

Návrh databázové struktury a prezentačních šablon pro prodejce automobilů

Denis Ferda

Bakalářská práce
2019



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Denis Ferda**
Osobní číslo: **A16336**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informační technologie v administrativě**
Forma studia: **prezenční**

Téma práce: **Návrh databázové struktury a prezentačních šablon pro prodejce automobilů**

Téma anglicky: **A Design of Database Structures and Presentation Templates for Car Sales Companies**

Zásady pro vypracování:

1. Provedte rešerši existujících řešení.
2. Vypracujte stručný rozbor technologií, které budou použity k návrhu.
3. Provedte rozbor a analýzu požadavků na zvolené řešení.
4. Realizujte navrženou aplikaci.
5. Navrženou aplikaci vhodným způsobem popište.
6. Věnujte pozornost zabezpečení.

Rozsah bakalářské práce: –

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **GOLDSTEIN, Alexis, Louis LAZARIS a Estelle WEYL. HTML5 a CSS3 pro webové designéry. Brno: Zoner Press, 2011. Encyklopedie webdesignera. ISBN 978-80-7413-166-0.**
2. **MICHÁLEK, Martin. Vzhůru do (responzivního) webdesignu. Verze 1.1. Praha: vlastním nákladem autora, 2017. ISBN 978-80-88253-00-6.**
3. **LAMBERT, Matt. Learning Bootstrap 4. 2nd Revised edition. Birmingham: Packt Publishing, 2016. ISBN 978-1785881008.**
4. **AQUINO, Chris a Todd GANDEE. Front-end web development: the Big Nerd Ranch guide. Atlanta, GA: Big Nerd Ranch, 2016. ISBN 978-0134433943.**
5. **DUCKETT, Jon, Gilles RUPPERT a Jack MOORE. JavaScript & jQuery: interactive front-end web development. Indianapolis, IN: Wiley, 2014. ISBN 9781118531648.**
6. **SKLAR, David a Jan POKORNÝ. PHP 7: Praktický průvodce nejrozšířenějším skriptovacím jazykem pro web. Brno: Zoner Press, 2018. ISBN 978-80-7413-363-3.**

Vedoucí bakalářské práce:

Ing. Petr Šilhavý, Ph.D.

Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce:

30. listopadu 2018

Termín odevzdání bakalářské práce:

15. května 2019

Ve Zlíně dne 7. prosince 2018



doc. Mgr. Milan Adámek, Ph.D.
děkan

doc. Ing. Martin Šysel, Ph.D.
garant oboru

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 17. 5. 2019

Denis Ferda v. r.
podpis diplomanta

ABSTRAKT

Bakalářská práce se zabývá návrhem databázové struktury a realizací webové prezentace s prvky jednoduché aplikace pro prodejce automobilů, která disponuje relevantními funkcionalitami pro daný obor. Součástí práce je popis použitých technologií, analýza existujících řešení a definice požadavků na výsledné řešení.

Návrh databázové struktury byl realizován pomocí online nástroje Lucidchart a je určen pro databázový systém MySQL. Při realizaci webové aplikace byly využity standardní technologie a programové vybavení pro vývoj webových aplikací. Šablona aplikace je založena na frameworku Bootstrap. Zohledněno je taktéž zabezpečení aplikace.

Výsledek práce lze rozdělit na dvě části. První je realizovaný návrh databázové struktury, která je použitelná u prodejce automobilů pro potřeby webové aplikace a k evidenci dalších dat, která jsou relevantní pro daný obor. Druhá část je responzivní webová prezentace s prvky jednoduché aplikace, která umožňuje výpis a filtrování vozidel, registraci, přihlášení a umožňuje online rezervaci automobilu. Popis návrhu databázové struktury a realizované aplikace je předmětem praktické části práce. Přílohou bakalářské práce je archiv obsahující zdrojové kódy webové aplikace a dokument ve formátu PDF, který obsahuje fyzický model databázové struktury.

Klíčová slova: webová prezentace, webová aplikace, databázová struktura, relační model, prodejce automobilů, html, css, php, bootstrap

ABSTRACT

This bachelor thesis is about design of database structure and realization of web presentation with elements of simple application for car sellers with relevant functionalities for a given field. Part of the thesis is description of used technologies, analysis of existing solutions and definition of requirements for final solution.

Database structure design was implemented using the Lucidchart online tool and is intended for the MySQL database system. For the implementation of the web application were used standard technologies and software for web application development. The application template is based on the Bootstrap framework. Application security is also taken into account.

The result of the work can be divided into two parts. The first is the implementation of a database structure which can be used by car sellers for the needs of a web application and to record other data that are relevant to the industry. The second part is a responsive web presentation with elements of a simple application that allows the listing and filtering of vehicles, registration, login and allows online car reservation. The description of the database structure design and the implemented application is the subject of the practical part of the thesis. Attached to the bachelor thesis is an archive containing the source code of the web application and a PDF document that contains the physical model of the database structure.

Keywords: web presentation, web application, database structure, relation model, car sales, html, css, php, bootstrap

Rád bych poděkoval vedoucímu mé bakalářské práce Ing. Petrovi Šilhavému, PhD., za poskytnuté rady, tipy a uskutečněné konzultace týkající se tématu práce.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD.....	8
I TEORETICKÁ ČÁST.....	9
1 POUŽITÉ TECHNOLOGIE	10
1.1 WORLD WIDE WEB.....	10
1.2 HYPERTEXT MARKUP LANGUAGE.....	10
1.2.1 VÝVOJ HTML.....	10
1.2.2 TERMINOLOGIE JAZYKA HTML	12
1.2.3 HTML 5.....	13
1.2.4 SYNTAXE HTML5.....	13
1.2.5 SÉMANTIKA V HTML 5.....	14
1.1 CASCADING STYLE SHEETS	16
1.1.1 VÝVOJ CSS.....	16
1.1.2 APLIKACE CSS.....	17
1.1.3 SYNTAXE JAZYKA CSS.....	17
1.1.4 VLASTNOSTI CSS 3	18
1.1.5 CSS3 MEDIA QUERIES	18
1.2 JAVASCRIPT.....	20
1.3 FRAMEWORK A KNIHOVNY	20
1.3.1 FRAMEWORK BOOTSTRAP.....	20
1.3.2 KNIHOVNY ANIMATE.CSS A HOVER.CSS.....	21
1.3.3 KNIHOVNA FONT-AWESOME	22
1.3.4 KNIHOVNA JQUERY.....	22
1.4 HYPERTEXT PREPROCESSOR.....	23
1.5 DATABÁZOVÝ SYSTÉM.....	23
1.5.1 BÁZE DAT.....	24
1.5.2 SYSTÉM ŘÍZENÍ BÁZE DAT	24
1.5.3 DATABÁZOVÁ APLIKACE.....	24
1.5.4 UŽIVATELÉ.....	24
1.6 NÁVRH RELAČNÍ DATABÁZE.....	25
1.6.1 DATOVÉ MODELOVÁNÍ.....	25
1.6.2 DATOVÝ E-R MODEL.....	25
1.6.3 E-R DIAGRAM.....	26
1.6.4 NORMALIZACE DATABÁZE	26
1.7 RELAČNÍ MODEL	28
1.7.1 RELACE	28
1.7.2 TYPY KLÍČŮ.....	29
1.7.3 HODNOTA NULL	29
1.7.4 FUNKČNÍ ZÁVISLOST	30
1.8 JAZYK SQL	30

1.9	MYSQL	31
1.9.1	DATOVÉ TYPY MYSQL	31
1.10	ZABEZPEČENÍ	32
1.10.1	SQL INJECTION	33
1.10.2	UKLÁDÁNÍ HESEL	33
1.11	PHPMYADMIN	33
1.12	PHPSTORM	34
1.13	XAMPP	34
1.14	LUCIDCHART	34
2	ANALÝZA EXISTUJÍCÍCH ŘEŠENÍ	35
2.1	WEBOVÁ PREZENTACE MOTOTECHNA GROUP A.S.	35
2.1.1	POPIS VZHLEDU A FUNKCIONALIT	35
2.2	WEBOVÁ PREZENTACE AAA AUTO INTERNATIONAL A.S.	36
2.2.1	POPIS VZHLEDU A FUNKCIONALIT	37
2.3	WEBOVÁ PREZENTACE AUTOESA A.S.	39
2.3.1	POPIS VZHLEDU A FUNKCIONALIT	39
3	ROZBOR A DEFINICE POŽADAVKŮ	41
3.1	POŽADAVKY NA DATABÁZI ŘEŠENÍ	41
3.2	VŠEOBECNÉ POŽADAVKY NA APLIKACI	42
3.2.1	PŘÍSTUPNOST OBSAHU	42
3.2.2	RESPONZIVNÍ WEBDESIGN	42
3.2.3	MODERNÍ WEBDESIGN	43
3.2.4	INTERAKTIVNÍ A INTUITIVNÍ PROSTŘEDÍ	43
3.3	POŽADAVKY NA OBSAH APLIKACE	44
II	PRAKTICKÁ ČÁST	46
4	DATABÁZOVÝ NÁVRH	47
4.1	RELACE PARAMETR	48
4.2	RELACE HOD_VARCHAR	48
4.3	RELACE HOD_TINYINT	49
4.4	RELACE HOD_SMALLINT	50
4.5	RELACE HOD_MEDIUMINT	51
4.6	RELACE HOD_DECIMAL	52
4.7	RELACE HOD_DATE	53
4.8	RELACE HOD_YEAR	54
4.9	RELACE VOZIDLO	55
4.10	RELACE MODEL	56
4.11	RELACE ZNACKA	57
4.12	RELACE KATEGORIE	58
4.13	RELACE POBOCKA	58
4.14	RELACE ADRESNI_MISTO	60
4.15	RELACE VOLNA_POZICE	61

4.16	RELACE POZICE	62
4.17	RELACE UZIVATEL	62
4.18	RELACE REZERVACE	63
4.19	RELACE OBLIBENE	64
4.20	RELACE PUVOD	65
4.21	RELACE NABIDKA	65
4.22	RELACE SMLOUVA.....	66
4.23	RELACE SLUZBA	68
4.24	RELACE PLATBA	68
4.25	RELACE ZAKAZNIK.....	69
4.26	RELACE SLUZBA_PRODAT	70
5	DESIGN APLIKACE	72
6	ČÁSTI APLIKACE	73
6.1	ZÁHLAVÍ APLIKACE	73
6.2	ZÁPATÍ APLIKACE.....	73
6.3	ÚVODNÍ STRANA	74
6.4	HLEDAT VŮZ.....	74
6.5	DETAIL VOZIDLA	76
6.6	PRODAT VŮZ.....	77
6.7	O NÁS.....	78
6.8	KONTAKT.....	79
6.9	PŘIHLÁŠENÍ A REGISTRACE UŽIVATELE	80
6.10	UŽIVATELSKÉ PROSTŘEDÍ.....	82
6.11	OSTATNÍ STRANY.....	83
7	ZABEZPEČENÍ APLIKACE.....	86
	ZÁVĚR	88
	SEZNAM POUŽITÉ LITERATURY.....	89
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	93
	SEZNAM OBRÁZKŮ	94
	SEZNAM TABULEK.....	96
	SEZNAM KÓDŮ	97
	SEZNAM PŘÍLOH.....	98

ÚVOD

Neodmyslitelnou součástí působení dnešních firem poskytující služby svým zákazníkům je jejich činnost na Internetu. Na vlastních webových stránkách se v dnešní době nahlíží jako na neodmyslitelnou část úspěšné firmy. V jednodušších případech se může jednat o webové stránky, poskytující pouze statické informace. V těch složitějších se potom jedná o rozsáhlé webové aplikace, u kterých se obsah přizpůsobuje potřebám uživatele. Někde mezi těmito dvěma případy se nachází populární internetové obchody.

Vzhledem k faktu, že trend nakupování na internetu již mnoho let roste, a analogický vývoj je předpokládán i do budoucna, vzniká potřeba přizpůsobovat tomuto trendu i nabízené služby, především jejich prezentaci. Tato „revoluce“ v nakupování se nevyhnula ani oblasti prodeje automobilů. Konkrétně u prodejců automobilů se může jednat o webové stránky poskytující informace o prodejci, které jsou založeny pouze z důvodu prezentace prodejce a poskytovaných služeb. Pro tento obor jsou však mnohem rozšířenější jednodušší webové aplikace, které umožňují uživateli zobrazovat výpis nabízených vozidel a další funkce.

Cílem práce je navrhnout databázovou strukturu, která je vhodná pro prodejce automobilů a zkonstruovat funkční webovou aplikaci, která může být použita pro prezentaci prodejce a bude implementovat i další funkce, které jsou relevantní pro daný subjekt. Před počátkem realizace je nutno provést analýzu existujících řešení a definici požadavků na výsledné řešení. Součástí teoretické části je popis technologií, které jsou využity při realizaci cíle práce.

Popis výsledku práce je předmětem praktické části. Vyobrazen je návrh struktury databáze, s bližším popisem konkrétních částí. Webová aplikace je představena graficky dle částí, které obsahují podrobnější popis. Součástí bakalářské práce je dokument ve formátu PDF, obsahující fyzický model databázové struktury a archiv, který obsahuje zdrojové kódy aplikace. Webová aplikace je dostupná online na adrese [HTTP://MAUTO-BP.WZ.CZ/](http://MAUTO-BP.WZ.CZ/).

I. TEORETICKÁ ČÁST

1 POUŽITÉ TECHNOLOGIE

K vytvoření aplikace a návrhu databáze byly použity níže popsané technologie. Popsány jsou technologie určené k vývoji webových aplikací, návrhu struktury databáze a programové vybavení, které bylo využito ke kompletaci projektu.

1.1 World Wide Web

World Wide Web (zkráceně WWW nebo web) označuje celosvětový systém určený k prohlížení, ukládání a odkazování dokumentů. Autorem systému je Tim Berners-Lee, který mimo jiné navrhl jazyk HTML, protokol HTTP určený pro přenos dokumentů a první webový prohlížeč s názvem WorldWideWeb. V roce 1990 byl spuštěn první webový server info.cern.ch. V roce 1994 bylo založeno mezinárodní konsorcium World Wide Web Consortium (zkráceně W3C), které Web vyvíjí do současnosti. [1]

1.2 Hypertext Markup Language

Značkovací jazyk známý pod zkratkou HTML je určený k vytváření obsahu pro webové stránky a aplikace v systému World Wide Web. Jedná se o otevřený formát standardizovaný mezinárodní normou ISO/IEC 15445:2000. Kód jazyka se ukládá v podobě prostého textu do dokumentu s příponou *.htm* nebo *.html*. Mimo kód jazyka mohou jeho dokumenty obsahovat i DTD¹ direktivy určené pro prohlížeče, kaskádové styly, kódy skriptů a komentáře kódu. Dokumenty obsahující kód HTML se umisťují na webový server. Uživatelé mohou k dokumentům přistupovat pomocí webového prohlížeče. [2],[3]

1.2.1 Vývoj HTML

První definice jazyka HTML vznikla v roce 1991. Byla součástí projektu World Wide Web, který byl realizován v CERNu². Autorem první definice jazyka je Tim Berners-Lee, který stojí za zrodem systému World Wide Web. V době vzniku bylo cílem jazyka umožnit vědcům sdílení informací a výsledků výzkumů po celém světě. První verze byla popsána v dokumentu s názvem HTML Tags. Umožňovala rozdělovat text do logických úrovní, zvýrazňovat ho, vkládat obrázky a odkazy. [2],[4]

¹ DTD – Document Type Definition (jazyk určený pro popis struktury XML nebo SGML dokumentu)

² CERN – Conseil Européen pour la recherche nucléaire (v překladu Evropská organizace pro jaderný výzkum)

Původní verzi rozšiřovali vývojáři webových prohlížečů o nové součásti. Vzhledem k zachování kompatibility mezi modifikacemi byl v roce 1994 vydán standard HTML 2.0, který obsahuje všechny v té době používané elementy. Obsahuje dvě úrovně. Úroveň 1 rozšiřuje původní verzi, úroveň 2 obsahuje definici formulářových prvků. [4][5]

Rozšíření známé pod označením HTML+ rozšiřovalo dosavadní standardy o možnost vytváření tabulek a kontrolovanější práci se vzhledem textu. V roce 1995 formalizoval Dave Raggett tohle rozšíření a vytvořil definici DTD (Document Type Definiton) v jazyce SGML³, čímž vznikl návrh nové verze – HTML 3.0. Úplnou podporu rozšíření, které tato verze přinesla neměly téměř žádné běžně používané prohlížeče. [4]

V roce 1996 mělo na starost vývoj standardů pro systém web již konsorcium W3C, mezi jehož členy patří přední softwarové společnosti. Následně byla vytvořena verze HTML 3.2, která nebyla až tak rozšířením verze 3.0, jako spíš rozšířením verze 2.0. Z verze HTML 3.0 přebírá pouze možnost vytváření tabulek. Ostatní nové součásti jsou prvky, které podporovaly tehdejší prohlížeče. Verze přinesla lepší kontrolu formátování a podporu Java appletů. V roce 1997 se verze 3.2 stala doporučením konsorcia W3C. Doporučení konsorcia si klade za cíl sjednotit všechny uživatele, aby využívaly stejnou verzi s cílem zajištění kompatibility. V době vydání a doporučení verze 3.2 již většina prohlížečů podporovala rámy (frames) a klientské skriptovací jazyky, jako je například Javascript. [4]

V roce 1997 zveřejnilo konsorcium dokument popisující návrh HTML 4.0 a následně ke konci roku ho přijalo jako standard. V roce 1999 byl vydán standard HTML 4.01, který opravoval drobné chyby, přinášel nové tagy (div, span) a deklaroval oddělení vzhledu od významu stránek, jelikož již počítal s existencí kaskádových stylů. [4]

Konsorcium v 90. letech 20. století vydalo standard XML, popisující značkovací jazyk sloužící k ukládání dat. Tento jazyk ovlivnil vývoj HTML tak, že v roce 2000 byla vydána verze XHTML 1.0. Možnosti HTML zůstaly nezměněny, pouze se upravila syntaxe, které v této specifikaci odpovídá jazyku XML (původní HTML odvozuje syntaxi od jazyka SGML). Později vznikla verze XHTML 1.1, která nepřinášela novou funkčnost, ale pouze popisovala možnost vytváření vlastních odvozenin. Problémem specifikace XHTML bylo, že ji správně nezpracovával v té době nejrozšířenější webový prohlížeč – Internet Explorer.

³ SGML – Standard Generalized Markup Language (metajazyk určený k definici jiných jazyků)

Zároveň specifikace nepřinášela nic zásadního. Nevznikal tak vážný důvod k přechodu od běžně používaného HTML. Následně začalo konsorcium vyvíjet novou verzi XHTML 2.0. Součástí nové verze bylo mnoho nových vlastností, za cenu porušení kompatibility s předchozími verzemi. V reakci na další rozvoj XHTML a jeho nekompatibilitu vyjádřili vývojáři webových prohlížečů svou nespokojenost, která dospěla k vytvoření společné pracovní skupiny pod zkratkou WHATWG (The Web Hypertext Application Technology Working Group). [4]

Od roku 2007 se sjednotila skupina WHATWG s konsorciem W3C. Sjednocením vznikla pracovní skupina, která s cílem vytvoření specifikace převzala návrh, který vznikl ve skupině WHATWG. Započal vývoj doposud poslední verze HTML5. Specifikace navazuje na verzi 4.01 a umožňuje používat syntaxi XML. Postupným vývojem HTML5 vzniká skutečnost, že mnoho nových funkcí je stabilně podporováno novými prohlížeči. [4]

1.2.2 Terminologie jazyka HTML

Způsob zápisu kódu je převzat z jazyka SGML. Skládá se ze značek neboli tagů a jejich vlastností, které se označují jako atributy. Atributy mohou nabývat hodnot, které se označují jako hodnoty atributů. Značky existují převážně párové, vyskytují se ale i nepárové. Název značky se zapisuje do špičatých závorek <značka>. Pro zápis párové značky se používá otevírací značka <značka> a ukončovací značka s povinným lomítkem před názvem ukončovací značky </značka>. Do párových značek se mohou vnořovat další značky. Nepárové značky nemají žádný obsah a nemají uzavírací značku. Zapisují se do závorek <značka />. Lomítko umístěné za značkou není povinné. [2]

Tabulka 1: Obecné atributy jazyka HTML

Atribut	Význam
class	třída elementu
id	identifikátor elementu
style	kaskádové styly elementu
lang	jazyk elementu

Část textu obsahující otevírací značku, obsah a uzavírací značku tvoří tzv. element. Pokud nemá žádný obsah, označuje se jako prázdný element, v opačném případě jako neprázdný. Zápis elementu může vypadat následovně: <tag atribut="hodnota-

atributu “>obsah-tagu</tag>. Elementy se před příchodem HTML5 dělily na řádkové a blokové. Nově je rozdělení přepracováno, a elementy se dělí na formulační, nadpisový a rozdělující obsah. [2]

1.2.3 HTML 5

Verze jazyka, která rozšiřuje doposud vydané verze o nové prvky. Vznikla jako reakce na potřeby dnešní doby, přičemž se snaží naplňovat současné potřeby i budoucí očekávání. Přináší nové a užitečné funkce a klade důraz na sémantiku kódu. Významným rozšířením je přidaná podpora přehrávání audio a video souborů bez nutnosti instalace doplňků. Návrh verze HTML5 byl vydán v roce 2007. Doporučením se první verze 5.0 stala v roce 2014. Nyní aktuální verze vyšla v prosinci roku 2017 s označením 5.2. Ve stejném roce byl vydán návrh verze 5.3. [2]

Tabulka 2: Subverze HTML 5 [6]

Verze	Návrh	Nominace na doporučení	Doporučení
HTML 5.0	2007	2012	2014
HTML 5.1	2012	2015	2016
HTML 5.2	2015	2017	2017
HTML 5.3	2017	-	-

1.2.4 Syntaxe HTML5

V ukázce je uvedena základní struktura dokumentu, která obsahuje kód napsaný v HTML5. První řádek – `<!DOCTYPE html>` informuje o použité verzi jazyka v dokumentu. Zápis informuje o použití HTML5. [2]

Do elementu `<html>`, který následuje po informaci o verzi dokumentu se zapisuje kompletně celý kód stránek, který je dále rozdělen na hlavičku dokumentu – element `<head>` a tělo – element `<body>`. V ukázce obsahuje element `<html>` atribut `lang` s hodnotou `cs`. Atribut definuje jazyk, který je v daném elementu používán. [2]

Obsah elementu `<head>`, je určen převážně webovým prohlížečům a vyhledávacím botům. Jediný obsah z elementu `<head>` viditelný uživatelem je obsah elementu `<title>`, do kterého se zapisuje text, který se zobrazuje jako název karty, popřípadě okna. Titulek je využíván i vyhledávacími roboty. Do elementu `<title>` nelze vnořovat další elementy. Dále je v elementu `<head>` definována znaková sada (standardně UTF-8) pomocí elementu

`<meta>`. V hlavičce se standardně uvádějí odkazy na externí soubory jako jsou kaskádové styly (element `<link>`) a JavaScriptové soubory (element `<script>`). Kaskádové styly i kód JavaScriptu lze zapsat i přímo v elementu `<head>`, v příslušném elementu. Do elementu `<body>` jsou vnořeny všechny elementy, popisující obsah viditelné části. [2]

```
<!DOCTYPE html>
<html lang="cs">
  <head>
    <meta charset="utf-8" />
    <title></title>
  </head>
  <body>
    <!--Obsah stránky-->
  </body>
</html>
```

Kód 1: Syntaxe HTML5

1.2.5 Sémantika v HTML 5

Oproti předchozím verzím jazyka je v HTML5 kladen důraz na sémantiku. Pomocí nových elementů je možno daleko přesněji popsat jednotlivé části a definovat tak jejich význam v rámci celého dokumentu. Obsah dokumentu je třeba označovat příslušnými elementy, a nikoliv jen za účelem změny vzhledu. Je třeba brát v úvahu, že dokument s kódem HTML slouží ke značení obsahu stránky. Stylování vzhledu stránky má za úkol dokument kaskádových stylů náležící stránce. [2]

Sémantické elementy

S příchodem HTML5 přibýly do jazyka nové sémantické elementy, které nahrazují doposud používaný element `<div>`. Element `<div>` nemá žádný sémantický význam a využívá se především k logickému rozdělování dokumentu kvůli stylování. Elementem `<div>` se v dřívějších verzích HTML označoval veškerý obsah, bez ohledu na jeho význam v dokumentu. HTML 5 přináší blokové elementy rozdělující obsah dokumentu na jednotlivé části, přičemž má každý svůj význam. Mezi tyto elementy patří elementy `<article>`, `<aside>`, `<header>`, `<footer>`, `<nav>`, `<section>`, `<figure>` a `<figcaption>`. [2][7]

Tabulka 3: Sémantické elementy [2][7]

Element	Označuje
<article>	článek
<aside>	dodatečný obsah (boční panel)
<header>	záhlaví elementu nebo celé stránky
<footer>	zápatí elementu nebo celé stránky
<nav>	navigace webu (menu)
<section>	sekce
<figure>	obrázek
<figcaption>	popisek elementu <figure>

Ostatní nové elementy

Mezi nové elementy se řadí elementy <bdi> a <bdo>, které mají své využití při kombinaci jazyků se zápisem zleva doprava a zprava doleva. Element <meter>, který má své uplatnění při označování částí ve známém rozsahu, například výsledků voleb či zaplnění diskového prostoru. Element pro specifikaci času <time>. Elementy <rp> a <rt> jsou dceřinými elementy elementu <ruby>, který definuje poznámku o výslovnosti, která se používá v některých asijských jazycích. Element <rt> specifikuje znaky ruby a pomocí <rp> lze zobrazit závorky na začátku a konci ruby v prohlížečích, které element <ruby> nepodporují. Element <wbr>, který je příbuzný elementu
. Na rozdíl od svého příbuzného elementu nevynucuje zalomení řádku, ale pouze doporučuje prohlížeči, kde by bylo ideální text zalomit. Element <progress> označuje indikátor průběhu, například při načítání obsahu či ukládání souboru. [2]

Nové formulářové prvky v podobě elementů <datalist>, který označuje skupinu vytvořených možností, které jsou následně nabízeny v <input> v podobě našeptávače. Dalším novým elementem je <output>, který označuje výsledky určité operace (například výstup z JavaScriptu). [8]

Mezi nové grafické elementy, vykreslující 2D grafický obsah, patří element <canvas>, který vykresluje obsah pomocí skriptu a element <svg>, který vykresluje obsah pomocí jazyka XML. [8]

Nově HTML obsahuje elementy popisující média. Zvukový obsah definuje element `<audio>` a element `<video>` popisuje video obsah. Element `<embed>` slouží jako kontejner pro externí aplikaci. Element `<source>` definuje zdroj pro elementy `<audio>` a `<video>`. Element `<track>` popisuje textové stopy náležící elementům `<audio>` a `<video>`. [8]

1.1 Cascading Style Sheets

V překladu Kaskádové styly je jazyk, který stylizuje dokumenty napsané v jazyce HTML do podoby, v které jsou prezentovány uživateli na displeji nebo jiném médiu. V 90. letech 20. století došlo k velkému rozšíření webových stránek. Společně s rozšířením začal vznikat důraz na výsledný vzhled dokumentu. Kontrola nad vzhledem byla ovlivněna novými atributy a rozšířeními, které si jednotlivé prohlížeče přidávaly. Tohle počínání mělo celkově špatný dopad. V HTML dokumentu se současně definoval význam i vzhled stránky a v případě potřeby následné úpravy vzhledu byla editace pracná a neefektivní. [2]

Kaskádové styly oddělují část stránek definující vzhled od významové části. Zároveň je jejich využití efektivní, jelikož mohou být použity pro více dokumentů a jedno pravidlo může být aplikováno na více elementů. Největší výhodou oddělení vzhledu od významu stránek je zvýšení přístupnosti stránek. Každý webový prohlížeč má své předdefinované kaskádové styly, které používá k zobrazování stránek. Výchozí kaskádové styly prohlížeče lze přepsat dokumentem kaskádových stylů, připojeným ke stránce. [2]

1.1.1 Vývoj CSS

Koncept jazyka byl navržen v roce 1994. Návrh provedl Håkon Wium Lie, v té době jako pracovník CERNu. V roce 1996 vznikla verze CSS 1, jako doporučení konsorcia W3C. Umožňovala stylovat vizuální podobu dokumentu HTML. Poslední revize CSS 1 byla provedena v roce 1999, kdy již většina prohlížečů podporovala téměř všechna doporučení této verze. Druhá verze kaskádových stylů byla vytvořena v roce 1997 a doporučením se stala rok poté. Specifikace řeší problémy předešlé verze a umožňuje definovat styl dokumentu pro různá média. [9]

Vývoj CSS3 byl zahájen v roce 1998 a probíhá dodnes. Oproti předešlým verzím, které jsou definovány jako jedna velká specifikace je popsán v několika dokumentech, které se nazývají moduly. Tento přístup je pro vývoj mnohem komfortnější a rychlejší. Jednotlivé moduly přidávají nové funkce nebo rozšiřují ty stávající. [9]

1.1.2 Aplikace CSS

Kód kaskádových stylů lze zapsat přímo do dokumentu HTML. Zápis se provádí do záhlaví dokumentu pomocí elementu `<style>`, nebo přímo k elementu formou atributu `style`. Nejefektivnější řešení využívané v reálných projektech spočívá v zápisu kódu popisující vzhled do externího souboru s příponou `.css`. Na externí dokument se v jazyce HTML odkazuje pomocí elementu `<link />`. [2]

1.1.3 Syntaxe jazyka CSS

Soubor kaskádových stylů se skládá z pravidel. Každé pravidlo je definováno selektorem a deklaračním blokem. Selektor ukazuje na element, na který se má pravidlo aplikovat. V deklaračním bloku jsou obsaženy konkrétní vlastnosti a jejich hodnoty, které se na selektor aplikují. [2]

```
element, .trida, #id{  
    vlastnost: hodnota;  
}
```

Kód 2: Syntaxe kaskádových stylů

Selektory

Cílem selektorů je jednoznačně ukazovat na elementy, na které se má aplikovat obsah deklaračního bloku, který selektoru náleží. V selektorech lze používat až pět různých kritérií pro výběr požadovaného elementu. Lze se odkazovat pomocí názvu elementu, podle kontextu, ve kterém se požadovaný element nachází, pomocí třídy či identifikátoru, pseudotřídy či pseudoelementu a pomocí atributů daných elementů a jejich hodnot. Všechny zmíněné způsoby odkazování na element je možné kombinovat. Selektory se využívají k odkazování na elementy i v Javascriptu. [2]

Deklarační blok

Zapisuje se za selektor nebo skupinu selektorů. Obsah deklaračního bloku se vkládá mezi složené závorky `{obsah-deklaračního-bloku}`. Hodnoty jednotlivých vlastností mohou nabývat textových, číselných nebo předem definovaných hodnot. [2]

1.1.4 Vlastnosti CSS 3

Třetí verze kaskádových stylů rozšiřuje předešlou verzi o několik nových vlastností. Vzhledem k faktu, že vývojáři webových prohlížečů implementují vlastnosti, které se doposud nestaly standardem, vznikla potřeba využívat tzv. vendor prefixy. Vývojáři prohlížečů používají prefixy k přidání podpory nových vlastností před tím, než budou plně podporovány ve všech prohlížečích, Vlastnosti obsahující vendor prefix slouží jako zkušební verze k odlišení aktuálně implementované verze od finální, která se může lišit chováním. Změna chování finální verze by mohla zapříčinit nežádoucí chování vzhledu stránek. V dnešní době již není využití prefixů tolik rozšířené a vývojáři webových prohlížečů pracují na jejich úplném odstavení. Prohlížeč Internet Explorer používá prefix `-ms-`, prohlížeč Firefox používá prefix `-moz-` a prohlížeče s jádrem Webkit používají prefix `-webkit-`. Starší verze prohlížeče Opera využívají prefix `-o-`. [2],[10]

Tabulka 4: Praktické CSS3 vlastnosti [2]

Vlastnost	Použití
<code>border-radius</code>	zakulacení rohů elementu
<code>text-shadow</code>	stín textu
<code>box-shadow</code>	stín objektu
<code>opacity</code>	průhlednost
<code>linear-gradient</code>	přechodné pozadí
<code>rotate</code>	rotace objektu

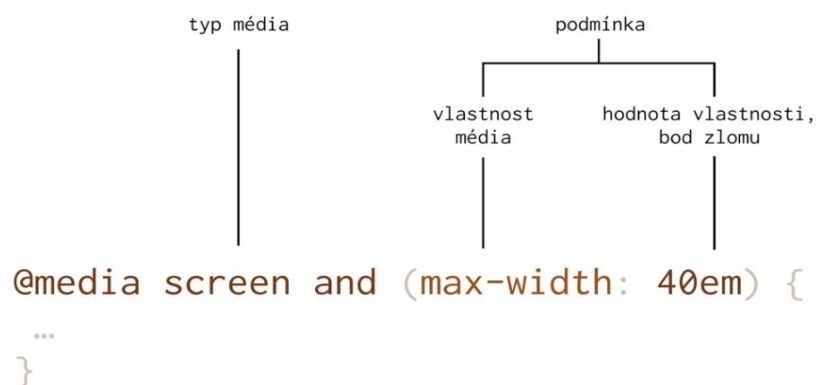
Třetí verze přidává mimo jiné i podporu nových typů selektorů a barevných modelů. Rozšiřuje stávající RGB model o alfa kanál⁴ pod zkratkou RGBA. Dále přidává možnost použití barevného modelu HSL a jeho rozšířené verze HSLA obsahující alfa kanál. [2]

1.1.5 CSS3 media queries

Pravidla, pomocí kterých je možno aplikovat rozdílná pravidla kaskádových stylů na různá zařízení dle technických aspektů (např. šířka okna). Media queries jsou nedílnou

⁴ Alfa kanál – složka pixelu nesoucí informaci o průhlednosti

součástí responzivního webdesignu. Dotaz na médium začíná zpravidla zápisem @media, který je následován podmínkami. Podmínky je možné modifikovat pomocí logických operátorů „and“ (a zároveň), čárkou (nebo), not (negace) a only (pouze). Podmínkami lze definovat pro jaké médium podmínky platí – screen pro displeje a print pro tisk stránky. Lze definovat mnoho dalších podmínek na základě technických parametrů zařízení. Nejpoužívanější podmínkou pro responzivní design je maximální šířka okna. Pravidla kaskádových stylů pro danou podmínkou se zapisují do deklaračního bloku umístěného za samotným dotazem. [11]



Obrázek 1: Syntaxe CSS3 media queries [12]

Tabulka 5: Základní podmínky CSS3 media queries

Název	Význam podmínky
width	Šířka viewportu
height	Výška viewportu
min-width	Minimální šířka viewportu
max-width	Maximální šířka viewportu
resolution	Hustota pixelů (v jednotkách dpi)
orientation	Orientace displeje (na výšku/na šířku)
aspect-ratio	Poměr šířky s výškou

1.2 Javascript

Klientský skriptovací jazyk, určený k tvorbě dynamického obsahu na webových stránkách. Jedná se o case sensitivní objektově orientovaný interpretovaný jazyk, který je multiplatformní. Disponuje podobnou syntaxí jako C/C++/Java, ale v podstatě je odlišný. Vznikl v roce 1995 a standardizován byl v roce 1997 asociací ECMA a rok později i asociací ISO. Vykonávání skriptu je řízeno událostmi vzniklými na webových stránkách. [13],[14]

Kód skriptu se do dokumentu HTML implementuje pomocí elementu `<script>`. Vykonávaný skript upravuje obsah stránky pomocí selektorů elementů. Javascript může být deaktivován ve webovém prohlížeči uživatele. Na tuto skutečnost by měl být brán při psaní kódu zřetel a funkcionalita stránek by měla být zachována i v případě jeho deaktivace.

1.3 Framework a knihovny

Použití knihoven si klade za cíl zjednodušit a urychlit vývoj projektu za pomoci připojených, již napsaných souborů, obsahující předem napsané funkcionality. Zpravidla se jedná o funkcionality, které se často opakovaně používají. S využitím knihoven se může vývoj konkrétních webových stránek soustředit na konkrétní problematiku zadání, a nikoli na všeobecnou, která je totožná u více projektů. U HTML, CSS, JS knihoven se jedná například o zjednodušení při přípravě responzivního webu, či různé knihovny pracující se vzhledem stránek. Velká knihovna nebo sada knihoven se označuje termínem framework.

1.3.1 Framework Bootstrap

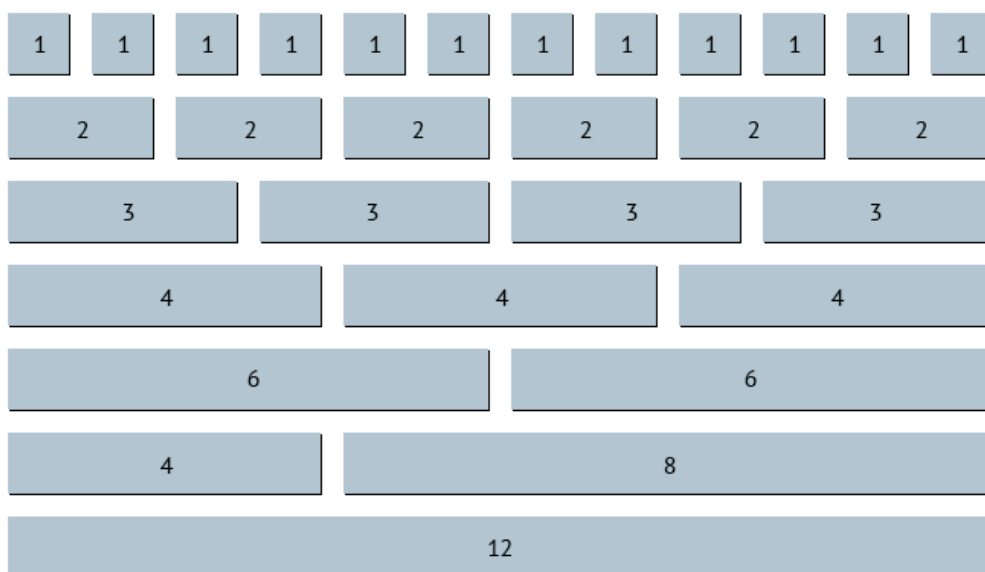
Open-source front-end framework, který vznikl reakcí zaměstnanců Twitteru Marka Otta a Jacoba Thortona na nekonzistentnost vzhledu a vysoké nároky na údržbu aplikací v rámci firmy. První verze byla vydána v roce 2011 pod názvem Twitter Blueprint a doposud poslední verze označená jako Bootstrap 4 byla vydána v roce 2018. Vytváří moderní vzhled stránek, je 100% responzivní a obsahuje grid systém (12-sloupcový). Od verze 3 podporuje přístup Mobile First. Bootstrap se k dokumentu připojuje pomocí odkazu na soubory (umístění v uložišti) nebo pomocí CDN⁵. Jednotlivé funkce se na elementy aplikují pomocí

⁵ CDN (Content delivery network) – síť propojených zařízení, která zpřístupňuje určitý obsah

definovaných tříd. Na internetu existuje mnoho volně dostupných šablon, které jsou na Bootstrapu založeny. [15]

Grid systém

Umožňuje rozčlenit obsah stránky do pravidelné mřížky sloupců a řádků. Využívá se z důvodu estetiky, ale také proto, že jsou takto zarovnané informace pro člověka mnohem lépe pochopitelné a zapamatovatelné. Všeobecně nahrazuje Grid systém dříve používaný tabulkový vzhled, který plnil podobnou funkci (obsah webu byl součástí buněk tabulky). Problém tabulkového vzhledu je, že není sémantický a na obsah tabulky se nahlíží jako na zdroj informací a zároveň není responzivní. Nedokáže se přizpůsobovat velikosti stránky a u malých zařízeních je nepoužitelný. Oba nedostatky řeší Grid systém, který se vytváří pomocí blokových elementů a tříd a je součástí nástroje Bootstrap. Obsah Grid systémů se v Bootstrapu vkládá do třídy *container* (popř. *container-fluid*). Ta se dále rozděluje na řádky (třída *row*) a do řádků se vkládají sloupce (třída *col-{počet sloupců}*). Existuje možnost definovat různý počet sloupců pro zařízení s různým rozlišením. [16]



Obrázek 2: Bootstrap grid systém [17]

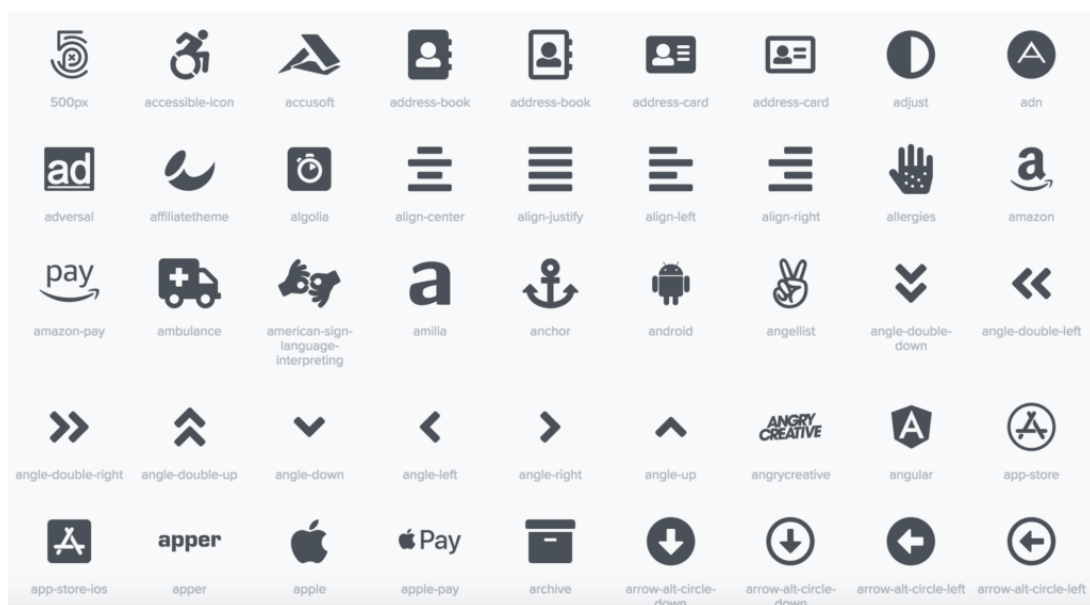
1.3.2 Knihovny Animate.css a Hover.css

Animate.css je jednoduchá CSS knihovna menšího rozsahu šířena pod licencí MIT, která pomocí tříd umožňuje animovat elementy. Poprvé byla zveřejněna v roce 2013 a stále se vyvíjí. Její tvůrce je Daniel Eden. Hover.css je knihovna určená především k animaci a

tvorbě přechodů tlačítek a jiných elementů. Obsahuje řadu 2D animací, které jsou určeny pro interakci s uživatelem. [18]

1.3.3 Knihovna Font-awesome

Knihovna postavená na technologii CSS a LESS, která obsahuje sadu fontů a ikon. Společně s knihovnou Google Fonts se řadí mezi nejpoužívanější knihovny, které jako třetí strana poskytují fonty písem a ikony. První verze byla vydána v roce 2012. Ikony je možné do elementů načítat pomocí tříd. Původně byla vydána pro použití s knihovnou Twitter Bootstrap, a později se stala součástí BootstrapCDN. Verze 5 byla vydána v roce 2017. Do čtvrté verze byla knihovna kompletně zdarma, od verze 5 je možné zdarma využívat verzi FREE, která obsahuje omezený počet ikon oproti verzi PRO, kterou lze využívat za poplatek. Implementace knihovny se provádí pomocí CDN. [19]



Obrázek 3: Ikony knihovny Font-awesome [20]

1.3.4 Knihovna jQuery

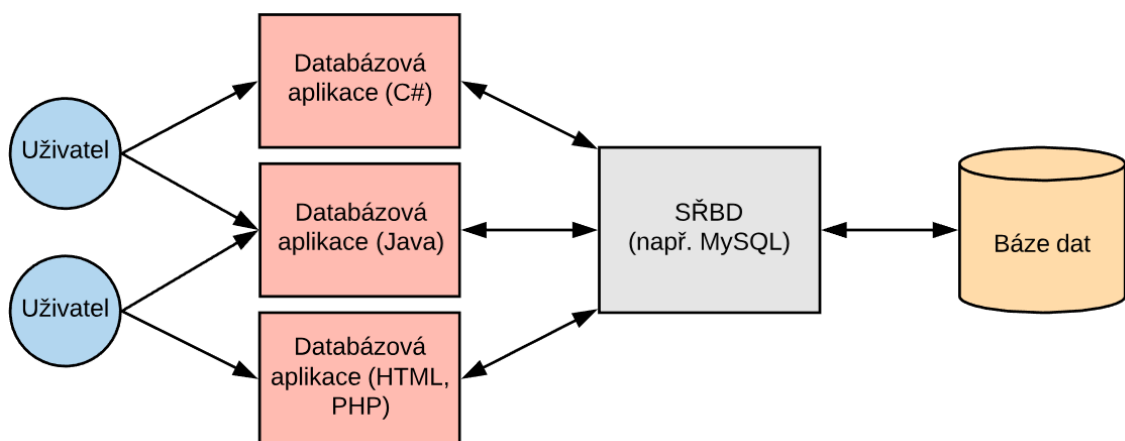
Javascriptová knihovna šířena pod licencí GPL a MIT. Disponuje jednodušší syntaxí oproti samotnému JavaScriptu. První verze byla představena v lednu roku 2006. Usnadňuje vytváření událostí a animací. Výhodou je možnost doinstalování zásuvných modulů. Spolupracuje s technologií AJAX, která klade důraz na interaktivitu webových stránek. Knihovna je využívána rozšířeným frameworkem Bootstrap. V dnešní době této knihovně představují konkurenci vícevrstvé frameworky jako React nebo Angular. [21]

1.4 Hypertext Preprocessor

Multiplatformní, serverový, skriptovací programovací jazyk známý pod zkratkou PHP, který je určen k vývoji dynamických webových stránek s možností vkladu do jazyka HTML. Vznikl v roce 1994 jako projekt s otevřeným zdrojovým kódem. Poslední verze s kompletně přepsaným jádrem disponuje pořadovým číslem 7 a byla vydána v roce 2015. Kód jazyka je interpretován webovým serverem. Do webového prohlížeče uživatele jsou odesílány pouze výsledky provedeného skriptu. Jedná se o jeden z nejpoužívanějších programovacích jazyků dynamických stránek. Odhaduje se, že pohání až 80% všech dynamických stránek. Syntaxí se podobá jazykům Perl, C a Java. Pro svůj výkon je využíván i ve velkých projektech jako je například Facebook a WordPress. Podporuje práci s databázovými systémy (např. MySQL, Oracle) a internetovými protokoly (např. HTTP, SMTP). [22],[23]

1.5 Databázový systém

Systém zahrnující uživatele, databázovou aplikaci, systém řízení báze dat a samotnou databázi (bázi dat). Báze dat obsahuje struktury tabulek a jejich obsah. Na jednom konci spektra existují osobní databázové systémy, které umožňují evidovat data v databázi běžným uživatelům (bez nutnosti znát jazyk SQL). Takový systém obsahuje databázovou aplikaci i SŘBD. Hojně využívaný je například program MS Access. Na druhém konci stojí databázové systémy podnikové třídy, které obsahují stovky tabulek, s velkým obsahem záznamů a umožňují souběžný přístup mnoha uživatelů. Data větších databázových systémů se ukládají na více discích na speciálních počítačích, které se nazývají databázové servery. [24]



Obrázek 4: Databázový systém

1.5.1 Báze dat

Kolekce vzájemně souvisejících dat a jejich popis. Konkrétně u relační databáze lze bázi dat definovat jako kolekci souvisejících tabulek a jejich popis. Součástí báze dat je vlastní popis, který obsahuje data o struktuře databáze označující se jako metadata (názvy tabulek, atributů, vlastnosti atd.), uživatelská data, indexy a další informace. Některé databáze uchovávají i aplikační metadata, jako jsou formuláře a sestavy (např. MS Access). [24]

1.5.2 Systém řízení báze dat

Zkráceně SŘBD je termín pro softwarový systém, který umožňuje řešit operace v databázi a pracovat s uloženými daty. Úkolem systému je tvorba, zpracování, správa báze dat. Systém přijímá příkazy od databázové aplikace, které transformuje na akce, které vykonává v bázi dat. Má na starost kontrolu souběžnosti přístupu uživatelů, a zamezuje kolidování přístupu jednoho uživatele s druhým. Mimo to disponuje bezpečnostním systémem pro autorizaci přistupujících uživatelů a omezení jejich rozsahu možných akcí s databází. V případě přerušení transakce v databázi umožňuje zotavení databáze do konzistentního stavu. V poslední řadě zajišťuje zálohu a možnost obnovení dat. Převažující systémy na trhu jsou Microsoft SQL Server, MySQL, Oracle Database a DB2. [24]

1.5.3 Databázová aplikace

Aplikace, která pracuje s databází na základě specifické aplikační logiky. Vytváří a zpracovává sestavy, formuláře a uživatelské dotazy. Vytváří dotazy (například pomocí formulářů) a odesílá je do SŘBD ke zpracování. Vrácená data, vytvořená na základě dotazu aplikace formuluje do tzv. sestav. Ty se zobrazují dle požadavků uživatele, nezávisle na struktuře databáze. Musí nabízet pouze logické možnosti a kontrolovat datové aktivity SŘBD (kompletně provedená změna, nebo žádná změna). Několik databázových aplikací, které jsou napsány v několika programovacích jazycích může přistupovat souběžně k SŘBD. U databázových systémů podnikové třídy se může jednat běžnou aplikaci napsanou v jazyce C# a webovou aplikaci napsanou pomocí jazyka HTML a PHP. [24]

1.5.4 Uživatelé

Lidé, kteří pomocí databázové aplikace přistupují k SŘBD a spravují data. Každý uživatel disponuje při přístupu k databázi svou rolí. Například správce databáze, může pracovat s mnohem větším spektrem akcí než běžný uživatel. [24]

1.6 Návrh relační databáze

Proces, který si klade za cíl realizovat funkční databázi, která je založena na relačním modelu. Samotný návrh obsahuje další dílčí procesy, kterými se při návrhu postupně prochází.

1.6.1 Datové modelování

Jedná se o proces, který naplňuje požadavky uživatelů a postupně modeluje strukturu databáze pro konkrétní případ. Postup je realizován od všeobecného modelu, ke konkrétnějšímu. Celý proces lze rozdělit do tří fází. V první fázi návrhu se provádí sběr informací a popis různých případů užití, na základě, kterých se provádí analýza a definice požadavků. **Fáze analýzy požadavků** je klíčová především proto, že hlubší zásah do struktury databáze v pokročilejších fázích životního cyklu databáze je komplikovaný, u rozsáhlejších systémů téměř nemožný. V reálném světě je špatný návrh předmětem nadbytečných nákladů časových a finančních. Vzhledem k této skutečnosti je nutno správně navrhnout strukturu databáze již v počátečních fázích projektu a minimalizovat tak potřebu zásahů v pokročilejších fázích. Druhou fází lze nazvat jako **fáze návrh komponent**. V této fázi je realizován převod datového modelu na databázový návrh. Součástí databázového návrhu jsou názvy relací, atributů, datové typy atributů, integritní omezení a popis primárních a cizích klíčů. V poslední fázi – **fázi implementace**, dochází k implementaci navržené databáze v konkrétním databázovém systému, importu dat do tabulek, tvorbě dotazů a sestav. Součástí implementační fáze je i testovací provoz, zavedení databáze do běžného užívání a školení uživatelů. Po implementaci je žádoucí databázi monitorovat, udržovat a dle potřeby aktualizovat. [24]

1.6.2 Datový E-R model

Zkratka pro Entity-relationship model, který slouží pro vyobrazení dat v softwarovém inženýrství. Tvoří základ pro návrh databáze. Obsahuje zjištěné požadavky na základě analýzy. Stejně jako u aplikací, kde se při návrhu využívá vývojový diagram se u databáze využívá datový model. Důležité součásti modelu jsou entity, atributy, identifikátory a vztahy. Entity vyobrazují nějaký objekt a stejný typ entit se seskupuje pod označením třída entit. Dále konkrétnější Instance entity popisuje konkrétní entitu dané třídy. Příkladem instance entity je konkrétní zákazník ve třídě Zákazník. Entity disponují atributy, které definují jejich vlastnosti. Například u entity Zákazník je předpoklad, že obsahuje atributy

jméno a příjmení. Každý atribut musí být popsán datovým typem, který definuje, o jaký typ dat se jedná (např. číslo nebo text). Dále může obsahovat vlastnosti jako výchozí hodnotu, povinnost zadání hodnoty a omezení. Dalším prvkem E-R modelu jsou identifikátory. Jedná se o atributy, které identifikují jednotlivé instance tříd. Mohou být složeny z jednoho nebo více atributů. V případě složení z více atributů se jedná o složený identifikátor. Mimo to mohou být identifikátory jedinečné a nejedinečné. Jedinečné identifikují jednoznačně označují právě jednu instanci třídy, naopak nejedinečné identifikují více instancí. Pomocí vztahů se definuje propojení mezi entitami. Dle počtu tříd entit ve vztahu se určuje stupeň vztahu. Vztah vzniklý mezi dvěma entitami se označuje jako binární a mezi třemi jako ternární. V binární vztazích mohou vznikat typy vztahů 1:1, 1:N a N:M. [24]

1.6.3 E-R diagram

Schéma popisující E-R model pomocí obdélníků, který reprezentuje entitu a kosočtverců vyobrazující vztahy mezi entitami. Součástí kosočtverců je zápis maximální kardinality⁶ vztahů. Minimální kardinalita⁷ vztahů je vyjádřena pomocí oválu a křížku vedle entit. [24]

1.6.4 Normalizace databáze

Normalizace databáze je proces, který si klade za cíl zjednodušit práci s daty, odstranit redundanci dat a vylepšit konzistenci dat v databázi. Jedná se o posloupnost kroků rozdělující jednu tabulku s více tématy na několik tak, aby každá obsahovala pouze jedno téma. Ve většině případů má normalizace více řešení – nemusí dopadnout vždy shodně. Normální formy jsou pravidla, které databáze v rámci procesu normalizace plní. Existuje několik úrovní normálních forem. [24] [25]

Jádro normalizace tvoří dva principy návrhu tzv. **relace ve správném formátu**:

- Každý determinant, musí být kandidátním klíčem.
- Každá relace ve špatném formátu musí být rozdělena na dvě a více relací.

⁶ Maximální kardinalita – nejvyšší možný počet entit účastnících se vztahu

⁷ Minimální kardinalita – nejmenší možný počet entit účastnících se vztahu

Proces normalizace zkoumá relace a upravuje je do správného formátu. Snaží se zamezit modifikačním chybám, které v tabulkách vznikají. Různé problémy, které může relace obsahovat se zařazují do různých typů normálních forem. [24]

Kroky normalizace

1. Identifikace všech kandidátních klíčů.
2. Identifikace všech funkčních závislostí.
3. Průzkum determinantů funkčních závislostí. (v případě nesplnění podmínky, že determinant musí být kandidátním klíčem – relace je v nesprávném formátu a následují kroky a-d)
 - a. Umístit sloupce FZ⁸ do nové relace.
 - b. Nastavit determinant FZ jako primární klíč.
 - c. V původní tabulce označit determinant jako cizí klíč.
 - d. Omezení referenční integrity mezi relacemi (stávající a novou).
4. Opakování třetího kroku dle potřeby.

Normální formy

První normální formě (1NF) odpovídá tabulka splňující definici relace. Tabulky v první normální formě mohou vykazovat určitý druh anomálií. Ve snaze odstranit anomálie byla definována **druhá normální forma** (2NF). Tabulka je v druhé normální formě, pokud je v první (1NF) a zároveň jsou všechny její neklíčové atributy určeny celým primárním klíčem. V praxi to znamená, že v případě, že je primární klíč složený, nemohou být zbylé neklíčové atributy určeny pouze částí klíče. Tabulka je v **třetí normální formě** (3NF), pokud je v 2NF a zároveň neobsahuje žádný neklíčový atribut, který by byl definován jiným neklíčovým atributem relace. Krátce po vydání 3NF bylo zjištěno, že i tabulky splňující tuto normální formu disponují vznikem anomálií. V reakci na tuto skutečnost byla vydána **Boyce-Coddova normální forma** (BCNF), která odstraňovala anomálie, které byly součástí 3NF. Tabulka se nachází v BCNF tehdy, pokud se nachází v 3NF a zároveň jsou všechny její determinanty zároveň kandidátními klíči. Všechny doposud zmíněné definice normálních forem jsou vytvořeny tak, že relace vyšší formy jsou zároveň vyhovující pro

⁸ FZ – funkční závislost

formu nižší. Později byl nalezen nový druh závislosti – vícehodnotová závislost. S cílem eliminace anomálií této závislosti je definována **čtvrtá normální forma (4NF)**. Tabulka splňuje 4NF, pokud všechny její vícenásobné hodnoty, které jsou určeny vícehodnotovou závislostí byly přesunuty do samotné tabulky. Další druh anomálie se týká problémů, který vzniká při rozdělení tabulek, které posléze nelze spojit zpět. Tento problém se snaží eliminovat **pátá normální forma (5NF)**. [24]

Každá normální forma se snaží eliminovat určitý druh anomálií, ale žádná neodstraňuje všechny. V roce 1981 definoval americký vědec a matematik Ronald Fag **doménovou/klíčovou normální formu (DK/NF)**, která k problematice přistupuje odlišným způsobem. Popisuje přístup, který říká, že by se před vytvořením relace měly hledat a splnit takové podmínky pro její vznik, které by předešly vzniku jakéhokoliv druhu anomálie. DK/NF se mezi odborníky příliš neujala. [24]

1.7 Relační model

Relační databázový model je založen na relační algebře. Byl popsán a publikován v roce 1970. Vyvinul jej Edgar Frank Codd, tehdejší zaměstnanec společnosti IBM. Model se v současnosti využívá při návrhu a implementaci většiny komerčních databází. Data ukládá ve formě relací. [24]

1.7.1 Relace

Speciální typ dvourozměrné tabulky, která je složena ze sloupců (atributů) a řádků (záznamy). Každá tabulka má v databázi jedinečný název. Aby se tabulka dala považovat za relaci, musí splňovat každou z těchto podmínek:

1. Každý řádek tabulky obsahuje data o entitě, nebo její části.
2. Každý sloupec tabulky obsahuje data, která představují atribut entity.
3. Každá buňka v tabulce, může uchovávat jedinou hodnotu.
4. Data v jednom sloupci musí být stejného druhu (datový typ).
5. Každý sloupec musí mít jedinečný název.
6. Nezáleží na pořadí řádků ani sloupců.
7. Každý záznam v tabulce musí být jedinečný. (v praxi se často ignoruje)

Tabulka 6: Ukázka relace

	Sloupec 1 (atribut)	Sloupec 2 (atribut)	...	Sloupec [n] (atribut)
Řádek 1 (záznam)				
Řádek 2 (záznam)				
...				
Řádek [n] (záznam)				

1.7.2 Typy klíčů

Klíč je jeden nebo více sloupců, pomocí kterých lze jednoznačně identifikovat záznam (řádek) v relaci. Může být jedinečný a nejedinečný. Jedinečný klíč jednoznačně identifikuje jeden záznam, naopak nejedinečný klíč může identifikovat více záznamů. Jako složený se označuje klíč, který je složen ze dvou a více atributů. Složené klíče mohou být jedinečné i nejedinečné. Kandidátní klíč jedinečně identifikuje záznam (řádek) v tabulce (může být i složený). Kandidátní klíč, který identifikuje všechny záznamy v tabulce se označuje jako Primární (PK). Náhradní klíč je označení pro atribut relace, který přiřadil systém řízení báze dat, aby plnil roli primárního klíče. Takový klíč je v ideální formě krátký, číselný a nikdy se nemění. Náhradní klíč bývá převážně generován automaticky systémem. Počáteční hodnota se označuje jako identity seed. Hodnota, která se přičítá k identity seed, aby vytvořila následující hodnotu se označuje jako identity increment (u MySQL funkce AUTO_INCREMENT, kde inkrement je vždy 1). Cizí klíč se využívá při vytváření vztahu mezi relacemi. Označuje se tak atribut jedné relace, který reprezentuje primární klíč relace druhé, s kterou vytváří vztah. V obou tabulkách musí mít klíče stejný datový typ. Tam, kde se vyskytuje cizí klíč, by se také mělo kontrolovat omezení referenční integrity. Například zamezení použití určité hodnoty v první tabulce, pokud se daná hodnota nevyskytuje v druhé. [24]

1.7.3 Hodnota NULL

Jedná se o chybějící buňku relaci. Hodnota null může mít více významů. Může znamenat, že hodnota doposud nebyla zadána (nebo se ji nehodí zadávat), že je neznámá, nebo že zadána ani být nemá. Hodnoty null je žádoucí co nejvíce eliminovat. Pomocí

databázových systémů lze určit, zdali atribut hodnotu null může obsahovat či nikoliv (NOT NULL). [24]

1.7.4 Funkční závislost

Vztah mezi atributy, pomocí kterého lze navrhnout schéma databáze a předcházet redundanci, nekonzistenci a problémům s vkládáním dat. Vztah, kde atribut A je funkčně závislý na atributu B, lze zapsat jako: $B \rightarrow A$. Atribut B se v tomto vztahu označuje jako determinant. Příklad z praxe: (CenaPoložky, PočetPoložek) \rightarrow CelkováCena. Zapsaný vztah popisuje, že celková cena je funkčně závislá na attributech ceny položky a počtu těchto položek. [24]

1.8 Jazyk SQL

Celým názvem Structured Query Language (v překladu Strukturovaný dotazovací jazyk) je textově orientovaný datový podjazyk, který slouží k práci s relačními databázemi. Vnikl v 70. letech 20. století ve společnosti IBM Corporation. Jazyk byl standardizován americkou organizací ANSI a později i organizací ISO. Pomocí konstrukcí jazyka lze definovat a zpracovávat data. Příkazy lze zadávat pomocí SŘBD. Pro využití řídicích konstrukcí a plně programátorských možnosti vzniká nutnost vkládat jej do tzv. hostitelského jazyka (programovacího jazyka), jako je například Java, C# nebo PHP. [24],[26]

Syntaxe SQL skriptu se skládá z tzv. klauzulí, které lze označit jako příkazy, a které definují, co má skript provést. Příkazy jazyka lze rozdělit do skupin Data definition language (DDL), Data manipulation language (DML), Data control language (DCL) a Transactional control language (TCL). Do skupiny DDL patří příkazy, které vytváří, upravují či mažou strukturu databáze. Jedná se například o klauzule CREATE, ALTER, DROP. Další skupinou je DML, do které patří příkazy, které slouží pro manipulaci s daty. Jsou to klauzule SELECT, INSERT, UPDATE, DELETE. Skupina DCL je skupinou, která je určena pro správu uživatelských rolí a práv. Jedná se příkazy GRANT a REVOKE. Poslední skupinou je TCL, do které patří příkazy jako BEGIN, COMMIT, ROLLBACK. Tyto příkazy slouží pro správu databázových transakcí. [26]

1.9 MySQL

System řízení báze dat relačních databází, který vyniká robustností a rychlostí. Jedná se o veřejně dostupný systém vydaný v roce 1995 určený především k ukládání dat webových stránek. Potencionálními konkurenty jsou PostgreSQL a MS SQL Server. Mezi jeho hlavní přednosti patří rychlost, jednoduchost konfigurace, přenositelnost mezi operačními systémy a dostupnost podpory. Je volně dostupný, v případě že není distribuován jako součást aplikace. V opačném případě je zapotřebí komerční licence s poplatkem. [27]

1.9.1 Datové typy MySQL

Při návrhu relací databáze je třeba brát v potaz, jaké data se do jednotlivých sloupců (atributů) budou ukládat. Existuje mnoho datových typů, které se liší v první řadě tím, zdali ukládají číselná data text či například datum. Datové typy vztahující se k danému typu dat (například číselné datové typy) se dále liší dle jejich rozsahu. Níže uvedené tabulky poukazují na nejpoužívanější datové typy v databázi MySQL včetně jejich rozsahů. [28]

Tabulka 7: Číselné datové typy [29]

Datový typ	Min. hodnota	Max. hodnota	Min. hodnota (unsigned)	Max. hodnota (unsigned)
TINYINT	-128	127	0	255
SMALLINT	-32768	32767	0	65535
MEDIUMINT	-8388608	8388607	0	16777215
INT	-2147483648	2147483647	0	4294967295
BIGINT	-2^{63}	$2^{63}-1$	0	$2^{64}-1$

V textech s kódováním ASCII platí, že jeden znak zabírá 1 byte. Tohle pravidlo ovšem nemusí platit pro kódování Unicode, kde může znak zabírat i více místa. Datový typ TEXT odpovídá dlouhému textu jako je například článek. [28]

Tabulka 8: Textové datové typy [28]

Datový typ	Rozsah
TINYTEXT	max. 255 B
TEXT	max. 64 KB
MEDIUMTEXT	max. 2^{24} B
LONGTEXT	max. 2^{32} B
VARCHAR([<i>max-počet-znaků</i>])	max. 64 KB
CHAR([<i>počet-znaků</i>])	max. 255 znaků

Mimo číselné a textové existují i ostatní datové typy, které najdou své uplatnění například při ukládání data nebo času. Tyto datové typy je nutno ukládat v předepsaném tvaru, který podporují.

Tabulka 9: Ostatní datové typy [28]

Datový typ	Rozsah
DATE	Datum v podobě textového řetězce ve tvaru ‚rrrr-mm-dd‘
TIME	Čas v podobě textového řetězce ve tvaru ‚hh:mm:ss‘
DATETIME	Zřetězení data a času ve tvaru ‚rrrr-mm-dd hh:mm:ss‘
TIMESTAMP	Chová se stejně jako DATETIME, ale reálně se jedná o počet uplynulých sekund od data 1.1.1970. Rozsah končí roku 2038.
BLOB	Umožňuje ukládat data v binárním tvaru, obrázky či zvuky.

1.10 Zabezpečení

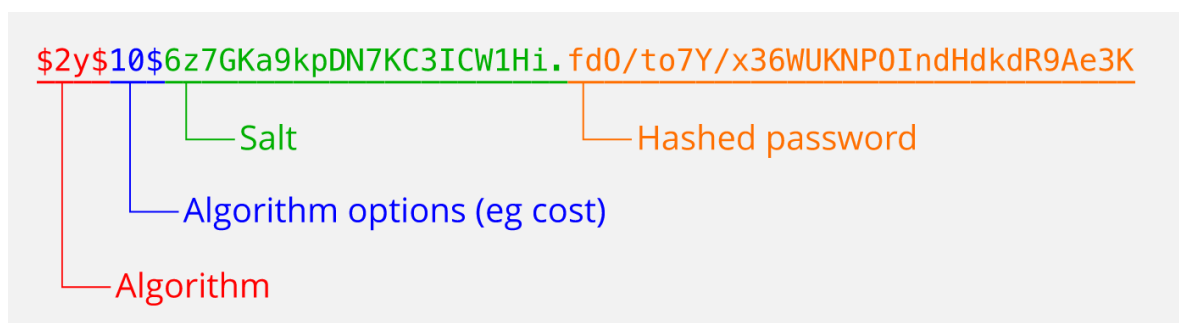
Webové aplikace bývá zpravidla přístupné mnoha uživatelům. Vzhledem k tomuto faktu je vystavena samotná aplikace i databáze možným útokům, kterým musí čelit. Existuje mnoho typů útoků, které si kladou za cíl například získat data z databáze či změnit obsah aplikace.

1.10.1 SQL injection

Technika napadení databáze, která se provádí tzv. vsunutím vlastního kódu přes neošetřený vstup. Útočník tímto počínáním upravuje výsledný dotaz na databázi, nejčastěji u klauzulí SELECT, INSERT, UPDATE, DELETE a WHERE. Nezabezpečené vstupy jsou jednou z nejčastějších problémů zabezpečení databáze. Problém byl v minulosti řešen tzv. escapováním, které ovšem neřeší tento problém stoprocentně. Proti tomuto útoku je správnou ochranou u jazyka PHP využití parametrizovaných dotazů (Prepared Statements). Ty jsou založeny na předávání dotazu a hodnot databázi zvlášť. Místo proměnných se do dotazu uvádějí zástupné znaky, za které si samotná databáze vkládá zvlášť přijaté hodnoty. [30]

1.10.2 Ukládání hesel

U aplikací, které umožňují registraci a přihlášení vzniká potřeba zabezpečovat uživatelská hesla. Hesla jsou zpravidla ukládána do databáze. Vzhledem k bezpečnosti se hesla do databáze neukládají ve stejné formě, v které byla zadána, ale před uložením se vytváří tzv. otisk (neboli hash), který je do databáze uložen. Pro vytvoření otisku hesla slouží v jazyce PHP funkce `password_hash()`, která disponuje parametrem samotného hesla a zvoleného algoritmu. Při přihlašování, nebo všeobecně všude tam, kde je potřeba ověřit heslo se využívá funkce `password_verify()`. [31]



Obrázek 5: Otisk hesla vytvořen algoritmem BCrypt [32]

1.11 phpMyAdmin

Volně dostupný nástroj vyvíjený v jazyce PHP. Je určený ke správě systému řízení báze dat MySQL pomocí webového rozhraní. Rozhraní provádět různé operace nad databází. Operace je možno provádět standardně pomocí jazyka SQL. Některé operace lze vykonávat pomocí grafického rozhraní (sktruktura relací, plnění dat).

1.12 PhpStorm

Multiplatformní komerční vývojářské prostředí napsané v jazyce Java, které vyvíjí společnost JetBrains s.r.o.. Je určeno k vývoji webových aplikací. Integruje funkce pro efektivní vývoj. Aplikaci lze rozšiřovat o zásuvné moduly.

1.13 XAMPP

Multiplatformní software vyvíjený skupinou Apache friends určený k vývoji webových stránek. Obsahuje webový server Apache, pomocí kterého lze interpretovat PHP skripty a databázový server MariaDB, ke kterému lze přistupovat pomocí nástroje phpMyAdmin. Umožňuje vytvořit server lokálně bez potřeby existence fyzického serveru.

1.14 Lucidchart

Webová platforma umožňující kreslení vývojových diagramů a vytváření dalších návrhů, jako jsou například datové modely databáze nebo drátěné modely aplikací. Nástroj umožňuje vyvázat diagramy přímo ve webovém prohlížeči a sdílet je dalším uživatelům.

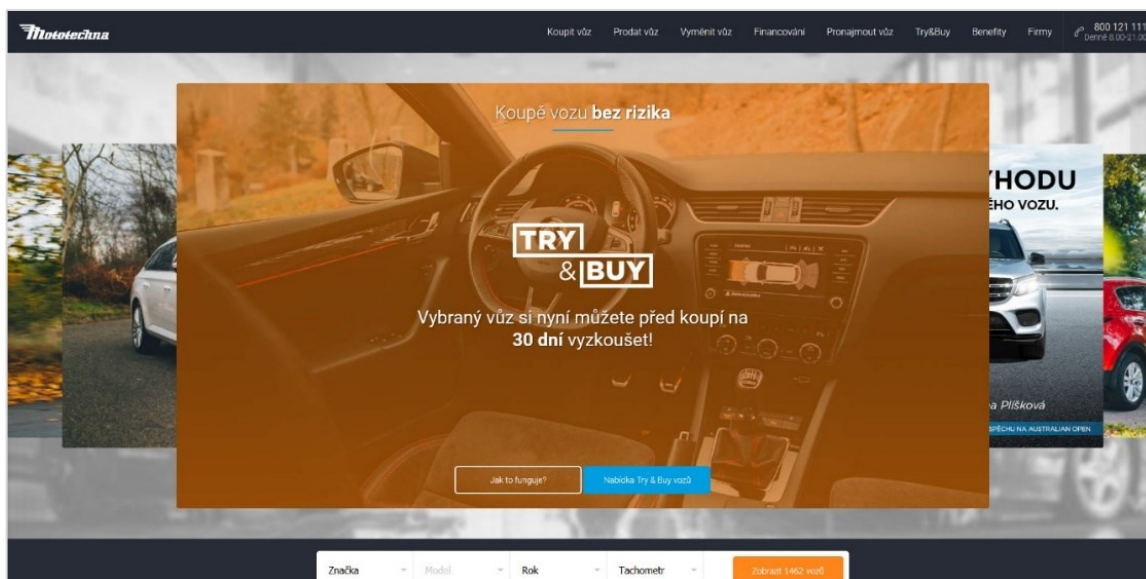
2 ANALÝZA EXISTUJÍCÍCH ŘEŠENÍ

Předmětem analýzy jsou webové prezentace prodejců automobilů působících v České republice. Analýza je zaměřena na webové prezentace společností Mototechna Group a.s., AAA auto International a.s. a AutoEsa a.s.. Součástí analýzy je stručný popis funkcionalit a vzhledu, který poskytují.

2.1 Webová prezentace Mototechna Group a.s.

Mototechna působila od počátku 50. let 20. století jako státní podnik, který se zabýval prodejem osobních automobilů, motocyklů a náhradních dílů. Od roku 2012 používá obchodní značku Mototechna společnost Mototechna Group a.s., která se zabývá prodejem automobilů, distribucí autodoplňků a nabídkou služeb autoservisu. [33]

Webové stránky Mototechny nabízí plně responzivní a moderní design. Využívají jednotný grafický design v barvách šedé, bílé, oranžové a modré. Prezentace je dostupná pouze v českém jazyce. [34]



Obrázek 6: Webová prezentace Mototechna Group a.s. [34]

2.1.1 Popis vzhledu a funkcionalit

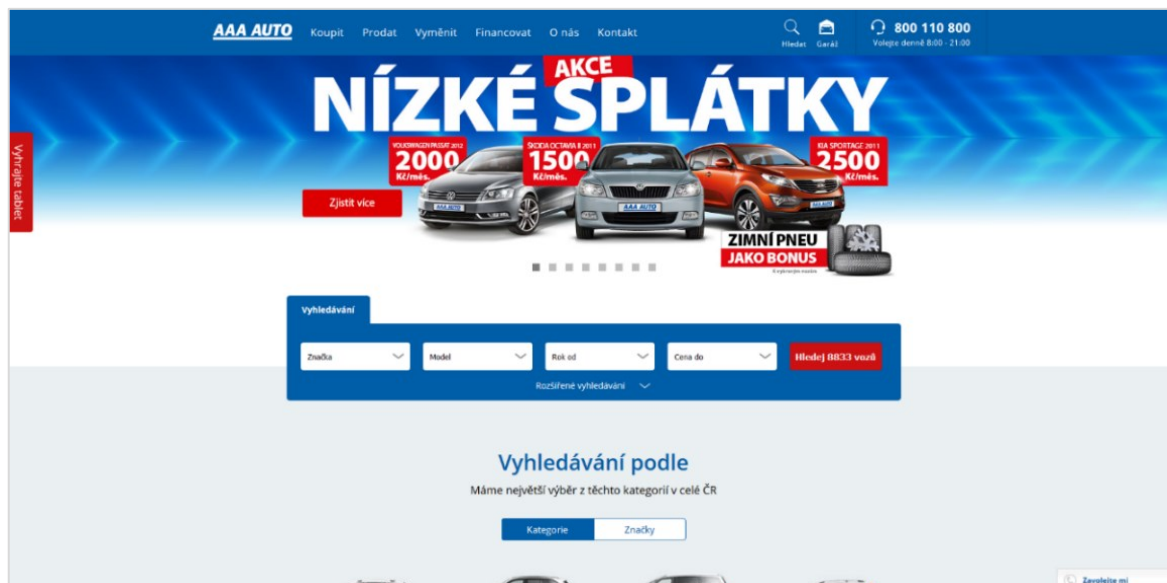
Webové stránky obsahují záhlaví, které je při posunu stránky zafixováno v horní části viditelné oblasti prohlížeče v neměnní se formě. Záhlaví obsahuje logotyp společnosti, navigační lištu a telefonní kontakt. Úvodní strana informuje o službách poskytovaných společností, referencích a obsahuje sekce s vozidly a rychlé vyhledávání automobilu na

základě základních parametrů. Navigační lišta obsahuje odkazy na strany Koupit vůz, Prodat vůz, Vyměnit vůz, Financování, Pronajmout vůz, Try&Buy, Benefity a Firmy. V položce Koupit vůz se nachází filtr určený k vyhledávání vozidla dle zadaných parametrů společně s výpisem vozidel. Položka Prodat vůz a Vyměnit vůz obsahuje totožný formulář určený k orientačnímu online ocenění vozidla a informace s danou službou spojené. Strana Financování se věnuje finančním službám, kterými Mototechna umožňuje financovat zakoupený vůz. Oproti jiným zmíněným prodejčům provozuje Mototechna službu pronájmu vozidla. Tomuto tématu se věnuje strana Pronajmout vůz, která obsahuje dostupné vozy, informace o službě a poptávkový formulář. Strana Try&Buy se věnuje stejnojmenné službě, která umožňuje zákazníkům si vůz nejprve vyzkoušet, a následně se rozhodnout o jeho koupi. Strana Benefity obsahuje reference a informace o společnosti se snahou přesvědčit zákazníka o kvalitních službách. Strana Firmy se zabývá službami, které jsou nabízeny firmám. Zápatí webové prezentace obsahuje odkazy na sociální síť Facebook a Instagram v podobě logotypů, copyright, logo společnosti a další odkazy na strany obsahující všeobecné obchodní podmínky, informace o servisu vozidel, kontakty a stranu Kariéra, která obsahuje nabídku volných pracovních míst. [34]

2.2 Webová prezentace AAA Auto International a.s.

Akciová společnost AAA Auto International, a.s. provozuje svou webovou prezentaci na adrese www.aaaauto.cz. Společnost je celorepublikovým prodejcem nových i ojetých automobilů. Na českém trhu působí od roku 2013 a řadí se mezi největší a nejznámější prodejce automobilů v Česku. [35]

Webová prezentace společnosti disponuje jednotným designem, který je využíván ve všech prezentačních a propagačních materiálech. Prezentace je založena na modrobílém motivu s příležitostnou přítomností červené barvy. Stránka podporuje 5 jazyků, mezi které patří čeština, slovenština, angličtina, polština a maďarština. [36]



Obrázek 7: Webová prezentace AAA Auto International a.s. [36]

2.2.1 Popis vzhledu a funkcionalit

Záhlaví stránek tvoří logo společnosti společně s navigační lištou, vyhledávacím polem, kontaktem a odkazem na uživatelské prostředí s názvem Garáž. Záhlaví je při posunu stránek zafixováno v horní části viditelné oblasti prohlížeče. Úvodní strana obsahuje výhodné nabídky společnosti, zjednodušený formulář vyhledávání automobilu a sekce obsahující reference, souhrn poskytovaných služeb, nejžádanější a doporučené vozy. Ke stránkám lze přistupovat pomocí klasické verze, která není responzivní a je určena pro desktopové zařízení s vyšším rozlišením a pomocí mobilní verze, která je určena pro zařízení s nižším rozlišením. Prezentace v mobilní verzi je dostupná na adrese m.aaaauto.cz. Na mobilní verzi jsou automaticky odkázáni uživatelé používající zařízení s nižším rozlišením. [36]

Navigační lišta obsahuje odkazy na strany Koupit, Prodat, Vyměnit, Financovat, O nás a Kontakt. Strana Koupit obsahuje filtrovaný výpis dostupných automobilů. Filtrovat je uživateli umožněno dle základních parametrů jako je značka, model, rok výroby či cena. Tento filtr lze rozšířit o další, podrobnější parametry (např. součásti výbavy). Strana Prodat informuje o způsobu a průběhu výkupu vozidel u společnosti. K dispozici je formulář určený k ocenění vozidla online. [36]

Značka Model Rok od Cena do **Hledej 8837 vozů**

Hlavní parametry

Hlavní parametry

Převodovka Palivo

Karoserie

Cena

od do Kč

Stav tachometru

od do km

Rok

od do

Ostatní

Pohon 4x4 Jen s odpočtem DPH

[Skrýt rozšířené vyhledávání](#)

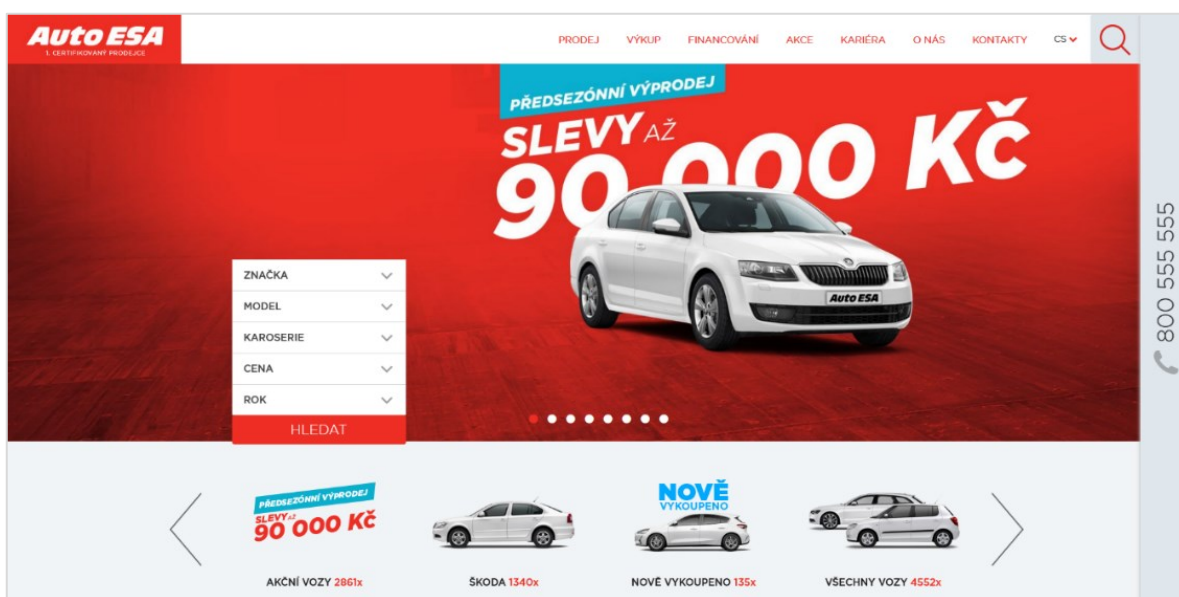
Obrázek 8: Filtr vyhledávání vozidel AAA Auto [36]

Strana Vyměnit informuje o možnosti výměny vozidla za dostupné vozidlo z nabídky včetně referencí. Strana Financovat informuje o možnostech financování. Strana O nás obsahuje informace o společnosti. Strana Kontakt obsahuje kontaktní údaje poboček. Součástí stránek je uživatelské prostředí s názvem Garáž. Uživateli je umožněna registrace pomocí emailu, jména a hesla nebo pomocí sociální sítě Facebook. Uživatelské prostředí umožňuje ukládat vozidla a vyhledávání. Na základě činnosti uživatele je generován seznam doporučených vozidel, který je dostupný v uživatelském prostředí. K dispozici je záložka s historií prohlížených vozidel a tzv. Servisní knížka, pomocí které si uživatel na základě čísla smlouvy a VIN kódu může do profilu přidat vůz, který v minulosti od společnosti zakoupil. Ve spodní části viditelné oblasti prohlížeče se nachází fixně umístěný box s textem Zavolejte mi, který pomocí formuláře umožňuje zákazníkovi zanechat své kontaktní údaje, v případě, že má zájem o to, aby ho společnost z nějakého důvodu kontaktovala. Zápatí prezentace obsahuje méně používané odkazy na strany poradny, kariéry u společnosti, péči o zákazníky a rychlé odkazy. Součástí zápatí je copyright, podmínky užití, nakládání s osobními údaji a odkaz na mobilní verzi webu a na distribuční kanály systémů iOS a Android, na kterých je dostupná nativní aplikace s podobnou funkcionalitou, kterou nabízí webová prezentace. [36]

2.3 Webová prezentace AutoEsa a.s.

Provozovatelem webové prezentace www.autoesa.cz je akciová společnost AutoEsa a.s. se sídlem v Praze, a více než dvaceti letou tradicí na trhu s vozidly v České republice. Společnost se zabývá nákupem a prodejem ojetých osobních a užitkových vozů. Provozuje celorepublikovou síť poboček a řadí se mezi největší autobazary v Česku. [37]

Webová prezentace je responzivní a obsahuje moderní, jednotný design. Podporuje 8 jazyků, mezi které patří čeština, slovenština, angličtina, němčina, francouzština, polština, ruština a vietnamština. [38]



Obrázek 9: Webová prezentace AutoEsa a.s. [38]

2.3.1 Popis vzhledu a funkcionalit

Záhlaví stránek utváří logotyp společnosti, navigační lišta, výběr jazyka a vyhledávací pole. Podstatnou část úvodní strany tvoří výhodné nabídky. Mimo to obsahuje úvodní strana vyhledávací formulář automobilů se základními parametry a sekce obsahující vozidla, reference a vyznačené pobočky na mapě České republiky. Strana Prodej umožňuje zobrazit

aktuální nabídku vozidel, pomocí formuláře zadat poptávku po vozidle a informovat se o náležitostech spojených s koupí vozidla ze strany zákazníka. [6]

ZÍSKEJTE GARANTOVANOU SLEUVU AŽ 30 000 Kč CHCI SLEUVU

Škoda Superb III 2.0 TSI L&K **SLEVA 31 000 Kč** koupeno nové v ČR / po prvním majiteli
2015 / 2.0 TSI / 162 kW / benzín / liftback / automat / bílá / stav tach. 35313 km / 5 míst / 5 dveří

Měsíčně od **2 166 Kč**
Spočítat splátky

Akční cena **568 999 Kč**
bez DPH 470 247 Kč

Původní cena 699 999 Kč
Cena v hotovosti **619 000 Kč**
bez DPH 511 570 Kč

Spočítat splátky Mám zájem Získat VIN Nezávazně rezervovat

Záruka až 36 měsíců

Obrázek 10: Detail vozidla AutoEsa [38]

Strana Výkup obsahuje informace spojené s výkupem vozidel zákazníků společnosti a formulář určený k ocenění automobilu online. Oceňování vozů se provádí na základě zadaných parametrů vozidla jako je značka, model, rok výroby, stav tachometru a palivo. Všechny hodnoty pro výběr jsou předem definované a uzavřené. Pod formulářem pro zápis parametrů o vozidle následuje formulář s kontaktními údaji, do kterého uživatel zadává jméno, emailovou adresu a telefonní číslo. Strana Kariéra odkazuje na adresu www.kariera.autoesa.cz. Strana Kariéra obsahuje formulář, pomocí kterého lze vyhledávat volné pracovní pozice. Strana O nás, obsahuje informace o společnosti a její partnerech. Obsahem jsou současně i reference prezentovány pomocí spokojených zákazníků. Strana Kontakty obsahuje seznam poboček včetně mapy, kontakt na zákaznické centrum, tiskového mluvčího, korespondenční adresu administrativního centra a sídlo společnosti, formulář Napište nám a formulář Reklamacce. Zápatí prezentace obsahuje copyright, odkazy na právní ujednání a ochranu osobních údajů, odkazy na sociální síť Facebook, Instagram, LinkedIn a kanál YouTube. [38]

3 ROZBOR A DEFINICE POŽADAVKŮ

Obsahem této kapitoly je rozbor požadavků, které vyplívají z analýzy existujících řešení a všeobecných trendů a jejich následná definice. Z definovaných požadavků je následně možno vycházet při realizaci řešení.

3.1 Požadavky na databázi řešení

Databáze slouží v případě tohoto řešení primárně jako uložení nabízených vozidel. Zbylý obsah aplikace databázi sám o sobě nevyžaduje. Seznam vozidel vyžaduje evidenci základních parametrů vozidel jako je značka, model, počet najetých kilometrů, palivo, barva, cena atd.. S přihlédnutím k nárokům dnešních uživatelů se zpravidla evidují, a do webové aplikace vypisují i ostatní údaje o vozidle jako typ karoserie, převodovky či jednotlivé součásti výbavy. Vzhledem rychlému technickému rozvoji automobilů, vzniká nutnost připravit databázi na možnost snadného přidávání nových parametrů, bez nutnosti hlubšího zásahu do struktury.

Aplikace, umožňující v případě zájmu o prodej vozidla odeslání parametrů vozidla pobočce vyžadují uložení dat do databáze pro možnost následného zobrazení příslušnou pobočkou. V případě zakomponování uživatelského prostředí do webové aplikace je nutno promítnout tuto skutečnost do databáze v podobě relací, které jsou určeny pro ukládání přihlašovacích údajů a ostatních informací o uživateli. Tato varianta intuitivně nabízí zpřístupnění možnosti rezervace vozidla či seznam oblíbených vozidel ze strany uživatele a uložení těchto hodnot do databáze pro další použití. Databáze v tomto případě musí spojovat relace vozidla a uživatele další relací, která obsahuje informace o rezervacích a oblíbených vozidlech. Mimo data, u kterých je nutnost ukládat je do databáze mohou být v databázi uloženy i jiné položky, jako například názvy poboček společnosti (ve variantě většího rozsahu), jejich adresy, kontaktní údaje, otevírací doba apod.. V případě nabídky volných pozic na webových stránkách je nutno vytvořit relace k jejich evidenci.

Databáze může sloužit mimo jiné i jako úložna dat, se kterými webová aplikace nepracuje, ale prodejce automobilů může mít nutnost je ukládat. Takovými daty se rozumí například seznam smluv či zákazníků.

3.2 Všeobecné požadavky na aplikaci

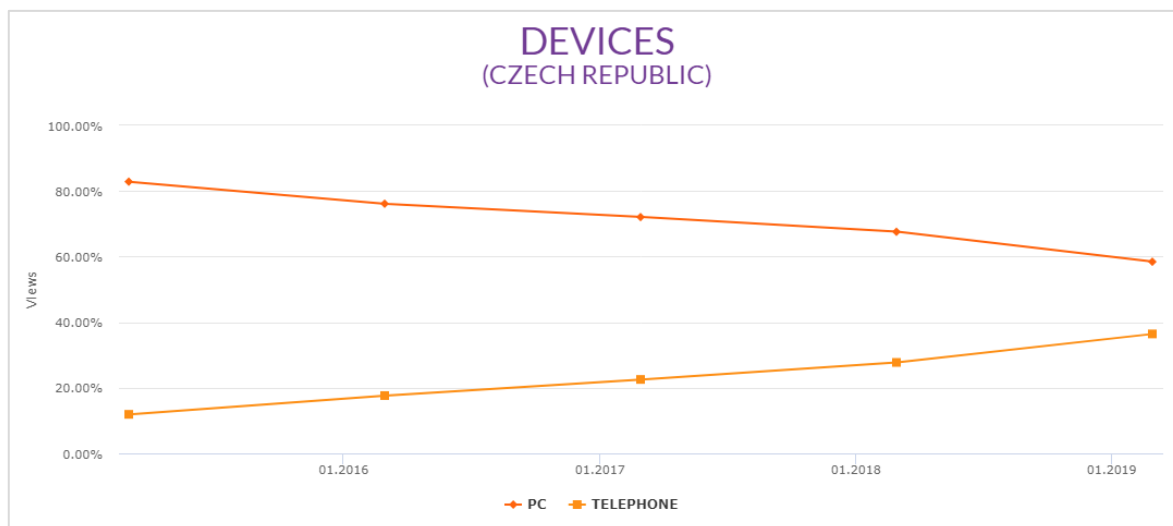
Požadavky, které jsou všeobecně vnímány jako součást kvalitních webových stránek. Jedná se o požadavky, které jsou dány trendem a ovlivňují celkový vývoj webových stránek a aplikací. Jsou totožné pro většinu projektů.

3.2.1 Přístupnost obsahu

Důležitý požadavek, který poukazuje na nutnost zpřístupnit obsah webových stránek všem uživatelům, kteří ke stránkám mají přístup bez ohledu na zařízení, které využívají, jak moc jsou zdatní v ovládání webových stránek jsou, jak rychlým připojením k internetu disponují v rámci možností atd.. Zobrazený obsah na jednom typu zařízení by měl být totožný (v podobném rozložení) i na ostatních typech zařízení. Prezentované informace by měly být strukturované a přehledné bez ohledu na typ zařízení.

3.2.2 Responzivní webdesign

Vzhledem ke stále rostoucímu využívání mobilních zařízení při prohlížení webových stránek vzniká nutnost přizpůsobovat web i mobilním zařízením. Responzivním webdesignem se rozumí takový webdesign, který dokáže přizpůsobovat rozložení a vzhled stránek pro zařízení, na kterém je stránka zobrazena, přičemž do všech zařízení je odesílána stejná verze HTML kódu a změny se provádí pomocí kaskádových stylů (popřípadě i JavaScriptu). Tento způsob je žádoucí oproti vytváření dvou webů, kdy jeden je určen pro zařízení s větší zobrazovací plochou (např. *web.com*) a druhý (např. *m.web.com*) pro mobilní zařízení. Způsob využití tzv. „m tečka“ webů je náročný na údržbu a celkově není vhodné tento způsob využívat, jelikož je totožný obsah webu přístupný na dvou adresách současně. Současně jsou weby, které nejsou responzivní degradovány nejpoužívanějším vyhledávačem – Googlem. [11]



Obrázek 11: Podíl zařízení na českém internetu [39]

Responzivita plní v dnešní době významnou roli při udržení uživatele na stránce a zpřístupnění všech dostupných funkcí na všech typech zařízení. Při vývoji responzivních stránek je žádoucí využít metodu psaní kódu zvanou Mobile first. Způsob psaní kódu metodou Mobile First spočívá ve vývoji kódu nejprve pro mobilní zařízení, a až následně přizpůsobovat kód pro větší zařízení. Pro vytvoření responzivního web designu je vhodné použít některý z nástrojů, který usnadňuje práci s webem. Typickým nástrojem je například framework Bootstrap. [13]

3.2.3 Moderní webdesign

Významnou estetickou roli hraje u webových prezentací jednotný grafický design, který je pro jednotlivé části stránky stejný (v rámci možností podobný). Jednotným grafickým designem se rozumí využití vybrané škály barev, stejného písma a rozvržení elementů na stránce. Trendem dnešní doby je jednoduchost, hranatý styl a stručný text.

3.2.4 Interaktivní a intuitivní prostředí

S rozvojem webových technologií lze téměř identicky vyvinout webovou aplikaci v porovnání s desktopovou (např. Office365). Prostředí by mělo působit interaktivně. Vzhledem k převažujícímu využívání mobilních zařízení však není vhodné animovat všechny prvky na stránce. Například animace aktivovaná při použití polohovacího zařízení je na mobilním zařízení prakticky nevyužitá a v neposlední řadě může zhoršit výkon zařízení. Vhodně aplikované interaktivní prvky mohou zvyšovat uživatelský zážitek a činit webové stránky uživatelsky přívětivějšími a zajímavějšími. Rozvržení stránek by mělo

uživatelé intuitivně navádět k tomu, co chce udělat. Stejně jako u responzivního designu zde platí pravidlo: „Zpřístupnit webové stránky všem uživatelům.“, ve smyslu přizpůsobení tak, aby aplikaci zvládl ovládat i méně zdatný uživatel.

3.3 Požadavky na obsah aplikace

Hlavním cílem webové aplikace prodejců automobilů je nabídka vozidel a prezentace informací o prodejci. Na základě analýzy by měla aplikace disponovat možností výpisu dostupných vozidel, možností projevit zájem o prodej vozidla ze strany uživatele a informacemi, popřípadě referencemi o prodejci. Aplikace může zahrnovat i doplňující obsah, jako například výpis volných pozic, tipy pro zákazníky, možnost stažení dokumentů atd.. Umožňovat pohyb mezi jednotlivými částmi aplikace by měla umožňovat hlavní navigace, která by měla obsahovat důležité odkazy, a informovat uživatele, v které části se nachází. Nejvhodnější umístění navigace u tohoto typu aplikace je do záhlaví. Většina prodejců umísťuje do záhlaví i logotyp společnosti. Jako převážná většina webových aplikací, by výsledná verze měla obsahovat úvodní stranu, která si klade za cíl zaujmout. Obsahovat by měla pouze základní a stručné informace nebo shrnovat hlavní funkcionality.

Výpis vozidel by měl umožňovat filtrování vozidel na základě alespoň základních parametrů, ideálně těch nejdůležitějších pro uživatele. Mezi ty nejdůležitější potom patří značka, model, rok výroby a cena. Výpis automobilů by měl být spíše stručnějšího charakteru, například formou seznamu. Položky obsahující jednotlivá vozidla by měly obsahovat stručné informace o vozidle. Je žádoucí, aby položky shrnovaly klíčové údaje jako je stav tachometru, rok výroby, palivo. V případě většího prodejce i pobočku, na které se vůz nachází. Položky prezentující vozidla ve výpisu by měly utvářet odkaz na detailnější stranu konkrétního automobilu, která obsahuje více informací. Detailnější strana vozidla by měla obsahovat rozvedení stručných informací a doplnění těch méně důležitých, jako jsou detailnější technické parametry a informace o součástech výbavy. Je žádoucí, aby obsahovala minimálně jednu fotografii vozidla. Ideální je ale galerie obsahující více fotografií, umožňující si automobil prohlédnout z více úhlů, popřípadě nahlédnout do interiéru.

Dalším hlavním cílem je umožňovat prodat vozidlo (odkoupit od zákazníka). Na první pohled funkcionalita, která stejně jako možnost koupit vozidlo nejde uskutečnit pouze přes internet. Možnost výkupu vozidla lze řešit několika způsoby. Do části aplikace, věnující se prodeji vozidla lze umístit kontakty a informace o průběhu výkupu. V případě očekávání

výkupu většího množství vozidel, je žádoucí první kontakt částečně zautomatizovat. Automatizací se rozumí poskytnutí formuláře uživateli pro vyplnění základních parametrů o vozidle a zanechání kontaktu. Tento proces umožňuje lepší zpětnou reakci prodejce. Výhodu má tato automatizace například v tom, že v případě několika poboček lze odesílat údaje na konkrétní pobočky, kompetentním osobám.

Obsah aplikace by měl zahrnovat informace o prodejci, kontaktní údaje a reference. Jedná se o obsah informativního charakteru. Součástí informací může být například otevírací doba, adresa (mapa), nabízené služby, historie společnosti a specializace. Kontaktní údaje by měly obsahovat více možností, jako například kontaktovat prodejce pomocí telefonního čísla nebo emailové adresy. V případě prodejců většího rozsahu je žádoucí uvést kontaktní údaje například konkrétních oddělení apod.

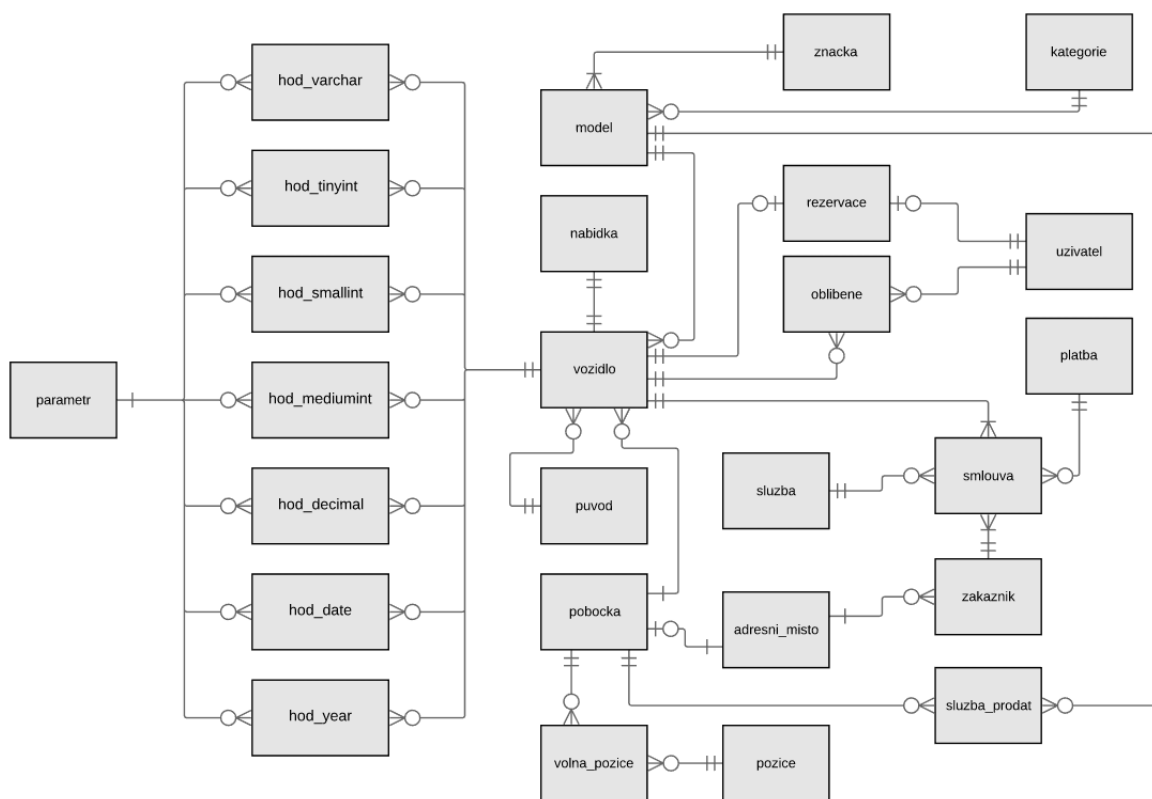
U aplikací, které umožňují registraci a následné přihlášení uživatele je očekávaná přidaná hodnota vyplývající z registrace. U webových aplikací prodejců automobilů se může jednat například o možnost uskutečnit online rezervaci vozidla, vytvoření seznamu oblíbených vozidel nebo usnadnění ostatních interakcí s aplikací. Aplikace s těmito možnostmi by měla nabízet uživatelské prostředí shrnující údaje o uživateli, související skutečnosti a možnost správy účtu (změnu údajů).

Zápatí webových stránek může obsahovat další navigaci, obsahující méně důležité odkazy. Do zápatí se v některých případech umísťuje mapa web a další užitečné odkazy. Užitečné odkazy mohou odkazovat na méně důležité součásti aplikace, jako je například výpis volných pozic, tipy související s prodejem či koupí vozidla, dokumenty ke stažení a ostatní odkazy, které nelze považovat za natolik důležité, aby byly umístěny v hlavní navigaci. Webová aplikace se může lišit dle nabízených služeb, zaměření a rozsahu působnosti prodejce. Někteří prodejci nabízejí i jiné služby než prodej a výkup (např. zkouška či pronájem vozidla).

II. PRAKTICKÁ ČÁST

4 DATABÁZOVÝ NÁVRH

Návrh databázové struktury obsahuje 26 relací. Umožňuje evidovat automobily včetně parametrů, značky, modelu, kategorie a jejich původ. Dále data o tom, v jakém časovém úseku byl nebo je automobil v nabídce, a zdali je nabídka automobilu aktivní. Návrh obsahuje relace určené k uchovávání uživatelských dat, seznamu rezervací a seznamu oblíbených vozidel. Součástí je relace určená k evidenci adresních míst České republiky, která je využitelná v mnoha ohledech, jako například u adresy pobočky či zákazníka, a v případě potřeby i všude tam, kde je potřeba nějakým způsobem pracovat s adresami. Obsahem návrhu jsou i relace určené k uložení dat o pobočkách, ke kterým se vztahuje evidence volných pracovních pozic. V rámci evidence volných pracovních pozic existuje relace určená k ukládání názvů pracovních pozic, které se mohou opakovat. Taktéž je zohledněno ukládání dat z online formuláře, který je určen k výkupu automobilu od zákazníka. Návrh obsahuje i relace, které se přímo nevztahují k webové prezentaci, jako například relace, která eviduje zákazníky, seznam uzavřených smluv, seznam služeb a možnosti platby, které se pojí s již uskutečněnými prodeji.



Obrázek 12: ER diagram

4.1 Relace parametr

Relace parametr je popsána atributem *parametr_id*, který slouží jako primární klíč relace a atributem *parametr_nazev*, který je určen pro uložení textové hodnoty označující daný parametr (např. karoserie, palivo atd.).

parametr		
PK	parametr_id	int(11) unsigned not null
	parametr_nazev	varchar(100) not null

Obrázek 13: Relace parametr

```
CREATE TABLE `parametr` (
  `parametr_id` int(11) unsigned AUTO_INCREMENT not null,
  `parametr_nazev` varchar(100) not null,
  PRIMARY KEY (`parametr_id`)
);
```

Kód 3: Vytvoření relace parametr

4.2 Relace hod_varchar

Slouží pro uložení textových hodnot parametrů vozidla. Příkladem může být například parametr karoserie a jeho hodnota „kombi“. Relace je popsána atributem *hod_varchar_id*, který je primárním klíčem relace, atributem *hod_varchar* datového typu varchar, který je určen k uložení samotné hodnoty, atributem *hod_varchar_par*, který je cizím klíčem atributu *parametr_id* relace parametr a přiděluje dané hodnotě parametr. Posledním atributem relace je *hod_varchar_vozidlo*. Jedná se o cizí klíč odkazující na atribut *vozidlo_id*, relace vozidlo, a přiděluje danou hodnotu parametru určitému vozidlu.

hod_varchar		
PK	hod_varchar_id	int(11) unsigned not null
	hod_varchar	varchar(50) not null
FK	hod_varchar_par	int(11) unsigned not null
FK	hod_varchar_vozidlo	int(11) unsigned not null

Obrázek 14: Relace hod_varchar

```
CREATE TABLE `hod_varchar` (
  `hod_varchar_id` int(11) unsigned AUTO_INCREMENT not null,
  `hod_varchar` varchar(50) not null,
  `hod_varchar_par` int(11) unsigned not null,
  `hod_varchar_vozidlo` int(11) unsigned not null,
  PRIMARY KEY (`hod_varchar_id`),
  KEY `FK` (`hod_varchar_par`, `hod_varchar_vozidlo`)
);
```

Kód 4: Vytvoření relace hod_varchar

4.3 Relace hod_tinyint

Slouží pro ukládání číselných hodnot parametrů v rozsahu od 0 do 255. Relace je určena i pro ukládání hodnot 0/1 (TRUE/FALSE). Příkladem může být uložení hodnot parametru počtu airbagů (nepředpokládá se více než 255 airbagů v jednom automobilu), či pravdivostní hodnoty u parametru servisní knihy (automobil má – 1, nemá – 0) a jemu podobných. Relace je popsána atributem *hod_tinyint_id*, který slouží jako primární klíč. Další atribut – *hod_tinyint* datového typu tinyint umožňuje uložení samotné hodnoty, atribut *hod_tinyint_par* je cizím klíčem atributu *parametr_id*, relace parametr, přičemž odkazuje na konkrétní parametr. Atribut *hod_tinyint_vozidlo* je cizím klíčem atributu *vozidlo_id*, a přiřazuje danou hodnotu a parametr vozidlu.

hod_tinyint		
PK	hod_tinyint_id	int(11) unsigned not null
	hod_tinyint	tinyint(3) unsigned not null
FK	hod_tinyint_par	int(11) unsigned not null
FK	hod_tinyint_vozidlo	int(11) unsigned not null

Obrázek 15: Relace hod_tinyint

```
CREATE TABLE `hod_tinyint` (
  `hod_tinyint_id` int(11) unsigned AUTO_INCREMENT not null,
  `hod_tinyint` tinyint(3) unsigned not null,
  `hod_tinyint_par` int(11) unsigned not null,
  `hod_tinyint_vozidlo` int(11) unsigned not null,
  PRIMARY KEY (`hod_tinyint_id`),
  KEY `FK` (`hod_tinyint_par`, `hod_tinyint_vozidlo`)
);
```

Kód 5: Vytvoření relace hod_tinyint

4.4 Relace hod_smallint

Relace hod_smallint umožňuje evidovat číselné hodnoty parametrů, které jsou definovány v rozsahu od 0 do 65535. Může se jednat například o výkon vozidla v kilowatech (např. 260 kW). Roli primárního klíče relace plní atribut *hod_smallint_id*. Atribut *hod_smallint* popsaný datovým typem smallint slouží k ukládání příslušné hodnoty, atribut *hod_smallint_par* je cizím klíčem atributu *parametr_id*, relace parametr a přiřazuje dané hodnotě určitý parametr. Atribut *hod_smallint_vozidlo* je cizí klíč atributu *vozidlo_id*, relace vozidlo a přiřazuje tak hodnotu a parametr konkrétnímu vozidlu.

hod_smallint		
PK	hod_smallint_id	int(11) unsigned not null
	hod_smallint	smallint(5) unsigned not null
FK	hod_smallint_par	int(11) unsigned not null
FK	hod_smallint_vozidlo	int(11) unsigned not null

Obrázek 16: Relace hod_smallint

```
CREATE TABLE `hod_smallint` (
  `hod_smallint_id` int(11) unsigned AUTO_INCREMENT not null,
  `hod_smallint` smallint(5) unsigned not null,
  `hod_smallint_par` int(11) unsigned not null,
  `hod_smallint_vozidlo` int(11) unsigned not null,
  PRIMARY KEY (`hod_smallint_id`),
  KEY `FK` (`hod_smallint_par`, `hod_smallint_vozidlo`)
);
```

Kód 6: Vytvoření relace hod_smallint

4.5 Relace hod_mediumint

Totožně jako u předešlých relací, je tato relace určena k evidenci číselných hodnot. Oproti předešlým relacím je určena k evidenci číselných hodnot v rozsahu od 0 do 16777215. Tento rozsah může sloužit ukládání hodnot pro parametry jako je stav tachometru automobilu či cena automobilu. Atribut *hod_mediumint_id* představuje primární klíč relace, atribut *hod_mediumint* slouží k ukládání hodnot datového typu *mediumint*, atribut *hod_mediumint_par* je cizím klíčem atributu *parametr_id*, relace *parametr* a atribut *hod_mediumint_vozidlo* je cizím klíčem atributu *vozidlo_id*, relace *vozidlo*. Oba cizí klíče představují stejnou funkci jako u výše zmíněných relací určených k ukládání číselných hodnot.

hod_mediumint		
PK	hod_mediumint_id	int(11) unsigned not null
	hod_mediumint	mediumint(8) unsigned not null
FK	hod_mediumint_par	int(11) unsigned not null
FK	hod_mediumint_vozidlo	int(11) unsigned not null

Obrázek 17: Relace hod_mediumint

```
CREATE TABLE `hod_mediumint` (
  `hod_mediumint_id` int(11) unsigned AUTO_INCREMENT not null,
  `hod_mediumint` mediumint(8) unsigned not null,
  `hod_mediumint_par` int(11) unsigned not null,
  `hod_mediumint_vozidlo` int(11) unsigned not null,
  PRIMARY KEY (`hod_mediumint_id`),
  KEY `FK` (`hod_mediumint_par`, `hod_mediumint_vozidlo`)
);
```

Kód 7: Vytvoření relace hod_mediumint

4.6 Relace hod_decimal

Relace `hod_decimal` slouží k ukládání číselných hodnot pro parametry, u kterých se nachází v čísle desetinná čárka. U vozidel představuje takovou hodnotu významná hodnota parametru obsahu automobilu (1.4 ccm; 1.9 ccm; 2.0 ccm atd.). Relace může být použita i pro jiné parametry u kterých obsahuje hodnota desetinnou čárku. Primární klíč relace představuje atribut `hod_decimal_id`. Atribut `hod_decimal` je určen k ukládání hodnoty datového typu `decimal`. Cizí klíč `hod_decimal_par` odkazuje na atribut `parametr_id`, který označuje určitý parametr a atribut `hod_decimal_vozidlo` přiřazuje hodnotu a parametr vozidlu jako cizí klíč atributu `vozidlo_id`, relace `vozidlo`.

hod_decimal		
PK	hod_decimal_id	int(11) unsigned not null
	hod_decimal	decimal(2,1) not null
FK	hod_decimal_par	int(11) unsigned not null
FK	hod_decimal_vozidlo	int(11) unsigned not null

Obrázek 18: Relace hod_decimal

```
CREATE TABLE `hod_decimal` (
  `hod_decimal_id` int(11) unsigned AUTO_INCREMENT not null,
  `hod_decimal` decimal(2,1) not null,
  `hod_decimal_par` int(11) unsigned not null,
  `hod_decimal_vozidlo` int(11) unsigned not null,
  PRIMARY KEY (`hod_decimal_id`),
  KEY `FK` (`hod_decimal_par`, `hod_decimal_vozidlo`)
);
```

Kód 8: Vytvoření relace hod_decimal

4.7 Relace hod_date

Slouží k ukládání hodnot data ve formátu „YYYY-MM-DD“, který se u vozidel dá využít například u parametru, který označuje konec platnosti STK, kde je důležitý konkrétní den. Relace je popsána atributem *hod_date_id*, který je primárním klíčem relace. Atribut *hod_year* slouží k ukládání hodnoty roku, atribut *hod_year_par* jako cizí klíč odkazuje na atribut *parametr_id*, relace parametr, čímž přiděluje hodnotu konkrétnímu parametru. Poslední atribut označený jako *hod_year_vozidlo* je cizím klíčem relace, který odkazuje na atribut *vozidlo_id*, relace vozidlo a přiděluje hodnotu a parametr vozidlu.

hod_date		
PK	hod_date_id	int(11) unsigned not null
	hod_date	date not null
FK	hod_date_par	int(11) unsigned not null
FK	hod_date_vozidlo	int(11) unsigned not null

Obrázek 19: Relace hod_date

```
CREATE TABLE `hod_date` (
  `hod_date_id` int(11) unsigned AUTO_INCREMENT not null,
  `hod_date` date not null,
  `hod_date_par` int(11) unsigned not null,
  `hod_date_vozidlo` int(11) unsigned not null,
  PRIMARY KEY (`hod_date_id`),
  KEY `FK` (`hod_date_par`, `hod_date_vozidlo`)
);
```

Kód 9: Vytvoření relace hod_date

4.8 Relace hod_year

Na rozdíl od relace hod_date slouží tato relace k ukládání hodnot roku ve formátu „YYYY“. Využitelná je tak pro ukládání konkrétních roků, bez nutnosti evidence konkrétního měsíce či dne. Příkladem může být hodnota atributu rok výroby automobilu, u kterého se eviduje pouze rok (např. 2005). Relace je popsána atributem *hod_year_id*, který plní funkci primárního klíče. Atributem *hod_year*, který slouží pro ukládání hodnoty roků. Cizí klíč označený jako *hod_year_par* odkazuje na relaci parametr, konkrétně atribut *parametr_id*, a přiřazuje hodnotu parametru. Atribut *hod_year_vozidlo* je taktéž cizí klíč. Odkazuje na atribut *vozdlo_id*, relace vozidlo a přiřazuje hodnotu a parametr vozidlu.

hod_year		
PK	hod_year_id	int(11) unsigned not null
	hod_year	year not null
FK	hod_year_par	int(11) unsigned not null
FK	hod_year_vozidlo	int(11) unsigned not null

Obrázek 20: Relace hod_year

```
CREATE TABLE `hod_year` (
  `hod_year_id` int(11) unsigned AUTO_INCREMENT not null,
  `hod_year` year not null,
  `hod_year_par` int(11) unsigned not null,
  `hod_year_vozidlo` int(11) unsigned not null,
  PRIMARY KEY (`hod_year_id`),
  KEY `FK` (`hod_year_par`, `hod_year_vozidlo`)
);
```

Kód 10: Vytvoření relace hod_year

4.9 Relace vozidlo

Relace vozidlo je hlavní relací automobilu. Je popsána primárním klíčem *vozidlo_id*, pomocí kterého je možné jednoznačně identifikovat každé vozidlo, s kterým se dostala společnost do styku. Atribut *vozidlo_model* je cizím klíčem atributu *model_id* relace model. Přiřazuje vozidlu informaci, o který model se jedná (v závislosti na modelu i značku). Atribut *vozidlo_puvod* je cizím klíčem atributu *puvod_id*, relace puvod, která je určena k evidenci seznamu států. Atribut původu nemusí být zadán. V případě zadání informuje o státu, z kterého automobil pochází. Poslední atribut označený jako *vozidlo_pobočka* je cizím klíčem atributu *pobočka_id* relace pobočka. Informuje o skutečnosti, na které pobočce se automobil nachází.

vozidlo		
PK	vozidlo_id	int(11) unsigned not null
FK	vozidlo_model	smallint(4) unsigned not null
FK	vozidlo_puvod	smallint(4) unsigned
FK	vozidlo_pobočka	tinyint(3) unsigned not null

Obrázek 21: Relace vozidlo

```
CREATE TABLE `vozidlo` (
  `vozidlo_id` int(11) unsigned AUTO_INCREMENT not null,
  `vozidlo_model` smallint(4) unsigned not null,
  `vozidlo_puvod` smallint(4) unsigned,
  `vozidlo_pobočka` tinyint(3) unsigned not null,
  PRIMARY KEY (`vozidlo_id`),
  KEY `FK` (`vozidlo_model`, `vozidlo_puvod`, `vozidlo_pobočka`)
);
```

Kód 11: Vytvoření relace vozidlo

4.10 Relace model

Relace model slouží pro evidenci modelů vozidel. Atribut *id_model* představuje primární klíč. Atribut *model_nazev*, datového typu *varchar*, je určen pro textový zápis konkrétních názvů modelu (Octavia, Passat atd.). Každý model patří ke značce (výrobci). Tento vztah je v relaci reprezentován atributem *model_znacka*. Jedná se o cizí klíč, který odkazuje na relaci *znacka*, atribut *znacka_id*. Atribut *model_kategorie* řeší rozlišení kategorií modelů (osobní, užitkové atd.). Jedná se o cizí klíč odkazující na relaci *kategorie*, konkrétně atribut *kategorie_id*.

model		
PK	model_id	smallint(4) unsigned not null
	model_nazev	varchar(50) not null
FK	model_znacka	smallint(4) unsigned not null
FK	model_kategorie	tinyint(3) unsigned not null

Obrázek 22: Relace model

```
CREATE TABLE `model` (
  `model_id` smallint(4) unsigned AUTO_INCREMENT not null,
  `model_nazev` varchar(50) not null,
  `model_znacka` smallint(4) unsigned not null,
  `model_kategorie` tinyint(3) unsigned not null,
  PRIMARY KEY (`model_id`),
  KEY `FK` (`model_znacka`, `model_kategorie`)
);
```

Kód 12: Vytvoření relace model

4.11 Relace znacka

Relace znacka je určena k evidenci značek vozidel. Obsahuje atribut *znacka_id*, který plní roli primárního klíče. Atribut *znacka_nazev*, datového typu varchar, umožňuje textový zápis konkrétního názvu značky (Škoda, Volkswagen atd.).

znacka		
PK	znacka_id	smallint(4) unsigned not null
	znacka_nazev	varchar(100) not null

Obrázek 23: Relace znacka

```
CREATE TABLE `znacka` (
  `znacka_id` smallint(4) unsigned AUTO_INCREMENT not null,
  `znacka_nazev` varchar(50) not null,
  PRIMARY KEY (`znacka_id`)
);
```

Kód 13: Vytvoření relace znacka

4.12 Relace kategorie

Relace je určena k uložení hodnot kategorie vozidel. Roli primárního klíče plní atribut *kategorie_id*. Atribut *kategorie_nazev*, datového typu *varchar* slouží pro uložení názvu kategorie (osobní, užitkový atd.).

kategorie		
PK	<i>kategorie_id</i>	<i>tinyint(3) unsigned not null</i>
	<i>kategorie_nazev</i>	<i>varchar(20) not null</i>

Obrázek 24: Relace kategorie

```
CREATE TABLE `kategorie` (
  `kategorie_id` tinyint(3) unsigned AUTO_INCREMENT not null,
  `kategorie_nazev` varchar(20) not null,
  PRIMARY KEY (`kategorie_id`)
);
```

Kód 14: Vytvoření relace kategorie

4.13 Relace pobočka

Relace pobočka je určena k ukládání informací o pobočkách. Role primárního klíče připadá na atribut *pobočka_id*. Každá pobočka disponuje svým názvem (pobočka Praha I, II atd.). Tento fakt reprezentuje atribut *pobočka_nazev*, datového typu *varchar*. Atribut *pobočka_adresni_misto* je cizím klíčem atributu *adresa_id*, relace *adresa*. Atribut odkazuje na informace o adrese pobočky. Atributy *pobočka_man_tel*, *pobočka_info_tel* a

pobočka_dil_tel, datového typu *varchar* slouží k ukládání telefonního čísla ve stejném pořadí pro manažera, infolinku a dílnu. Emailové adresy pro stejné destinace jsou uloženy v atributech *pobočka_man_email*, *pobočka_info_email* a *pobočka_dil_email*. Atributy emailových adres disponují datovým typem *tinytext*.

pobočka		
PK	<i>pobočka_id</i>	<i>tinyint(3) unsigned not null</i>
	<i>pobočka_nazev</i>	<i>varchar(50) not null</i>
FK	<i>pobočka_adresni_misto</i>	<i>int(11) unsigned not null</i>
	<i>pobočka_man_tel</i>	<i>varchar(10) not null</i>
	<i>pobočka_man_email</i>	<i>tinytext not null</i>
	<i>pobočka_info_tel</i>	<i>varchar(10) not null</i>
	<i>pobočka_info_email</i>	<i>tinytext not null</i>
	<i>pobočka_dil_tel</i>	<i>varchar(10) not null</i>
	<i>pobočka_dil_email</i>	<i>tinytext not null</i>

Obrázek 25: Relace pobočka

```
CREATE TABLE `pobočka` (
  `pobočka_id` tinyint(3) unsigned AUTO_INCREMENT not null,
  `pobočka_nazev` varchar(50) not null,
  `pobočka_adresni_misto` int(11) unsigned not null,
  `pobočka_man_tel` varchar(10) not null,
  `pobočka_man_email` tinytext not null,
  `pobočka_info_tel` varchar(10) not null,
  `pobočka_info_email` tinytext not null,
  `pobočka_dil_tel` varchar(10) not null,
  `pobočka_dil_email` tinytext not null,
  PRIMARY KEY (`pobočka_id`),
  KEY `FK` (`pobočka_adresni_misto`)
);
```

Kód 15: Vytvoření relace pobočka

4.14 Relace adresni_misto

Relace adresa je určena pro evidenci dostupných adres v České republice. Může sloužit současně pro relaci obsahující zákazníky, pobočky, a v případě potřeby i pro jiné entity, které disponují adresou. Pole *adresni_misto_id* slouží jako primární klíč relace. Atribut *adresni_misto_obec* slouží pro evidenci města, atribut *adresni_misto_cast_obec* nemusí být zadán, a slouží k uložení informace o části evidované obce. Součástí je i evidence ulice – atribut *adresni_misto_ulice*, který také nemusí být zadán a atribut čísla popisného – *adresni_misto_cp*. Poslední atribut označený jako *adresni_misto_psc* je určen k evidenci poštovního směrovacího čísla.

adresni_misto		
PK	adresni_misto_id	int(11) unsigned not null
	adresni_misto_obec	varchar(100) not null
	adresni_misto_cast_obec	varchar(100)
	adresni_misto_ulice	varchar(100)
	adresni_misto_cp	varchar(20) not null
	adresni_misto_psc	varchar(10) not null

Obrázek 26: Relace adresni_misto

```
CREATE TABLE `adresni_misto` (
  `adresni_misto_id` int(11) unsigned AUTO_INCREMENT not null,
  `adresni_misto_obec` varchar(100) not null,
  `adresni_misto_cast_obec` varchar(100),
  `adresni_misto_ulice` varchar(100),
  `adresni_misto_cp` varchar(20) not null,
  `adresni_misto_psc` varchar(10) not null,
  PRIMARY KEY (`adresni_misto_id`)
);
```

Kód 16: Vytvoření relace adresni_misto

4.15 Relace volna_pozice

Relace s názvem *volna_pozice* slouží pro evidenci volných pracovních míst, které společnost nabízí. Pomocí relace je možno následně data prezentovat ve webové prezentaci. Obsahuje atribut *volna_pozice_id*, který slouží jako primární klíč relace. Dále obsahuje atribut s názvem *volna_pozice_vlozenim*, datového typu *date*, který slouží pro uložení data, v kterém byla pozice do databáze uložena. Atribut *volna_pozice_nazev* je cizí klíč odkazující na primární klíč relace *pozice*, konkrétně atribut *pozice_id*, a slouží pro vložení názvu pozice (účetní, servisní technik). Atribut *volna_pozice_popis* disponuje datovým typem *text* a slouží pro uložení popisu pozice (popis se vztahuje ke konkrétní pozici, nikoli k názvu, proto je uložen v této relaci a nikoliv v relaci se seznamem názvů pozic). Poslední atribut relace má název *volna_pozice_pobočka*, je cizí klíč odkazující na relaci *pobočka*, atribut *pobočka_id*. Slouží pro informaci, na které pobočce je daná pozice nabízena.

volna_pozice		
PK	volna_pozice_id	int(11) unsigned not null
	volna_pozice_vlozeni	date not null
FK	volna_pozice_nazev	tinyint(3) unsigned not null
	volna_pozice_popis	text
FK	volna_pozice_pobočka	tinyint(3) unsigned not null

Obrázek 27: Relace volna_pozice

```
CREATE TABLE `volna_pozice` (
  `volna_pozice_id` int(11) unsigned AUTO_INCREMENT not null,
  `volna_pozice_vlozeni` date not null,
  `volna_pozice_nazev` tinyint(3) unsigned not null,
  `volna_pozice_popis` text,
  `volna_pozice_pobočka` tinyint(3) unsigned not null,
  PRIMARY KEY (`volna_pozice_id`),
  KEY `FK` (`volna_pozice_nazev`, `volna_pozice_pobočka`);
```

Kód 17: Vytvoření relace volna_pozice

4.16 Relace pozice

Relace pozice slouží pro ukládání názvů pracovních pozic. Umožňuje použít jeden název pozice na více pobočkách. Jako primární klíč relace slouží atribut *pozice_id*. Druhý atribut s názvem *pozice_nazev*, datového typu *varchar*, slouží k zápisu konkrétního názvu pozice (např. účetní, servisní technik atd.).

pozice		
PK	pozice_id	tinyint(3) unsigned not null
	pozice_nazev	varchar(50) not null

Obrázek 28: Relace pozice

```
CREATE TABLE `pozice` (  
  `pozice_id` tinyint(3) unsigned AUTO_INCREMENT not null,  
  `pozice_nazev` varchar(50) not null,  
  PRIMARY KEY (`pozice_id`)  
);
```

Kód 18: Vytvoření relace pozice

4.17 Relace uzivatel

Pro evidenci registrovaných uživatelů slouží relace *uzivatel*. Primární klíč relace je reprezentován atributem *uzivatel_id*. Atributy *uzivatel_username* a *uzivatel_password* reprezentují přihlašovací údaje uživatele. Pro uložení kontaktních údajů uživatele jsou určeny atributy *uzivatel_email* (emailová adresa) a *uzivatel_tel* (telefonní číslo). Atributy *uzivatel_jmeno* a *uzivatel_prijmeni* slouží pro uložení jména a příjmení uživatele.

uzivatel		
PK	uzivatel_id	int(11) unsigned not null
	uzivatel_username	tinytext not null
	uzivatel_password	longtext not null
	uzivatel_email	tinytext not null
	uzivatel_tel	varchar(10) not null
	uzivatel_jmeno	tinytext not null
	uzivatel_prijmeni	tinytext not null

Obrázek 29: Relace uzivatel

```

CREATE TABLE `uzivatel` (
  `uzivatel_id` int(11) unsigned AUTO_INCREMENT not null,
  `uzivatel_username` tinytext not null,
  `uzivatel_password` longtext not null,
  `uzivatel_email` tinytext not null,
  `uzivatel_tel` varchar(10) not null,
  `uzivatel_jmeno` tinytext not null,
  `uzivatel_prijmeni` tinytext not null,
  PRIMARY KEY (`uzivatel_id`)
);

```

Kód 19: Vytvoření relace uzivatel

4.18 Relace rezervace

Relace rezervace eviduje seznam rezervací, které byly provedeny uživateli. Roli primárního klíče plní atribut *rezervace_id*. Atribut *rezervace_datum* je určen k zaznamenání data, kdy byla rezervace provedena. Atribut *rezervace_vozidlo* je cizím klíčem atributu *vozidlo_id*, relace vozidlo a obsahuje informaci o tom, které vozidlo k dané rezervaci přiděleno. Poslední atribut s názvem *rezervace_uzivatel* je cizím klíčem atributu *uzivatel_id* relace uzivatel, přičemž informuje o uživateli, který danou rezervaci provedl.

rezervace		
PK	rezervace_id	int(11) unsigned not null
	rezervace_datum	date not null
FK	rezervace_vozidlo	int(11) unsigned not null
FK	rezervace_uzivatel	int(11) unsigned not null

Obrázek 30: Relace rezervace

```
CREATE TABLE `rezervace` (
  `rezervace_id` int(11) unsigned AUTO_INCREMENT not null,
  `rezervace_datum` date not null,
  `rezervace_vozidlo` int(11) unsigned not null,
  `rezervace_uzivatel` int(11) unsigned not null,
  PRIMARY KEY (`rezervace_id`),
  KEY `FK` (`rezervace_vozidlo`, `rezervace_uzivatel`)
);
```

Kód 20: Vytvoření relace rezervace

4.19 Relace oblíbené

Relace oblíbené slouží k evidenci oblíbených vozidel uživatelů. Primárním klíčem je atribut *oblíbené_id*. Atribut *oblíbené_vozidlo* je cizím klíčem atributu *vozidlo_id*, relace vozidlo a informuje o vozidle, které je přidáno do seznamu oblíbených. Atribut *oblíbené_uzivatel* je cizí klíč atributu *uzivatel_id*, relace uzivatel. Obsahuje informaci, ke kterému uživateli se daná oblíbená položka vztahuje.

oblíbené		
PK	oblíbené_id	int(11) unsigned not null
FK	oblíbené_vozidlo	int(11) unsigned not null
FK	oblíbené_uzivatel	int(11) unsigned not null

Obrázek 31: Relace oblíbené

```
CREATE TABLE `oblibene` (
  `oblibene_id` int(11) unsigned AUTO_INCREMENT not null,
  `oblibene_vozidlo` int(11) unsigned not null,
  `oblibene_uzivatel` int(11) unsigned not null,
  PRIMARY KEY (`oblibene_id`),
  KEY `FK` (`oblibene_vozidlo`, `oblibene_uzivatel`)
);
```

Kód 21: Vytvoření relace oblibene

4.20 Relace puvod

Relace puvod je určena k evidenci států pro potřeby původu vozidla. Primární klíč reprezentuje atribut *puvod_id*. Atribut *puvod_stat* je určen k uložení názvů států (Česká republika, Slovensko,..).

puvod		
PK	puvod_id	smallint(4) unsigned not null
	puvod_stat	varchar(80) not null

Obrázek 32: Relace puvod

```
CREATE TABLE `puvod` (
  `puvod_id` smallint(4) unsigned not null,
  `puvod_stat` varchar(80) not null,
  PRIMARY KEY (`puvod_id`)
);
```

Kód 22: Vytvoření relace puvod

4.21 Relace nabídka

Relace nabídka může obsahovat informace o nabídce dostupných vozidel. Atribut *nabidka_id* je primárním klíčem relace. Atribut *nabidka_vozidlo* je cizím klíčem atributu *vozidlo_id*, relace vozidlo. Atribut vozidla zařazuje konkrétní vozidlo do nabídky. Atribut

nabidka_datum_vlozeni eviduje datum, kdy byla daná nabídka zveřejněna. Naopak atribut *nabidka_datum_odstraneni* označuje datum, kdy bylo vozidlo z nabídky staženo (lze tak zpětně dohledat, v jakém časovém úseku bylo vozidlo v nabídce). Fakt, zdali je nabídka aktivní či nikoli je uložen v atributu *nabidka_aktivni* (0-neaktivní, 1-aktivní). V případě, že je vozidlo prodáno, a je staženo z nabídky, stane se nabídka neaktivní.

nabidka		
PK	nabidka_id	int(11) unsigned not null
FK	nabidka_vozidlo	int(11) unsigned not null
	nabidka_datum_vlozeni	date not null
	nabidka_datum_odstraneni	date
	nabidka_aktivni	tinyint(1) not null

Obrázek 33: Relace nabidka

```
CREATE TABLE `nabidka` (
  `nabidka_id` int(11) unsigned AUTO_INCREMENT not null,
  `nabidka_vozidlo` int(11) unsigned not null,
  `nabidka_datum_vlozeni` date not null,
  `nabidka_datum_odstraneni` date,
  `nabidka_aktivni` tinyint(1) not null,
  PRIMARY KEY (`nabidka_id`),
  KEY `FK` (`nabidka_vozidlo`)
);
```

Kód 23: Vytvoření relace nabidka

4.22 Relace smlouva

Relace smlouva je určena k evidenci smluv společnosti, které se vztahují k výkupu nebo prodeji automobilů. Primární klíč relace je atribut *smlouva_id*. V atributu *smlouva_datum* je uložen datum uzavření smlouvy. Typ služby, který je předmětem smlouvy je uveden v atributu *smlouva_sluzba*, který je cizím klíčem atributu *sluzba_id* relace sluzba. Atribut *smlouva_platba* informuje o způsobu platby smlouvy, a je cizím klíčem atributu

platba_id relace *platba*. Atribut *smlouva_vozidlo* je cizí klíč atributu *vozidlo_id* relace *vozidlo*, a odkazuje na vozidlo, které je předmětem smlouvy. Atribut *smlouva_zakaznik* je cizí klíč atributu *zakaznik_id*, a nese informaci o druhé straně, která se účastní smlouvy. Atribut *smlouva_castka* je určena k uložení částky, která je uvedena ve smlouvě.

smlouva		
PK	<i>smlouva_id</i>	int(11) unsigned not null
	<i>smlouva_datum</i>	date not null
FK	<i>smlouva_sluzba</i>	tinyint(1) unsigned not null
FK	<i>smlouva_platba</i>	tinyint(1) unsigned not null
FK	<i>smlouva_vozidlo</i>	int(11) unsigned not null
FK	<i>smlouva_zakaznik</i>	int(11) unsigned not null
	<i>smlouva_castka</i>	int(11) unsigned not null

Obrázek 34: Relace prodano

```
CREATE TABLE `smlouva` (
  `smlouva_id` int(11) unsigned AUTO_INCREMENT not null,
  `smlouva_datum` date not null,
  `smlouva_sluzba` tinyint(1) unsigned not null,
  `smlouva_platba` tinyint(1) unsigned not null,
  `smlouva_vozidlo` int(11) unsigned not null,
  `smlouva_zakaznik` int(11) unsigned not null,
  `smlouva_castka` int(11) unsigned not null,
  PRIMARY KEY (`smlouva_id`),
  KEY `FK` (`smlouva_sluzba`, `smlouva_platba`, `smlouva_vozidlo`,
`smlouva_zakaznik`)
);
```

Kód 24: Vytvoření relace smlouva

4.23 Relace sluzba

Relace sluzba je určena k evidenci poskytovaných služeb pro potřeby relace smlouva. Atribut *sluzba_id* je primární klíč relace. Pro uložení textového názvu typu služby je určen atribut *sluzba_typ*.

sluzba		
PK	sluzba_id	tinyint(1) unsigned not null
	sluzba_typ	varchar(50)

Obrázek 35: Relace sluzba

```
CREATE TABLE `sluzba` (
  `sluzba_id` tinyint(1) unsigned AUTO_INCREMENT not null,
  `sluzba_typ` varchar(50),
  PRIMARY KEY (`sluzba_id`)
);
```

Kód 25: Vytvoření relace sluzba

4.24 Relace platba

Relace platba je určena k evidenci dostupných způsobů platby vozidla zákazníkem. Atribut *platba_id* je primární klíč relace a atribut *platba_zpusob* obsahuje textově zapsaný způsob platby.

platba		
PK	platba_id	tinyint(1) unsigned not null
	platba_zpusob	varchar(30) not null

Obrázek 36: Relace platba

```
CREATE TABLE `platba` (
  `platba_id` tinyint(1) unsigned AUTO_INCREMENT not null,
  `platba_zpusob` varchar(20) not null,
  PRIMARY KEY (`platba_id`)
);
```

Kód 26: Vytvoření relace platba

4.25 Relace zakaznik

Relace zakaznik umožňuje evidovat zákazníky, kteří si od prodejce v minulosti zakoupili automobil. Roli primárního klíče disponuje atribut *zakaznik_id*. O zákazníkovi jsou uloženy údaje: rodné číslo – atribut *zakaznik_rc*, jméno – atribut *zakaznik_jmeno*, příjmení – atribut *zakaznik_prijmeni* a adresa, která je v relaci reprezentována formou cizího klíče – atribut *zakaznik_adresni_misto*. Atribut adresy odkazuje na atribut *adresni_misto_id*, relace *adresni_misto*.

zakaznik		
PK	zakaznik_id	int(11) unsigned not null
	zakaznik_rc	varchar(20) not null
	zakaznik_jmeno	tinytext not null
	zakaznik_prijmeni	tinytext not null
FK	zakaznik_adresni_misto	int(11) unsigned not null

Obrázek 37: Relace zakaznik


```
CREATE TABLE `zakaznik` (  
  `zakaznik_id` int(11) unsigned AUTO_INCREMENT not null,  
  `zakaznik_rc` varchar(20) not null,  
  `zakaznik_jmeno` tinytext not null,  
  `zakaznik_prijmeni` tinytext not null,  
  `zakaznik_adresni_misto` int(11) unsigned not null,  
  PRIMARY KEY (`zakaznik_id`),  
  KEY `FK` (`zakaznik_adresni_misto`)  
);
```

Kód 27: Vytvoření relace zakaznik

4.26 Relace sluzba_prodat

Relace *sluzba_prodat* je určena pro ukládání dat z online formuláře, který je na webových stránkách k dispozici pro odeslání informací o vozidle a kontaktu v případě, že má návštěvník webové aplikace zájem o prodej auta u společnosti. Atribut *prodat_id* představuje primární klíč relace. Atribut *prodat_model* je cizím klíčem atributu *model_id*, relace *model* a slouží k uložení modelu vozidla, který chce zákazník prodat. Další údaje o vozidle se ukládají do atributu *prodat_rok_vyroby*, který ukládá rok výroby vozidla a atribut *prodat_cen_predstava*, který je určen pro uložení cenové představy vlastníka automobilu. Pro uložení kontaktních údajů zákazníka z formuláře jsou určeny atributy *prodat_tel_cislo*, což je atribut určený pro uložení telefonního čísla zákazníka (musí být zadán vždy) a atribut *prodat_email*, určený k uložení emailové adresy (není povinný). Pro hodnotu vybrané pobočky je určen atribut *prodat_pobocka*, což je cizí klíč atributu *pobocka_id*, relace *pobocka*.

sluzba_prodat		
PK	prodat_id	int(11) unsigned not null
FK	prodat_model	smallint(4) unsigned not null
	prodat_rok_vyroby	year not null
	prodat_cen_predstava	int(11) unsigned not null
	prodat_tel_cislo	varchar(10) not null
	prodat_email	tinytext
FK	prodat_pobocka	tinyint(3) unsigned not null

Obrázek 38: Relace sluzba_prodat

```
CREATE TABLE `sluzba_prodat` (  
  `prodat_id` int(11) unsigned AUTO_INCREMENT not null,  
  `prodat_model` smallint(4) unsigned not null,  
  `prodat_rok_vyroby` year not null,  
  `prodat_cen_predstava` int(11) unsigned not null,  
  `prodat_tel_cislo` varchar(10) not null,  
  `prodat_email` tinytext,  
  `prodat_pobocka` tinyint(3) unsigned not null,  
  PRIMARY KEY (`prodat_id`),  
  KEY `FK` (`prodat_model`, `prodat_pobocka`)  
);
```

Kód 28: Vytvoření relace sluzba_prodat

5 DESIGN APLIKACE

Obsah jednotlivých částí aplikace je situován do sekcí. Nadpisy sekcí se nachází nad sekcemi mimo obsahovou, barevně zvýrazněnou část sekce. Další rozdělení obsahu v rámci sekce je realizováno pomocí vodorovných čar. Většina tlačítek kromě textu obsahuje i logotyp relevantní k jeho funkcionalitě.

Design využívá pět barev. Tmavě šedá barva je využita jako pozadí většiny sekcí, ohraničení a jako barva písma. Oranžová barva je primárně použita pro všeobecná tlačítka. Zelená barva je v některých částech aplikace využita v kombinaci s tmavě šedou jako pozadí sekce. Mimo pozadí je využita jako pozadí některých tlačítek. Červená barva je využita u záporných tlačítek (tlačítko zrušit, odstranit, zpět, apod.). Pozadí celých stránek, jakožto i podklad sekcí disponuje světle šedou barvou. Text, který se nachází v sekcích, využívá pro kontrastní zobrazení na tmavě šedém pozadí bílou barvu.

Tabulka 10: Použité barvy

Barva	Hexadecimální formát	RGB formát
Tmavě šedá	#343a40	rgb(52,58,64)
Oranžová	#D05E34	rgb(208,94,52)
Zelená	#28a745	rgb(40,167,69)
Červená	#dc3545	rgb(220,53,69)
Světle šedá	#f8f9fa	rgb(248,249,250)
Bílá	#ffffff	rgb(255,255,255)

Pro potřeby stránky byl vytvořen imaginární logotyp, které se typově podobá logotypu společnosti, která se zabývá prodejem automobilů. Využívá font Hemi Head. Barva logotypu je tmavě šedá, stejná která je využita i v designu webových stránek.



Mauto

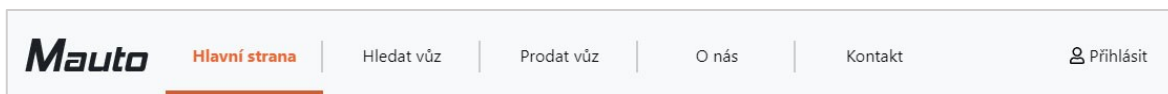
Obrázek 39: Imaginární logotyp

6 ČÁSTI APLIKACE

Aplikace je složena z několika částí. Jedná se o 13 různých stránek, přičemž některé vykazují čistě informativní charakter, a některé naopak disponují proměnným obsahem. Grafický a textový popis jednotlivých částí je obsahem této kapitoly.

6.1 Záhloví aplikace

Záhloví stránky tvoří lišta, která obsahuje logo prodejce, navigaci v podobě odkazů na jednotlivé stránky. V pravém rohu se nachází odkaz na přihlašovací stránku do uživatelského prostředí s logotypem uživatele. Záhloví se při scrollování stránky vždy fixuje k hornímu okraji zobrazovací plochy prohlížeče tak, že je dostupné z jakéhokoliv místa na stránce. Logo prodejce slouží jako odkaz na úvodní stranu. Jednotlivé odkazy navigace jsou odděleny vertikální čarou. Při použití polohovacího zařízení, se při najetí na odkaz text podtrhne tenkou čarou a barva textu se změní. Při výběru některého z odkazů se jeho barva textu změní a text podtrhne tlustou čarou.



Obrázek 40: Záhloví aplikace

6.2 Zápatí aplikace

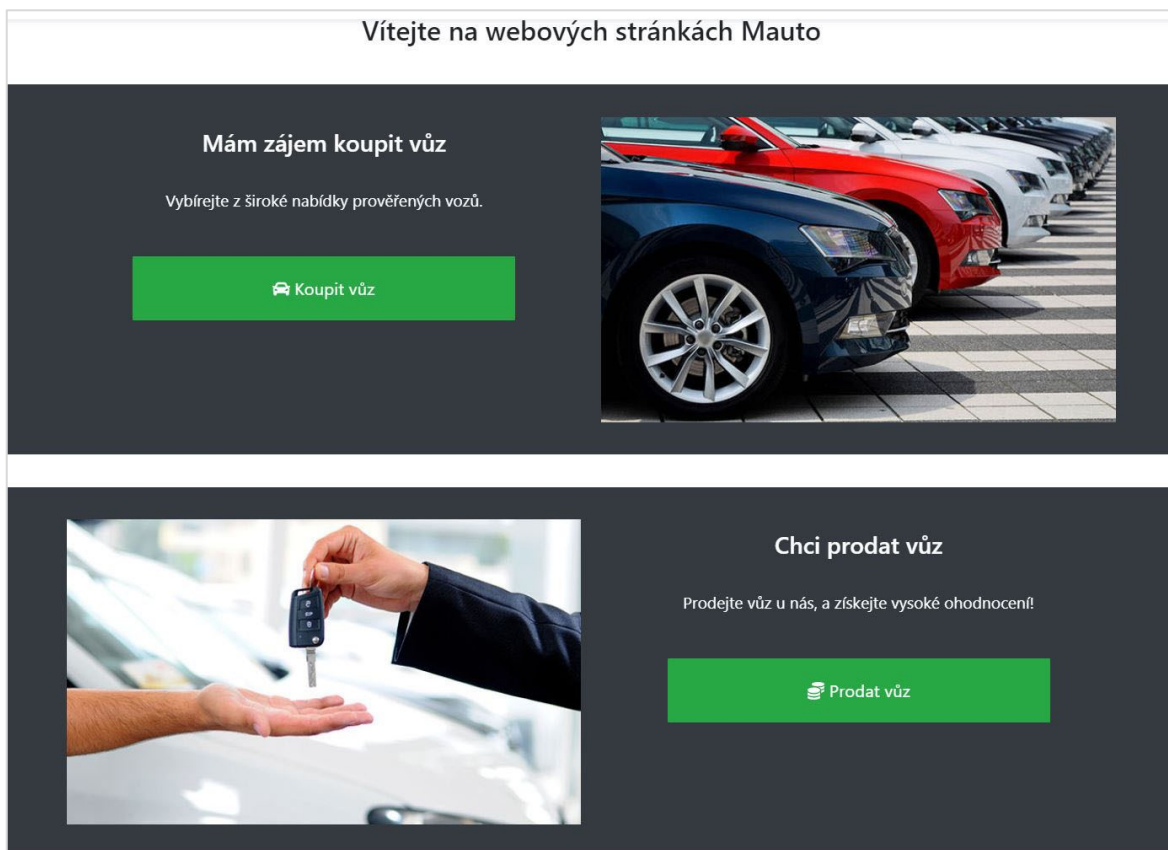
Zápatí je ve všech částech aplikace totožné. Součástí je navigace, která obsahuje méně důležité odkazy, mezi které patří odkaz na stránku Ochrana osobních údajů, Tipy při výběru vozidla, Dokumenty ke stažení a Kariéra u MAUTO. Pod navigací je zápatí rozděleno vodorovnou čarou, pod kterou se nachází do středu vycentrované odkazy na sociální sítě Facebook, Instagram a LinkedIn v podobě logotypů. Ve spodní části zápatí se uprostřed nachází copyright.



Obrázek 41: Zápatí aplikace

6.3 Úvodní strana

Úvodní strana obsahuje v horní části uvítací nápis „Vítejte na webových stránkách Mauto“. Pod nadpisem se nachází dvě sekce popisující hlavní funkcionality. První sekce se věnuje koupi vozu ze strany zákazníka a druhá prodeji automobilu zákazníka společnosti. Obě sekce obsahují nadpis, krátký popis funkcionality, tlačítko odkazující na část aplikace, která se funkcionalitě věnuje, a fotografii.

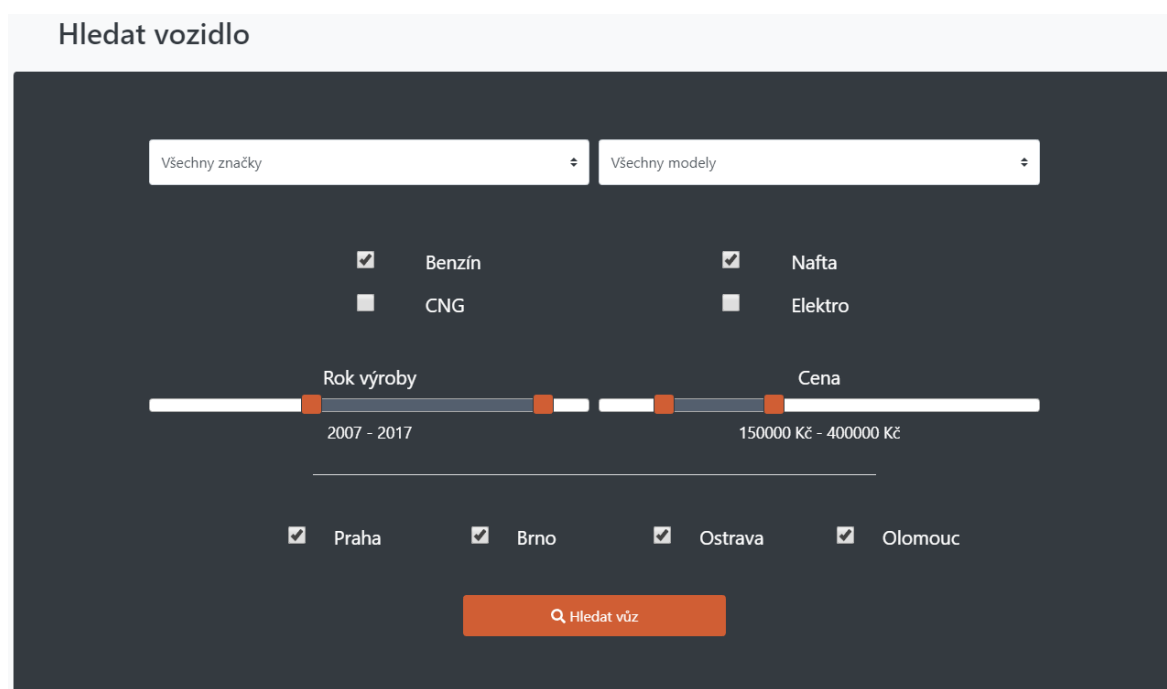


Obrázek 42: Úvodní strana

6.4 Hledat vůz

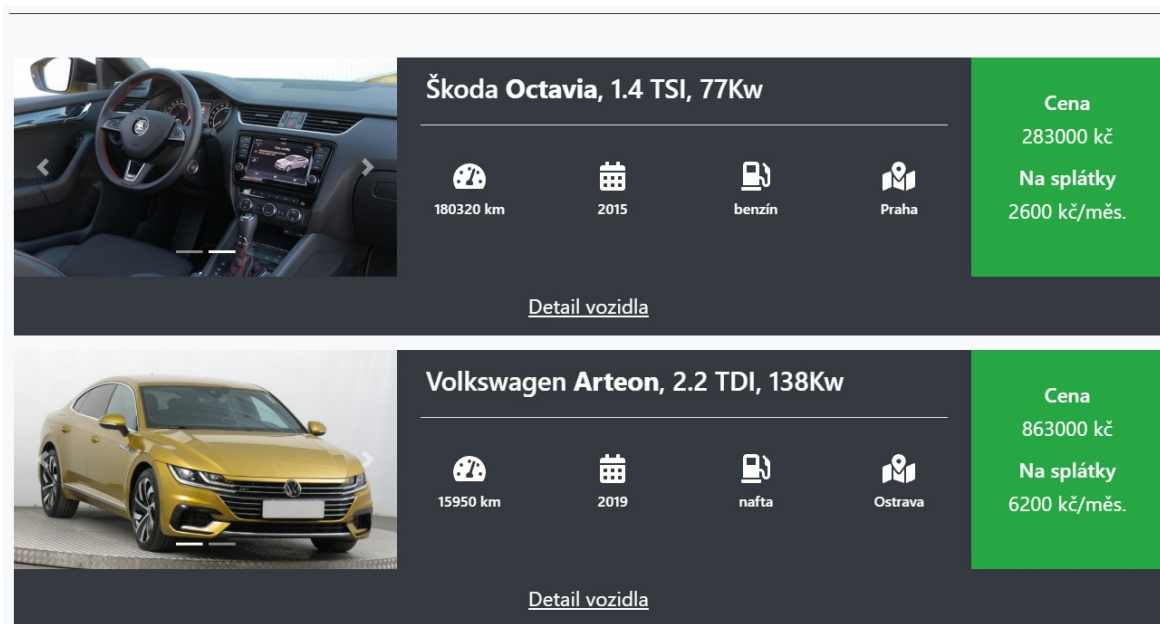
Strana Hledat vůz obsahuje formulář určený k filtrování výpisu vozidel. Výpis vozidel lze filtrovat dle značky, modelu, paliva, roku výroby, najetých kilometrů a pobočky. Výběr značky a modelu je formou výběrového pole. Vybírat je možno pouze z definovaných značek a jejich modelů. Filtr neumožňuje vyhledávat několik značek zároveň. Parametr Palivo je vyobrazen pomocí zaškrťovacích polí a nabízí výběr z hodnot Benzín, Nafta, CNG a jiné. V tomto případě filtr umožňuje kombinovat více parametrů v jednom vyhledávání. Parametr Rok výroby a Počet najetých kilometrů umožňuje uživateli definovat rozsah

hodnot pomocí posuvníků. Filtr umožňuje vyhledávat automobily, které byly vyrobeny mezi určitými roky nebo automobily v určitém cenovém rozsahu. Označení parametru je umístěno v obou případech nad posuvníkem a výběr hodnot pod posuvníkem. V poslední řadě je možno filtrovat pomocí poboček. Filtr umožňuje vypsát vozidla, které se nachází na některé pobočce. Stejně jako u parametru Palivo je vzhledem k řešení zaškrtačovacími poli možná kombinace více poboček (i všech) v jednom vyhledávání. Pod výběrem poboček se nachází uprostřed vycentrované tlačítko s textem „Hledat vůz“, pomocí které se provádí filtrování výsledků.



Obrázek 43: Formulář pro vyhledávání vozidla

Vypsaná vozidla jsou zobrazena formou karet. Každému vozidlu je přidělena jedna karta. Každá karta obsahuje v levé části interaktivní galerii dostupných fotografií, v které je umožněn pohyb mezi fotografiemi pomocí šipek na stranách. Název karty vozidla je složen ze značky, tučně zvýrazněného modelu, obsahu, označení motoru a výkonu vozidla. Pod nadpisem se nachází oddělovací čára. Základní, a pro zákazníka nejdůležitější parametry jsou umístěny pod oddělovací čárou ve formě logotypu a hodnoty parametru. Popsanými parametry jsou najeté kilometry, rok výroby, palivo a pobočka, na které se vůz nachází. V pravé, barevně oddělené části karty se nachází cena vozidla v hotovosti a na splátky. Ve spodní části karty je uprostřed vycentrován odkaz s textem „Detail vozidla“, který odkazuje na detailní stranu daného vozidla. Výpis vozidel je v horní a spodní části ohraničen tenkou čarou.

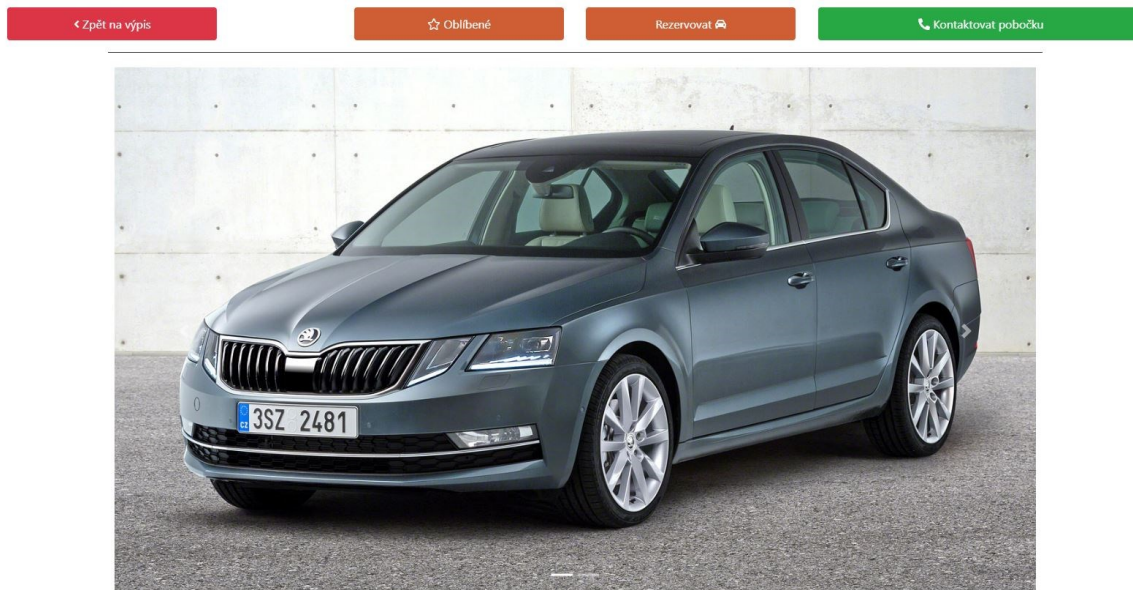


Obrázek 44: Výpis vozidel

6.5 Detail vozidla

Strana Detail vozidla obsahuje více informací o vozidle. Na straně se nachází interaktivní galerie s dostupnými fotografiemi automobilu, sekce se základními informacemi o vozidle včetně ceny, sekce s technickými parametry a sekce s informacemi o výbavě. Součástí strany jsou i tlačítka umístěna nad interaktivní galerií.

Galerie s fotografiemi umožňuje pohyb mezi fotografiemi pomocí šipek, které jsou umístěny na levé a pravé straně. Ve spodní části galerie je vyobrazena aktuální pozice v kontextu všech fotografií. Pod galerií se nachází dvě vertikálně oddělené sekce. Levá sekce obsahuje značku a tučně zvýrazněný model automobilu. Dalším obsahem jsou ostatní základní informace totožné s informacemi ve výpise vozidel (rok výroby, počet najetých kilometrů, palivo a pobočka). Ve spodní části sekce se nachází cena při platbě v hotovosti a cena na splátky v korunách českých. Pravá sekce obsahuje nadpis „Technické parametry“. Obsahem sekce jsou informace o technických parametrech vozidla jako je motorizace, počet míst k sezení, barva, platnost STK, typ karoserie a převodovky a VIN kód. Pod těmito sekcemi se nachází další sekce, rozměrově stejně velká jako dvě předešlé zároveň. Předmětem sekce jsou informace o výbavě, mezi které patří například klimatizace, ABS, navigace, tempomat, pohon všech kol, servisní kniha, ESP, parkovací asistent, centrální, elektronická okna.



Obrázek 45: Detail vozidla (1/2)

Volkswagen Arteon		Technické parametry	
2019	15950 km	Rok výroby	2019
Nafta	Ostrava	VIN kód	GV3B221B
Cena: 863000 Kč		Tachometr	15950 km
Splátka: od 6200 Kč/měs.		Motorizace	2.2 TDI, 138 kW
		Míst k sezení	5
		Převodovka	Automatická, 8 st.
		Barva	zlatá, matná
		Karoserie	Sedan
		STK	11/2020
		Palivo	Nafta
Výbava			
ABS, ESP, el. okna, klimatizace, xenony, vyhřívané sedačky, navigace, tempomat, parkovací asistent			

Obrázek 46: Detail vozidla (2/2)

6.6 Prodat vůz

Strana Prodat vůz obsahuje sekci, která obsahuje formulář určený k prodeji vozidla. Vrchní část sekce obsahuje nadpis „Vyplňte formulář a získejte bezkonkurenční cenovou nabídku“. Obsahem sekce je fotografie automobilu v levé části a seznam obsahující postup prodeje při využití níže umístěného formuláře v části právě. Každá položka seznamu obsahuje logotyp fajfky jako odrážku. Ve spodní části sekce se nachází formulář obsahující výběrové formulářové prvky pro značku a model vozidla, které si zákazník přeje prodat a výběr pobočky, která je zákazníkovi nejdostupnější, popřípadě které chce nabídku na výkup zaslat. Obsahem prodejního formuláře jsou dále prvky pro rok výroby, stav tachometru a

cenovou představu. Pro zanechání kontaktních údajů slouží pole určené pro telefonní číslo a email. Uprostřed sekce, pod vstupními prvky nalézá potvrzující tlačítko formuláře s textem „Odeslat na pobočku“, pomocí kterého se zadané údaje odesílají ke zpracování.

Obrázek 47: Sekce prodat vůz

6.7 O nás

Stránka O nás obsahuje pět sekcí, představující společnost a její reference. Každá sekce se věnuje určitému tématu. První sekce s názvem „O nás“ obsahuje stručný odstavec, představující prodejce. Druhá sekce s názvem „Proč zvolit nás?“ obsahuje popis výhod zákazníka u prodejce. Každá výhoda obsahuje příslušný logotyp a název výhody.

O nás

Společnost Mauto působí na českém trhu osobních a užitkových vozidel od roku 2018. Za své krátké působení si vytvořila rozsáhlou klientelu především díky svým kvalitním službám, širokému sortimentu vozů. Ceny Mauto patří dlouhodobě k nejnižším v České republice.

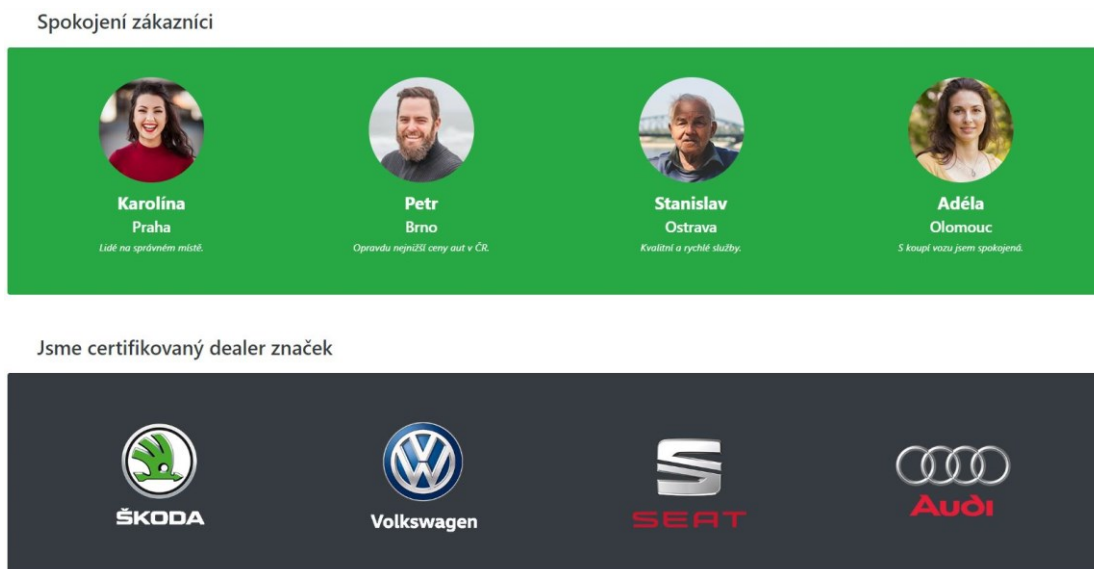
Proč zvolit nás?

- Kvalitní služby
- Široký výběr
- Péče o zákazníka
- Rychlé vyřízení
- Nízké ceny

Obrázek 48: Sekce O nás a Proč zvolit nás?

Třetí sekce „Spokojení zákazníci“ představuje spokojené zákazníky, kteří si auto u společnosti v minulosti zakoupili. Z každého města, v kterém se nachází pobočka prodejce

je vybrán jeden zákazník. Vypsáno je křestní jméno zákazníka, město a vzkaz zákazníka. Každý zákazník disponuje kulatou fotografií. Čtvrtá sekce s názvem „Jsme certifikovaný dealer značek“ obsahuje logotypy značek vozidel. Logotypy jsou umístěny vodorovně vedle sebe.



Obrázek 49: Sekce Spokojení zákazníci a Jsme certifikovaný dealer značek

Poslední sekce obsahuje vodorovný seznam poboček. Nad textem, který označuje město je umístěn logotyp mapy. Logotyp a město tvoří odkaz na stránku Kontakt s kotvou na sekci pobočky.



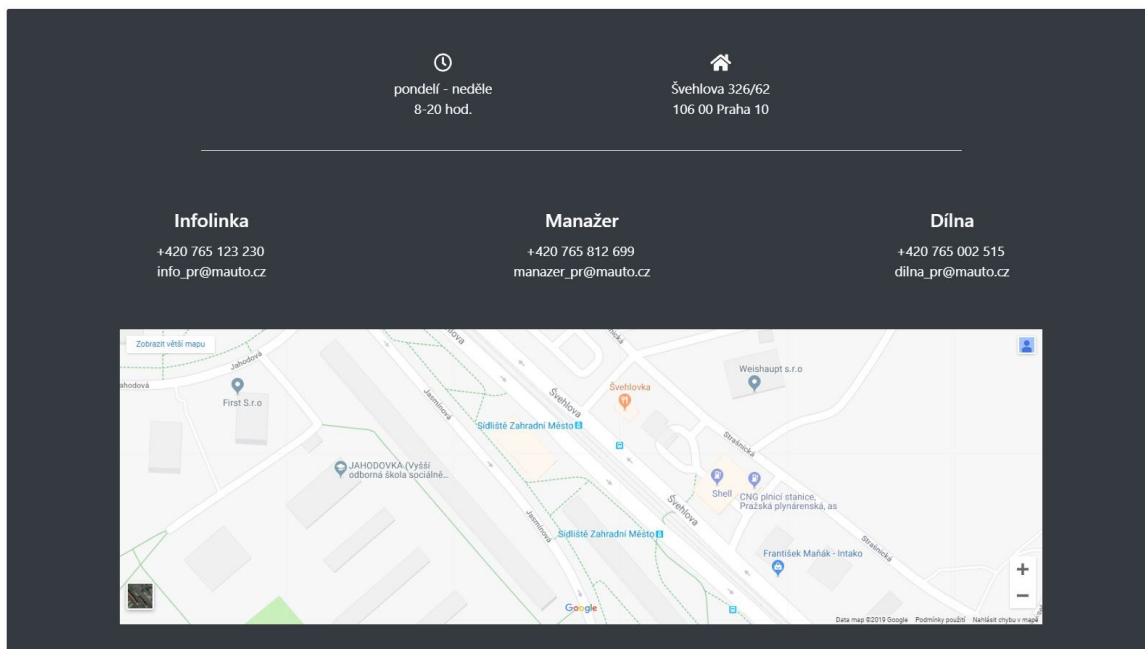
Obrázek 50: Sekce Pobočky

6.8 Kontakt

Strana Kontakt obsahuje několik sekcí, přičemž každá z nich obsahuje informace o jedné z poboček prodejce. Obsah sekce je rozčleněn na dva řádky. V prvním řádku se ve dvou sloupcích nachází otevírací doba a adresa pobočky s logotypem. V druhém řádku jsou informace rozčleněny do tří sloupců, kde v každém je umístěn telefonní kontakt a emailová adresa postupně dle sloupců na infolinku, manažera a dílnu. Pod kontakty se nachází

interaktivní objekt mapy (mapové aplikace Mapy Google), který poukazuje na adresu dané pobočky na mapě.

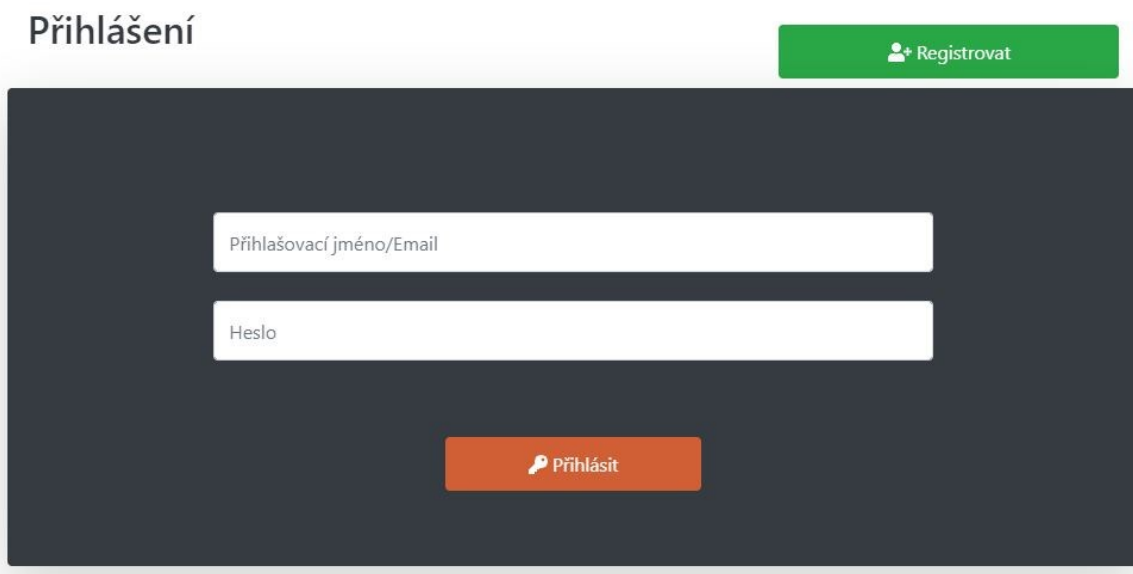
Pobočka Praha



Obrázek 51: Sekce pobočky

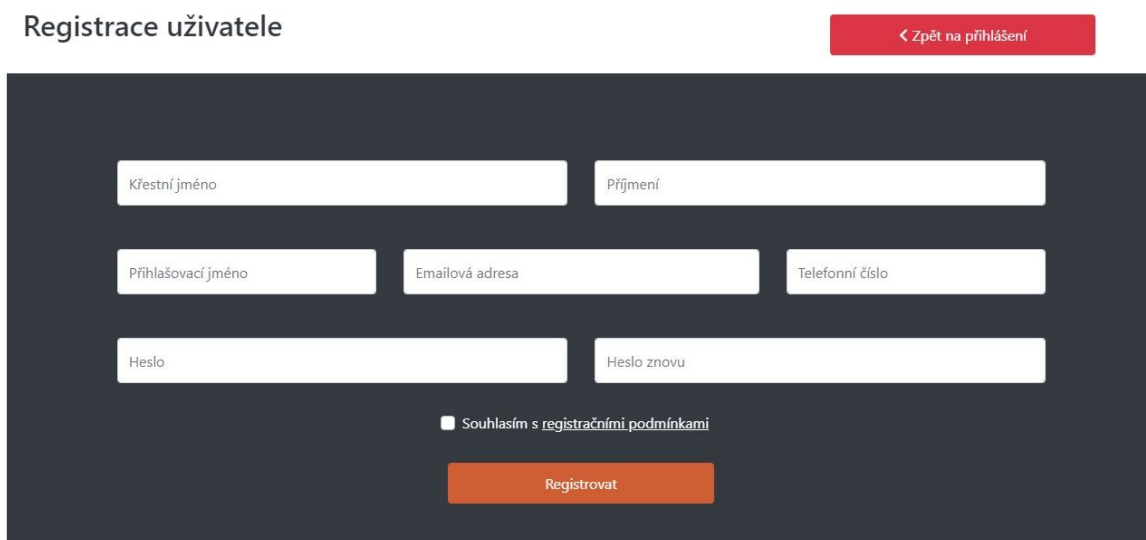
6.9 Přihlášení a Registrace uživatele

Pomocí strany Přihlásit, na kterou se odkaz v navigační liště nachází v pravém rohu, je registrovaným uživatelům umožněno přihlásit se do uživatelského účtu. Strana obsahuje sekci, v které je umístěn formulář, který obsahuje formulářové prvky, které jsou určeny pro zadání přihlašovacího jména a hesla. Popis vstupního pole je řešen pomocí tzv. placeholderů, které se nachází uvnitř pole. Odeslat přihlašovací údaje je možno pomocí tlačítka s logotypem klíče a textem „Přihlásit“, které je umístěno pod vstupními poli. Nad sekci s přihlašovacím formulářem se nachází tlačítko s textem „Registrovat“ a logotypem osoby se znakem plus. Tlačítko odkazuje na stranu, která obsahuje formulář pro registraci uživatele.



Obrázek 52: Přihlášení uživatele

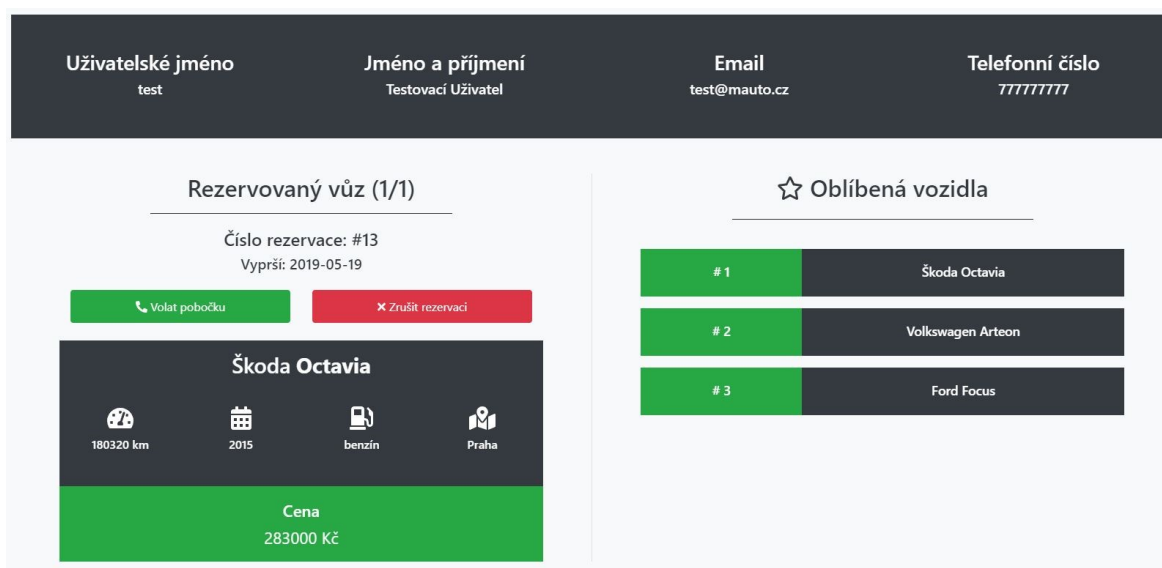
Strana registrace obsahuje registrační formulář, který obsahuje pole pro zadání křestního jména, příjmení, přihlašovacího jména, emailové adresy, telefonního čísla, hesla a jeho zopakování. Označení vstupních polí je stejně jako u formuláře pro přihlášení řešeno pomocí tzv. placeholderů. Pod zmíněnými prvky se nachází zaškrťovací pole s textem „Souhlasím s registračními podmínkami“, které potvrzuje přečtení podmínek pro registraci uživatele. Pod zaškrťovacím polem pro potvrzení registračních podmínek se nachází odesílací tlačítko formuláře s textem „Registrovat“. V případě přístupu ze strany Přihlášení je vyobrazeno tlačítko s textem „Zpět na přihlášení“, které se nachází nad sekci obsahující registrační formulář. Tlačítko odkazuje zpět na přihlašovací formulář.



Obrázek 53: Registrace uživatele

6.10 Uživatelské prostředí

Strana uživatelského prostředí je k dispozici pouze přihlášeným uživatelům. Poskytuje uživateli přehled vztahující se k jeho uživatelskému účtu. Obsahuje výpis údajů o uživateli, vypisuje rezervovaný automobil, seznam oblíbených automobilů a možnost editace uživatelských údajů. Informace o uživateli se nachází v horní části strany v podobě sekce. Zobrazují se vodorovně v pořadí uživatelské jméno, jméno a příjmení, email a telefonní číslo. Levá část obsahové části strany disponuje nadpisem „Rezervovaný vůz“. V případě že má uživatel přidělenou rezervaci vozidla vypisuje se automobil pod nadpisem. Výpis automobilu je formou karty, která obsahuje nadpis složený ze značky a modelu. Pod nadpisem karty jsou vyobrazeny základní parametry o vozidle (počet najetých kilometrů, rok výroby, palivo a pobočka). Ve spodní části karty je barevně oddělená část obsahující cenu automobilu v hotovosti. Nad kartou vozidla jsou umístěny dvě tlačítka. První tlačítko, obsahující text „Volat pobočku“ odkazuje na stranu Kontakt. Druhé tlačítko s textem „Zrušit rezervaci“ umožňuje rezervaci zrušit. V opačném případě, když uživatel nemá přidělenou rezervaci, vypíše se pod nadpis text „Není rezervován žádný vůz.“. Pravá část s nadpisem „Oblíbená vozidla“ obsahuje seznam vozidel v podobě malých karet. Jedná se o seznam automobilů, které uživatel označil jako oblíbené v detailu vozidla. Každá karta obsahuje název složený ze značky a modelu. V levé části karty se nachází barevně odlišená část obsahující identifikační číslo automobilu. V případě, že uživatel neoznačil žádné vozidlo jako oblíbené, zůstává obsah pod nadpisem prázdný.



Obrázek 54: Uživatelské prostředí

Pod obsahovou částí je umístěna sekce obsahující nadpis „Změna údajů“. V sekci se nachází formulář, který umožňuje změnit emailovou adresu a telefonní číslo a formulář určený pro změnu hesla. Potvrzovací tlačítko se nachází pod vstupními poli.

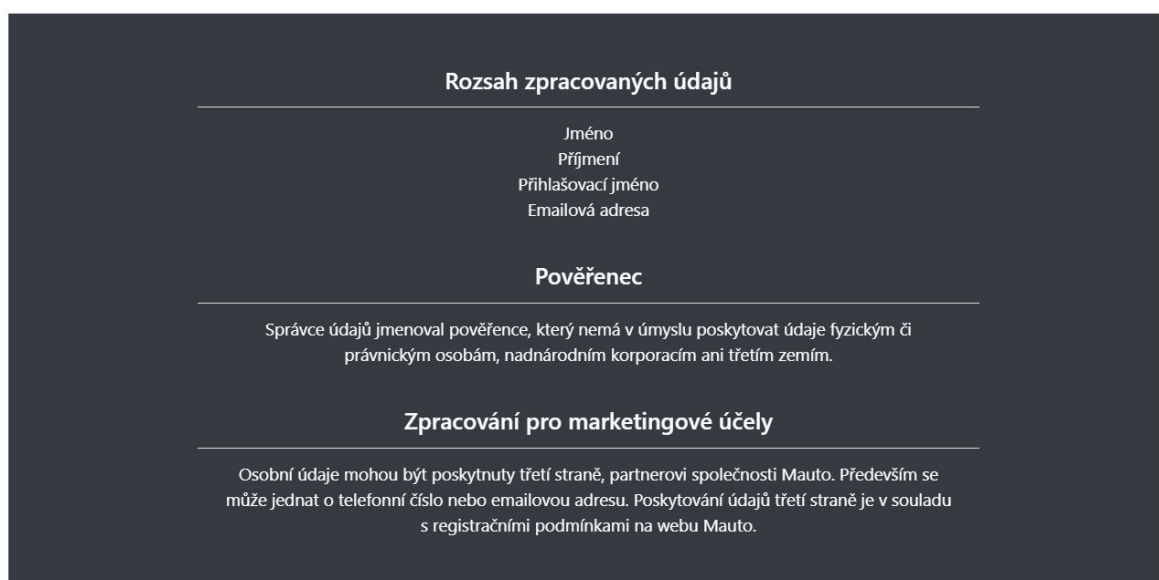


Obrázek 55: Změna údajů

6.11 Ostatní strany

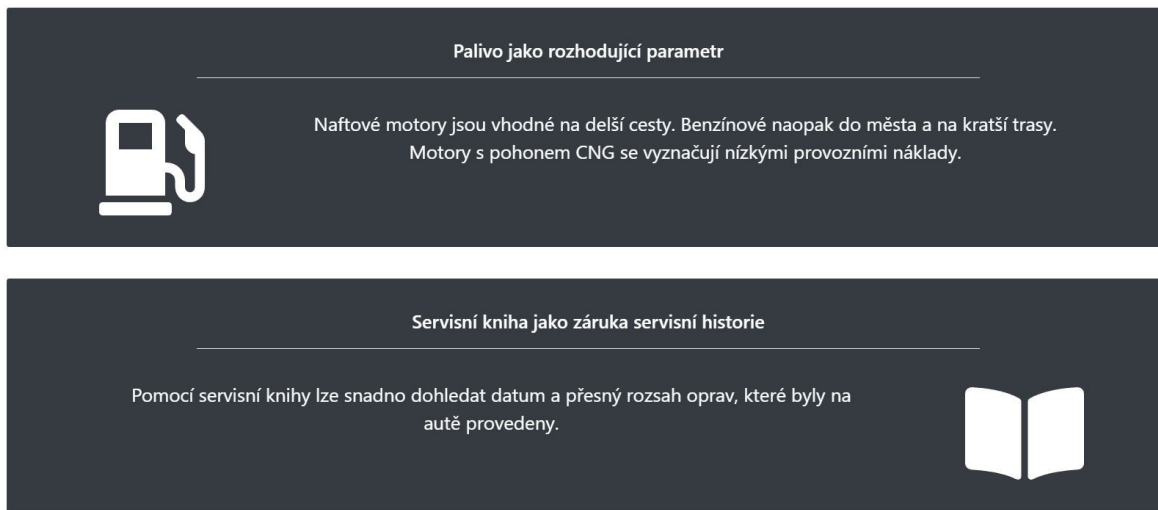
Odkazy na ostatní strany se nachází v zápatí. Jedná se o méně důležité strany, než na které je uveden odkaz v hlavní navigaci, která je umístěna v záhlaví. Odkazy jsou seřazeny následovně: Ochrana osobních údajů, Tipy při výběru vozidla, Dokumenty ke stažení a Kariéra u Mauto. Strana Ochrana osobních údajů obsahuje informace o ochraně a zpracování osobních údajů. Sekce obsahuje rozsah zpracovaných údajů, informace o pověření, a informace o zpracování údajů pro marketingové účely.

Ochrana a zpracování osobních údajů



Obrázek 56: Ochrana osobních údajů

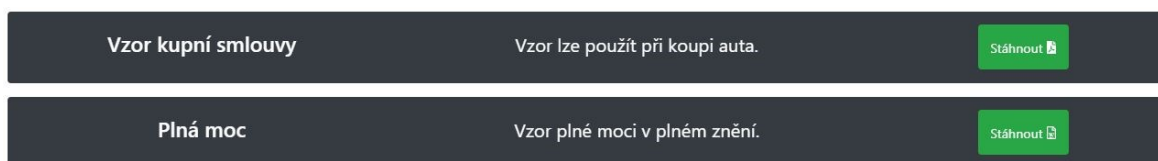
Strana Tipy při koupi vozu poskytuje zákazníkům praktické tipy k výběru vozidla. Jedna sekce obsahuje vždy jeden tip. V každé sekci je umístěn vystihující nadpis, vhodný logotyp a krátký text k danému tématu.



Obrázek 57: Tipy při koupi vozidla

Strana Dokumenty ke stažení obsahuje dokumenty úzce spojené s koupí či prodejem vozidla, které si může uživatel stáhnout. Dokumenty jsou rozděleny na malé sekce. Každá sekce obsahuje název dokumentu, popis a tlačítko Stáhnout s logotypem formátu dokumentu.

Dokumenty ke stažení



Obrázek 58: Dokumenty ke stažení

Strana Kariéra obsahuje sekci Volná místa, která obsahuje řádkový formulář, pomocí kterého lze filtrovat výpis volných pracovních pozic. Filtrovat lze dle textového pole názvu pozice a výběru jedné z poboček. Pod sekci obsahující filtrování je umístěn výpis volných pozic formou menších sekcí. Každá sekce obsahuje název pozice a pobočky, stručný popis a tlačítko Mám zájem, které odkazuje na modální okno obsahující kontakt na personální oddělení.

Volná místa

<input type="text" value="Název pozice"/>	<input type="text" value="Všechny pobočky"/>	<input type="button" value="Vyhledat"/>
---	--	---

Manažer Praha	Správa pobočky, řízení týmu.	<input type="button" value="Mám zájem"/>
Servisní technik Brno	Údržba a servisování vozidel.	<input type="button" value="Mám zájem"/>
Účetní Brno	Účtování, komunikace s finančním úřadem.	<input type="button" value="Mám zájem"/>

Obrázek 59: Volná místa

7 ZABEZPEČENÍ APLIKACE

Před vznikem anomálií v uložených datech jsou zabezpečeny vstupy formulářů, které ověřují, zdali jsou zadaná data odpovídající těm, která mají být uložena. Například vstup určený pro emailovou adresu je ověřen pomocí funkce `filter_var()` a vstupy telefonních čísel se ověřují, zdali se jedná o devítimístnou číselnou hodnotu.

Aplikace využívá ve většině případů komunikace s databází knihovnu Dibi, která v případě, že dotazy nejsou poskládány jako řetězec zamezuje možnost napadení typu SQL injection.

```
require 'pripojeni.php';
$date = date(format: 'Y-m-d');
$car_id = $_GET['car_id'];
$user_id = $_GET['user_id'];

$db->query( args: "INSERT INTO rezervace", [
    'rezervace_datum' => $date,
    'rezervace_vozidlo' => $car_id,
    'rezervace_uzivatel' => $user_id
]);

header( string: "Location: ../user-gui.php");
exit();
```

Obrázek 60: Uložení hodnot pomocí knihovny Dibi

Uložení hodnot registrace není provedeno knihovnou Dibi, ale pomocí parametrizovaného dotazu, který taktéž zamezuje možnost vzniku SQL injection pokud není dotaz sestaven jako řetězec, ale hodnoty jsou nahrazeny zástupnými znaky.

```
$sql = "INSERT INTO uzivatel (uzivatel_username, uzivatel_password, uzivatel_email,
uzivatel_tel, uzivatel_jmeno, uzivatel_prijmeni) VALUES (?, ?, ?, ?, ?, ?)";
$stmt = mysqli_stmt_init($conn);

if(!mysqli_stmt_prepare($stmt,$sql)){
    header( string: "Location: ../registration.php?err=sqlerror");
    exit();
}
else{
    $hashedpwd = password_hash($password, algo: PASSWORD_DEFAULT);

    mysqli_stmt_bind_param($stmt, types: "ssssss", &var1: $username,
    &...: $hashedpwd, $email, $tel, $firstname, $lastname );
    mysqli_stmt_execute($stmt);

    header( string: "Location: ../registration.php?registrace=success");
    exit();
}
```

Obrázek 61: Uložení hodnot registrace

Heslo je v databázi uloženo jako otisk, který provedla v jazyce PHP funkce `password_hash()`. Při přihlašování je takto uložené heslo ověřováno pomocí funkce `password_verify()`.

```
if($row = mysqli_fetch_assoc($res)){
    $pwdkontrola = password_verify($password, $row['uzivatel_password']);
    if($pwdkontrola == false){
        header( string: "Location: ../login.php?err=spatneheslo");
        exit();
    }
    elseif ($pwdkontrola == true){
        session_start();
        $_SESSION['id_user'] = $row['uzivatel_id'];
        $_SESSION['username'] = $row['uzivatel_username'];
        $_SESSION['email'] = $row['uzivatel_email'];
        $_SESSION['jmeno'] = $row['uzivatel_jmeno'];
        $_SESSION['prijmeni'] = $row['uzivatel_prijmeni'];
        $_SESSION['tel'] = $row['uzivatel_tel'];

        header( string: "Location: ../user-gui.php?login=success");
        exit();
    }
}
```

Obrázek 62: Ověření hesla při přihlášení

ZÁVĚR

V rámci bakalářské práce byl vytvořen návrh databázové struktury a prezentační šablona pro prodejce automobilů. Prezentační šablona disponuje prvky připomínající jednoduchou webovou aplikaci s funkcemi pro výpis vozidel, možností registrace či rezervace vozidla. Návrh databázové struktury zohledňuje evidenci dat webové aplikace i dalších relevantních dat pro daný obor.

Teoretická část popisuje technologie, které byly v průběhu realizace použity. Zjištění požadavků naplňuje kapitola Analýza existujících řešení. Požadavky jsou dále definovány a rozebrány v kapitole Rozbor a definice požadavků. Požadavky byly při realizaci postupně zakomponovány a zohledňovány při realizaci projektu.

Při realizaci webové aplikace byl využit framework Bootstrap. Pomocí jeho mřížkového systému je aplikace navržena jako plně responzivní. Prvotní návrh databázové struktury byl realizován ve webové aplikaci Lucidchart. Návrh je určen pro databázový systém MySQL. Dynamické prvky aplikace jsou realizovány jazykem PHP, který umožňuje interakce s databází pomocí jazyka SQL.

Výsledek realizace je obsahem praktické části práce, která podrobně popisuje účel části databáze, včetně kódů pro vytváření jednotlivých tabulek. Webová aplikace je pro popis rozdělena dle jednotlivých stránek na části. Ty obsahují snímek obrazovky dané části a popisný text. V závěru praktické části je stručně popsáno zabezpečení aplikace.

Webová aplikace může být dále rozšířena o systém, určený pro personál na straně prodejce automobilů, který by umožňoval správu automobilů, rezervací a další administrativní funkce. Tento systém by společně s realizovaným projektem mohl plnit funkční celek použitelný v praxi.

Výsledkem práce je webová prezentace s prvky jednoduché aplikace určená pro prodejce automobilů a návrh databázové struktury umožňující evidenci dat zmíněné aplikace, popřípadě dalších relevantních dat pro daný obor.

SEZNAM POUŽITÉ LITERATURY

- [1] History of the Web. WORLD WIDE WEB FOUNDATION [online]. Ženeva: World Wide Web Foundation, c2008-2019 [cit. 2019-03-01]. Dostupné z: <https://webfoundation.org/about/vision/history-of-the-web/>
- [2] CASTRO, Elizabeth a Bruce HYSLOP. HTML5 a CSS3: názorný průvodce tvorbou WWW stránek. Přeložil Ondřej BAŠE, přeložil Kristýna BAŠE. Brno: Computer Press, 2012. ISBN 978-80-251-3733-8.
- [3] ISO/IEC 15445:2000: Information technology -- Document description and processing languages -- HyperText Markup Language (HTML). International Organization for Standardization [online]. Ženeva, Švýcarsko: International Organization for Standardization [cit. 2019-05-15]. Dostupné z: <https://www.iso.org/standard/27688.html>
- [4] Historie a vývoj HTML. HTML5 [online]. Jiří Kosek, ©1997-2014 [cit. 2019-03-13]. Dostupné z: <http://htmlguru.cz/uvod-historie.html>
- [5] Verze HTML. Jak psát web [online]. Praha: Janovský [cit. 2019-03-23]. Dostupné z: <https://www.jakpsatweb.cz/html/verze-html.html>
- [6] HTML5 Introduction. W3schools [online]. Cambridge, Spojené státy americké: World Wide Web Consortium, c1999-2019 [cit. 2019-03-10]. Dostupné z: https://www.w3schools.com/html/html5_intro.asp
- [7] Webdesignérův průvodce po HTML5 – nová sémantika. Zdrojak.cz [online]. Praha: Devel.cz Lab, 2010 [cit. 2019-03-01]. Dostupné z: <https://www.zdrojak.cz/clanky/webdesigneruv-pruvodce-po-html5-nova-semantika/>
- [8] HTML5 New Elements. W3schools.com [online]. Cambridge: World Wide Web Consortium, c1999-2019 [cit. 2019-03-01]. Dostupné z: https://www.w3schools.com/html/html5_new_elements.asp
- [9] History of HTML and CSS. Introduction to HTML5 and CSS3 [online]. Liverpool, UK: Hissom, c2011 [cit. 2019-05-15]. Dostupné z: <http://amyhissom.com/HTML5-CSS3/history.html>
- [10] CSS prefixy. JE ČAS [online]. Praha: Jahoda, 2019 [cit. 2019-04-25]. Dostupné z: <http://jecas.cz/css-prefixy>
- [11] MICHÁLEK, Martin. Vzhůru do (responzivního) webdesignu. Verze 1.1. Praha: vlastním nákladem autora, 2017. ISBN 978-80-88253-00-6.

- [12] Anatomie Media Query. In: Vzhůru dolů [online]. Praha: Martin Michálek, c2019 [cit. 2019-04-24]. Dostupné z: <https://www.vzhurudolu.cz/prirucka-content/dist/images/medium/media-query.jpg>
- [13] Lekce 1 - Úvod do JavaScriptu. ITnetwork.cz [online]. Praha: Čápka, c2019 [cit. 2019-04-28]. Dostupné z: <https://www.itnetwork.cz/javascript/zaklady/javascript-tutorial-uvod-do-javascriptu-nepochopeny-jazyk>
- [14] A Brief History of JavaScript. Auth0 [online]. Washington, USA: Auth0, c2013-2019 [cit. 2019-04-05]. Dostupné z: <https://auth0.com/blog/a-brief-history-of-javascript/>
- [15] Lekce 1 - Úvod do CSS frameworku Bootstrap. Itnetwork.cz [online]. Praha: Čápka, c2019 [cit. 2019-03-20]. Dostupné z: <https://www.itnetwork.cz/html-css/bootstrap/kurz/uvod-do-css-frameworku-bootstrap>
- [16] Lekce 5 - Bootstrap - Úvod do grid systémů. Itnetwork.cz [online]. Praha: Čápka, c2019 [cit. 2019-03-20]. Dostupné z: <https://www.itnetwork.cz/html-css/bootstrap/kurz/bootstrap-uvod-do-grid-systemu/>
- [17] Bootstrap Grid System. In: C# Corner [online]. Philadelphia, USA: C# Corner, c2019 [cit. 2019-05-01]. Dostupné z: <https://csharpcorner-mindcrackerinc.netdna-ssl.com/article/bootstrap-grid-system/Images/1.png>
- [18] 9 of the Best Animation Libraries for UI Designers. Sitepoint [online]. Melbourne: SitePoint Pty., c2000-2019 [cit. 2019-03-20]. Dostupné z: <https://www.sitepoint.com/our-top-9-animation-libraries/>
- [19] Font-Awesome. GitHub [online]. San Francisco, USA: GitHub, c2019 [cit. 2019-05-05]. Dostupné z: <https://github.com/FortAwesome/Font-Awesome/>
- [20] Font Awesome 5 Update. In: PaperStreet [online]. Miami: PaperStreet Web Design, c2019 [cit. 2019-05-02]. Dostupné z: <https://www.paperstreet.com/blog/wp-content/uploads/2018/04/Screen-Shot-2018-04-17-at-5.25.30-PM-1024x562.png>
- [21] Lekce 1 - Úvod do jQuery. Itnetwork.cz [online]. Praha: Čápka, c2019 [cit. 2019-03-20]. Dostupné z: <https://www.itnetwork.cz/javascript/jquery-zaklady/javascript-tutorial-funkcionalni-programovani-a-jquery-webova-kalkulacka>
- [22] První díl o PHP – Historie a budoucnost. PHGame.cz [online]. Praha: Kozina, c2010-2019 [cit. 2019-05-06]. Dostupné z: https://phgame.cz/PHGame_serialy/serialy/php/prvni-dil-o-php-historie-a-budoucnost

- [23] Novinky jazyka PHP 7. ROOT.CZ [online]. Praha: Internet Info, 2019 [cit. 2019-05-10]. Dostupné z: <https://www.root.cz/clanky/novinky-jazyka-php-7/>
- [24] KROENKE, David a David J. AUER. Databáze. Brno: Computer Press, 2015, 496 s. ISBN 978-80-251-4352-0.
- [25] Normalizace relačních databází. Programujte.com [online]. Brno: Devtea.cz, c2003-2019, 23. 7. 2008 [cit. 2019-03-01]. Dostupné z: <http://programujte.com/clanek/2008071900-normalizace-relacnich-databazi/>
- [26] Jazyk SQL. Voho [online]. Lucemburk: Hordějčuk, c2008-2019 [cit. 2019-03-01]. Dostupné z: <http://voho.eu/wiki/sql/>
- [27] MySQL – čeština a slovenština. Interval.cz [online]. Brno: ZONER software, a.s [cit. 2019-05-10]. Dostupné z: <https://www.interval.cz/clanky/mysql-cestina-a-slovenstina/>
- [28] Lekce 6 - MySQL krok za krokem: Datové typy a NULL. ITnetwork.cz [online]. Praha: Čápka, c2019 [cit. 2019-03-01]. Dostupné z: <https://www.itnetwork.cz/mysql/mysql-tutorial-datove-typy-a-null>
- [29] Integer Types (Exact Value) - INTEGER, INT, SMALLINT, TINYINT, MEDIUMINT, BIGINT. MySQL [online]. Redwood City, Kalifornie, USA: Oracle Corporation, c2019 [cit. 2019-04-24]. Dostupné z: <https://mysql.com/doc/refman/8.0/en/integer-types.html>
- [30] Obrana proti útoku SQL injection v PHP. ITnetwork.cz [online]. Praha: Čápka, c2019 [cit. 2019-05-13]. Dostupné z: <https://www.itnetwork.cz/php/bezpecnost/tutorial-bezpecnost-v-php-utok-sql-injection-a-obrana>
- [31] Jak správně ukládat hesla. PAVEL KUTÁČ [online]. Ostrava: Kutáč, c2014-2019 [cit. 2019-05-13]. Dostupné z: <https://www.kutac.cz/blog/weby-a-vse-okolo/jak-spravne-ukladat-hesla/>
- [32] Understanding bcrypt salt as used by PHP password_hash. In: STACK OVERFLOW [online]. New York, USA: Stack Exchange, c2019 [cit. 2019-05-13]. Dostupné z: <https://i.stack.imgur.com/m8opZ.png>
- [33] Mototechna. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-03-01]. Dostupné z: <https://cs.wikipedia.org/wiki/Mototechna>
- [34] Mototechna [online]. Praha: Mototechna group, c2019 [cit. 2019-03-01]. Dostupné z: <https://www.mototechna.cz/>

- [35] AAA Auto. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-03-01]. Dostupné z: https://cs.wikipedia.org/wiki/AAA_Auto
- [36] AAA auto [online]. Praha: AAA Auto International, c2019 [cit. 2019-03-01]. Dostupné z: <https://www.aaaauto.cz/>
- [37] AutoESA. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2019-03-01]. Dostupné z: <https://cs.wikipedia.org/wiki/AutoESA>
- [38] AutoEsa [online]. Praha: AutoESA, 2019 [cit. 2019-03-01]. Dostupné z: <https://www.autoesa.cz>
- [39] DEVICES. Gemius Ranking [online]. Praha: Gemius, c2019 [cit. 2019-04-25]. Dostupné z: <http://ranking.gemius.com/cz/ranking/platforms/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

WWW	World Wide Web
HTTP	Hypertext Transfer Protocol
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
PHP	Hypertext Preprocessor
SQL	Structured Query Language

SEZNAM OBRÁZKŮ

Obrázek 1: Syntaxe CSS3 media queries [12].....	19
Obrázek 2: Bootstrap grid systém [17].....	21
Obrázek 3: Ikony knihovny Font-awesome [20].....	22
Obrázek 4: Databázový systém.....	23
Obrázek 5: Otisk hesla vytvořen algoritmem BCrypt [32].....	33
Obrázek 6: Webová prezentace Mototechna Group a.s. [34].....	35
Obrázek 7: Webová prezentace AAA Auto International a.s. [36].....	37
Obrázek 8: Filtr vyhledávání vozidel AAA Auto [36].....	38
Obrázek 9: Webová prezentace AutoEsa a.s. [38].....	39
Obrázek 10: Detail vozidla AutoEsa [38].....	40
Obrázek 11: Podíl zařízení na českém internetu [39].....	43
Obrázek 12: ER diagram.....	47
Obrázek 13: Relace parametr.....	48
Obrázek 14: Relace hod_varchar.....	49
Obrázek 15: Relace hod_tinyint.....	50
Obrázek 16: Relace hod_smallint.....	51
Obrázek 17: Relace hod_mediumint.....	52
Obrázek 18: Relace hod_decimal.....	53
Obrázek 19: Relace hod_date.....	54
Obrázek 20: Relace hod_year.....	55
Obrázek 21: Relace vozidlo.....	56
Obrázek 22: Relace model.....	57
Obrázek 23: Relace značka.....	57
Obrázek 24: Relace kategorie.....	58
Obrázek 25: Relace pobočka.....	59
Obrázek 26: Relace adresni_misto.....	60
Obrázek 27: Relace volna_pozice.....	61
Obrázek 28: Relace pozice.....	62
Obrázek 29: Relace uzivatel.....	63
Obrázek 30: Relace rezervace.....	64
Obrázek 31: Relace oblíbene.....	64
Obrázek 32: Relace puvod.....	65

Obrázek 33: Relace nabídka	66
Obrázek 34: Relace prodano	67
Obrázek 35: Relace sluzba	68
Obrázek 36: Relace platba	68
Obrázek 37: Relace zakaznik	69
Obrázek 38: Relace sluzba_prodat	71
Obrázek 39: Imaginární logotyp	72
Obrázek 40: Záhlaví aplikace	73
Obrázek 41: Zápatí aplikace	73
Obrázek 42: Úvodní strana	74
Obrázek 43: Formulář pro vyhledávání vozidla	75
Obrázek 44: Výpis vozidel	76
Obrázek 45: Detail vozidla (1/2)	77
Obrázek 46: Detail vozidla (2/2)	77
Obrázek 47: Sekce prodat vůz	78
Obrázek 48: Sekce O nás a Proč zvolit nás?	78
Obrázek 49: Sekce Spokojení zákazníci a Jsme certifikovaný dealer značek	79
Obrázek 50: Sekce Pobočky	79
Obrázek 51: Sekce pobočky	80
Obrázek 52: Přihlášení uživatele	81
Obrázek 53: Registrace uživatele	81
Obrázek 54: Uživatelské prostředí	82
Obrázek 55: Změna údajů	83
Obrázek 56: Ochrana osobních údajů	83
Obrázek 57: Tipy při koupi vozidla	84
Obrázek 58: Dokumenty ke stažení	84
Obrázek 59: Volná místa	85
Obrázek 60: Uložení hodnot pomocí knihovny Dibi	86
Obrázek 61: Uložení hodnot registrace	87
Obrázek 62: Ověření hesla při přihlášení	87

SEZNAM TABULEK

Tabulka 1: Obecné atributy jazyka HTML	12
Tabulka 2: Subverze HTML 5 [6]	13
Tabulka 3: Sémantické elementy [2][7].....	15
Tabulka 4: Praktické CSS3 vlastnosti [2]	18
Tabulka 5: Základní podmínky CSS3 media queries	19
Tabulka 6: Ukázka relace	29
Tabulka 7: Číselné datové typy [29].....	31
Tabulka 8: Textové datové typy [28].....	32
Tabulka 9: Ostatní datové typy [28]	32
Tabulka 10: Použité barvy	72

SEZNAM KÓDŮ

Kód 1: Syntaxe HTML5	14
Kód 2: Syntaxe kaskádových stylů	17
Kód 3: Vytvoření relace parametr	48
Kód 4: Vytvoření relace hod_varchar	49
Kód 5: Vytvoření relace hod_tinyint	50
Kód 6: Vytvoření relace hod_smallint	51
Kód 7: Vytvoření relace hod_mediumint	52
Kód 8: Vytvoření relace hod_decimal	53
Kód 9: Vytvoření relace hod_date	54
Kód 10: Vytvoření relace hod_year	55
Kód 11: Vytvoření relace vozidlo	56
Kód 12: Vytvoření relace model	57
Kód 13: Vytvoření relace znacka	58
Kód 14: Vytvoření relace kategorie	58
Kód 15: Vytvoření relace pobočka	59
Kód 16: Vytvoření relace adresni_misto	60
Kód 17: Vytvoření relace volna_pozice	61
Kód 18: Vytvoření relace pozice	62
Kód 19: Vytvoření relace uzivatel	63
Kód 20: Vytvoření relace rezervace	64
Kód 21: Vytvoření relace oblíbene	65
Kód 22: Vytvoření relace puvod	65
Kód 23: Vytvoření relace nabídka	66
Kód 24: Vytvoření relace smlouva	67
Kód 25: Vytvoření relace sluzba	68
Kód 26: Vytvoření relace platba	69
Kód 27: Vytvoření relace zakaznik	70
Kód 28: Vytvoření relace sluzba_prodat	71

SEZNAM PŘÍLOH

P I: OBSAH PŘILOŽENÉHO CD

PŘÍLOHA P I: OBSAH PŘILOŽENÉHO CD

Přiložené CD obsahuje:

- fulltext.pdf – bakalářská práce
- physical-model.pdf – fyzický model databáze
- code.zip – zdrojové kódy