2023.] <https://www.ackee.cz/blog/glossary/webpack>.
23. Dami development s.r.o. Backend. *Dami*. [Online] [Citace: 26. Únor 2023.] <https://www.damidev.com/slovník/backend>.
24. Lemonaki, Dionysia. Frontend VS Backend – What's the Difference? *FreeCodeCamp*. [Online] Březen. 18 2022. [Citace: 26. Únor 2023.] <https://www.freecodecamp.org/news/frontend-vs-backend-whats-the-difference/#beintro>.
25. Ferguson, Nicole. What's the Difference Between Frontend and Backend Web Development? *CareerFoundry*. [Online] 28. Listopad 2022. [Citace: 26. Únor 2023.] <https://careerfoundry.com/en/blog/web-development/whats-the-difference-between-frontend-and-backend/>.
26. Dostalová, Zuzana. FRONTEND VS. BACKEND. *Czechitas*. [Online] 1. Červenec 2014. [Citace: 26. Únor 2023.] <https://www.czechitas.cz/blog/frontend-vs-backend>.
27. Blažková, Tereza. Má smysl začínat s PHP? *ITnetwork.cz*. [Online] 24. Červenec 2021. [Citace: 26. Únor 2023.] <https://www.itnetwork.cz/blog/ma-smysl-zacinat-s-php>.
28. Kosek, Jiří. *PHP - tvorba interaktivních internetových aplikací*. Praha : Grada Publishing, spol. s r.o., 1999. ISBN 80-7169-373-1.
29. What is PHP and Why we use it? *GeeksforGeeks*. [Online] 12. Květen 2022. [Citace: 26. Únor 2023.] <https://www.geeksforgeeks.org/what-is-php-and-why-we-use-it/>.
30. Shukla, Surabhi. What Makes Symfony Framework a Great Choice for PHP Web Development? *Net Solutions*. [Online] 9. Prosinec 2022. [Citace: 26. Únor 2023.] <https://www.netsolutions.com/insights/symfony-framework-features/>.
31. Máca, Jindřich. Lekce 1 - Úvod do Symfony frameworku pro PHP. *ITnetwork.cz*. [Online] [Citace: 26. Únor 2023.] <https://www.itnetwork.cz/php/symfony/zaklady/uvod-do-symfony-frameworku-pro-php>.

32. Javorek, Honza. Úvod do API. *Co je API*. [Online] 2020. [Citace: 26. Únor 2023.] <https://cojeapi.cz/01-uvod-do-api.html>.
33. Smejkal, Petr. Co je API a k čemu slouží? *Petr Smejkal - tvorba webových stránek*. [Online] 25. Duben 2021. [Citace: 26. Únor 2023.] <https://www.petrsmejkal.cz/clanky/co-je-api-a-k-cemu-slouzi/>.
34. Personas. *Interaction Design*. [Online] [Citace: 27. Únor 2023.] <https://www.interaction-design.org/literature/topics/personas>.
35. Pavlíček, Josef. The Cookbook for Interaction Design and Human Computer Interaction. [Online] [Citace: 27. Únor 2023.] https://docs.google.com/presentation/d/1nbLjgEX5mS6kl_cRx6CeKuhd-fzz-kyYn_j03vMLkH4/edit#slide=id.g9a94b8846e_0_7.
36. Martinek, Jan. Use Case. *WikiKnihovna*. [Online] 15. Leden 2013. [Citace: 28. Únor 2023.] https://wiki.knihovna.cz/index.php/Use_Case.
37. Dubinská, Lída. Co je wireframu webu, proč ho potřebujete a jak ho vytvořit? *Rascasone*. [Online] 10. Listopad 2022. [Citace: 28. Únor 2023.] <https://www.rascasone.com/cs/blog/co-je-wireframe-predstavujeme-5-duvodu-proc-je-pro-klienty-drateny-model-dulezity>.
38. Kodřousková, Barbora. Architektura MVC: definice, struktura, frameworky. *Rascasone*. [Online] 13. Duben 2021. [Citace: 2. Březen 2023.] <https://www.rascasone.com/cs/blog/architektura-mvc-struktura-frameworky>.

8 Seznam obrázků a zkratk

8.1 Seznam obrázků

Obrázek 1 - Záhloví webové aplikace	27
Obrázek 2 - Záhloví webové aplikace	28
Obrázek 3 - Úvodní obrazovka webové aplikace	28
Obrázek 4 - Úvodní obrazovka webové aplikace	29
Obrázek 5 - Sekce O kalkulátoru	30
Obrázek 6 - Sekce O kalkulátoru	30
Obrázek 7 - Kalkulátor nemovitosti	32
Obrázek 8 - Kalkulátor nemovitosti	32
Obrázek 9 - Modální okno s odhadovanou cenou nemovitosti	33
Obrázek 10 - Modální okno s odhadovanou cenou nemovitosti	33
Obrázek 11 - Sekce Kontaktů	34
Obrázek 12 - Sekce Kontaktů	35
Obrázek 13 - Testování API v aplikaci Postman	36
Obrázek 14 - Implementace Valuo API v kódu	37
Obrázek 15 - Zasilání dat na Valuo API	37
Obrázek 16 - Vložení skriptu pro Google Maps API	38
Obrázek 17 - Skript na vytvoření Autocomplete do vstupního pole a vrácení souřadnic vybrané lokace	39
Obrázek 18 - Ajax zpracování kontaktního formuláře	40
Obrázek 19 - Ajax zpracování formuláře kalkulátoru	40

8.2 Seznam použitých zkratk

- API - Application Programming Interface
- PHP – dříve Personal Homepage a nyní Hypertext Preprocessor
- CSS - Cascading Style Sheets
- SCSS – Sassy Cascading Style Sheets
- HTML - Hypertext Markup Language
- JS - JavaScript
- UI - User Interface
- UX - User Experience
- HCI - Human-Computer Interaction

- UML - Unified Modeling Language
- MVC - Model-View-Controller
- URL - Uniform Resource Locator
- SMTP - Simple Mail Transfer Protocol

9 Přílohy

9.1 Příloha 1 – Drátěný model a prototyp webové aplikace

<https://www.figma.com/file/uIRpeQy7WSKpgL1JD2eDy/Bakal%C3%A1%C5%99k%C3%A1-pr%C3%A1ce?node-id=1%3A2&t=DCLy2IUdZAKDfsMH-1>

9.2 Příloha 2 – Paměťová karta s výslednou webovou aplikací