



## **Středoškolská technika 2022**

**Setkání a prezentace prací středoškolských studentů na ČVUT**

### **Správa vozového parku**

**Petr Boháč**

Střední průmyslová škola elektrotechnická Ječná  
Ječná 517, Nové Město

### **Anotace**

Tato práce se zabývá vytvořením webové aplikace, jejímž cílem je uživatelsky příjemný nástroj pro přidávání automobilů do vozového parku, vylepšený například o generování QR kódů s technickými specifikacemi automobilů. QR kódy budou sloužit zákazníkům pro čtení informací o jednotlivých automobilech. Práce mimo jiné také zahrnuje správu uživatelů a predikci ceny automobilů tvořené pomocí umělé inteligence. Ta byla testována na datech získaných z největšího českého autobazaru. V rámci tvorby aplikace byli mimo jiné osloveni specialisté v oboru prodeje automobilů. Získané informace a jejich zkušenosti s autobazary sloužily pro rozšíření funkcionalit aplikace.

# Obsah

1. Úvod.....	3
2. Technologie použité při vývoji .....	4
2.1. PHP – Laravel .....	4
2.2. Scrapy .....	4
2.3. Vývoj machine-learningového modelu .....	4
2.4. Flask .....	5
2.5. Apache.....	6
2.6. Virtual Private Server – Linux .....	7
2.7. Databáze – MariaDB .....	7
3. Realizace vývoje projektu.....	8
4. Požadavky pro spuštění.....	9
5. Základní funkce aplikace .....	9
5.1. Přihlášení do systému.....	9
5.2. Správa uživatelů .....	10
5.3. Feedback.....	11
5.4. Evidence aut .....	11
5.5. Tvorba faktur.....	12
5.6. Přínos aplikace pro uživatele a firmy .....	13
6. Závěr .....	14
7. Použitá literatura .....	15
8. Přílohy.....	16

# 1. Úvod

Hlavním úkolem aplikace je umožnit jednoduché a přehledné přidávání automobilů do databáze vozového parku. Aplikace nabízí i generování QR kódů s technickými specifikacemi automobilů. QR kódy jsou primárně určeny pro zákazníky k přehlednému získání informací o vybraném autě. Kromě přidávání automobilů do databáze vozového parku, aplikace také umožňuje jednoduchou a přehlednou správu uživatelů. Efektivní spravování a sledování aktivit všech uživatelů v rámci vozového parku je určeno zejména pro majitele vozového parku, který bude mít přehled o všech aktivitách svých zaměstnanců.

Projekt zároveň ukazuje, jak nám může umělá inteligence pomáhat. V mém projektu se jedná o predikci ceny automobilů. Predikce vychází z analýzy přibližně dvaceti tisíc vozů. Čím je databáze rozsáhlejší, tím přesnější a relevantnější výsledky dostaneme. Je to velký pokrok a určitě o hodně lepší než pouhé odhadnutí ceny. Umělou inteligenci s predikcí ceny automobilu jsem testoval na datech největšího českého autobazaru.

V průběhu tvorby aplikace jsem oslovil specialisty v oboru prodeje automobilů, kteří mi poskytli cenné informace jak s prací v autobazarech, tak při nákupu aut. Jejich zkušenosti a informace mi pomohly s vývojem aplikace tak, aby co nejlépe odpovídala reálným potřebám a požadavkům.

V rámci dalšího vývoje aplikace jsem přidal CRM model, který umožňuje sledovat a analyzovat interakce se zákazníky. Tento model umožňuje snadno zaznamenávat a spravovat kontakty se zákazníky, jejich požadavky a preference, a to vše na jednom místě. Díky tomu bude prodejní tým schopen poskytovat lepší a personalizované služby pro zákazníky, což může vést ke zvýšení prodejů. CRM model také umožňuje sledovat úspěšnost marketingových kampaní a zjišťovat preference zákazníků. Tyto informace mohou být velmi cenné pro rozhodování o tom, které produkty a služby nabízet a jak je nabízet. S implementací CRM modelu bude aplikace ještě více přizpůsobena potřebám prodejního týmu a pomůže mu lépe porozumět potřebám a preference zákazníků.

Při tvorbě aplikace jsem byl inspirován aktuálními potřebami a požadavky odborníků v oboru prodeje automobilů. Snažili jsme se také využít nejnovějších trendů a technologií, abychom vytvořili aplikaci, která bude odpovídat aktuálním potřebám a bude sloužit jako nástroj pro rozvoj celého odvětví.

## 2. Technologie použité při vývoji

Během vývoje této aplikace bylo použito nespočet technologií, knihoven či návrhových vzorů, kterými jsem se řídil, nebo které mi usnadnily vývoj. Ty hlavní, o kterých bych se rád zmínil, jsou uvedeny níže.

### 2.1. PHP – Laravel

PHP - Hypertext Preprocessor (pro zajímavost, první „P“ značí PHP) - je serverový skriptovací jazyk, který je široce podporován a využívá se pro vývoj webových aplikací (php.net, 2011). PHP kód se interpretuje na serveru a generuje HTML, které je posláno prohlížeči, takže uživatel vidí pouze výslednou stránku. Disponuje širokou škálou možností pro práci s daty, jako je přístup k databázím, manipulace s formulářovými údaji, generování a zpracování cookie nebo session proměnných. Tyto funkce umožňují vývojářům vytvářet složité aplikace, jako jsou například e-commerce stránky nebo diskusní fóra. Díky rozšířenosti tohoto jazyka má PHP širokou komunitu, která vytváří různé knihovny a moduly, které mohou vývojáři používat pro rychlejší vývoj.

Laravel je open-source PHP framework, který poskytuje širokou škálu funkcí a nástrojů pro jednodušší vytváření webových aplikací. Odlišuje se od ostatních PHP frameworků tím, že se snaží umožnit vývojářům vytvářet stránky rychle a efektivně, a zároveň zachovávat čistý a přehledný kód. Laravel nabízí širokou škálu funkcí, včetně jednoduchého způsobu práce s databázemi, autentifikaci, autorizaci a mnoho dalšího. Konkrétně používá návrhové vzory jako je například MVC - Model View Controller. Tento návrhový vzor se využívá k oddělení logiky, databáze a samotné stránky do jednotlivých souborů pro lepší orientaci. Laravel mi také umožnil snadnou implementaci bezpečnostních funkcí jako je například Cross-Site Request Forgery neboli CSRF. Tato ochrana spočívá v odeslání unikátního tokenu, který je vygenerován právě pro jednu transakci formuláře. Server ověří, zda token poslaný společně s formulářem odpovídá tokenu uloženého v session a následně požadavek buďto přijme nebo zamítne (laravel.com, 2022). Je tu i spousta dalších ochranných opatření mezi které patří například zabezpečení proti SQL injkcím, ochrana proti XSS útokům, session hijacking či hashování uživatelských hesel.

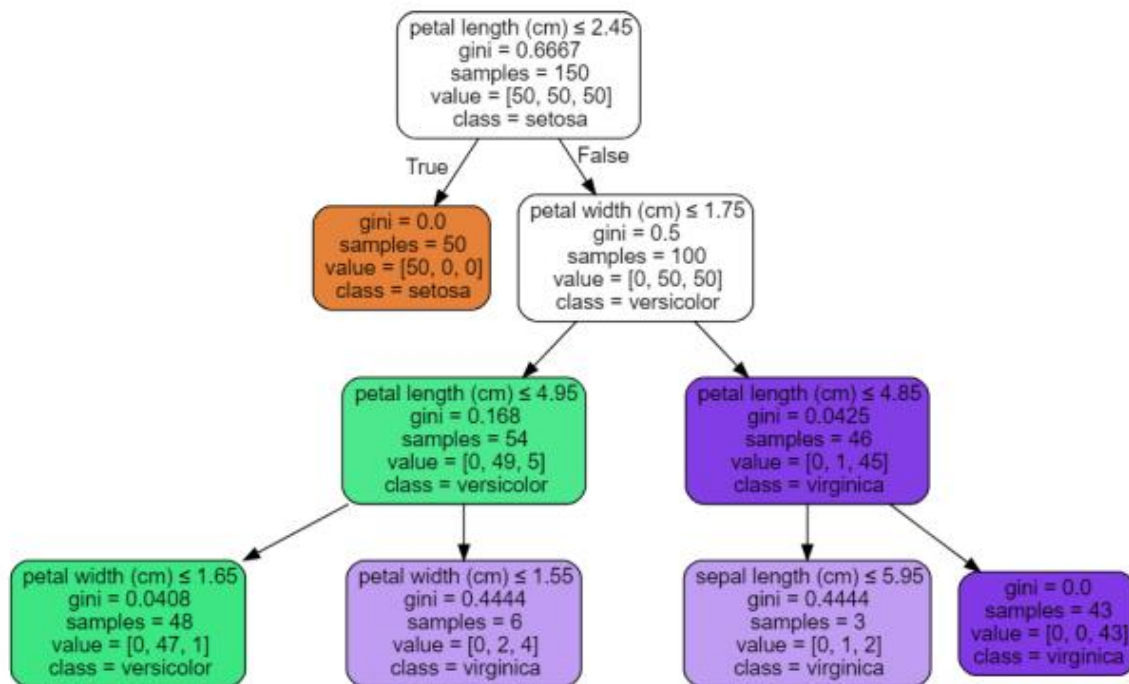
### 2.2. Scrapy

Pro sběr dat o automobilech jsem použil knihovnu Scrapy, která mi umožnila snadno procházet největší český autobazar a získat potřebná data, která byla následně využita ke strojovému učení. Scrapy umožňuje automatizovat proces sběru data a získat velké množství informací za krátkou dobu. Knihovna navíc poskytuje intuitivní uživatelské rozhraní pro nastavení a úpravu různých parametrů sběru dat. Scrapy se odlišuje od jiných knihoven pro web scraping tím, že nabízí funkce pro automatizaci sběru dat, jako jsou nastavitelná a jednoduchá pravidla pro výběr dat a funkce pro uložení dat. Také nabízí funkce pro úpravu a zpracování dat, jako je filtrování, výběr a transformace dat (docs.scrapy.org, 2022). Prozatím je sběr dat naprogramován pro český autobazar, ale cílem predikce cen automobilů je zohlednit i okolní státy. Proto bych rád v dalším vývoji aplikace začal sbírat data i z autobazarů v Německu nebo v Polsku.

### 2.3. Vývoj machine-learningového modelu

Data jsou načtená z databáze pomocí knihovny „mysql.connector“, která umožní snadné a přehledné připojení do databáze. S uložením dat do proměnné pomohla knihovna Pandas, která slouží k analyzování a manipulaci datových struktur. K rozdělení dat na testovací a trénovací slouží knihovna sklearn.model\_selection ve které je přímo navržena funkce

`train_test_split`. Tato funkce přijímá celý datový soubor a poměr, ve kterém se má soubor rozdělit. Výstupem je dvojice trénovacích a testovacích dat. V tomto okamžiku bylo zapotřebí vybrat algoritmus, který vytvoří následný model. Po vyzkoušení `DecisionTreeRegressor`, `LinearRegression` a `RandomForestRegressor` jsem nakonec vybral `DecisionTreeRegressor` se 11 skórem 1.0, který absolutně vyhovoval. Tato třída využívá teorii rozhodovacích stromů, což ji činí vhodnou pro řešení úloh zaměřených na predikci hodnot. Rozhodovací strom se snaží oddělit data na menší a jednoznačnější skupiny, aby bylo možné určit vztah. Tyto vztahy se využívají na predikci hodnot na základě nových dat.



Obrázek 1: Ukázka `DecisionTreeRegressor` na grafu ([scikit-learn.org](http://scikit-learn.org), 2007)

V poslední řadě stačilo použít knihovnu `joblib` pro serializaci a deserializaci objektů. Hlavním cílem této knihovny je umožnit ukládání a načítání objektů bez nutnosti jejich přepočítávání, což značně snižuje čas na predikci ceny.

## 2.4. Flask

Flask je jednoduchý framework pro vývoj webových aplikací v jazyce Python. Zaměřuje se na minimalismus a flexibilitu, což vývojářům značně usnadňuje používání jiných knihoven nebo nástrojů, které potřebují ([flask.palletsprojects.com](http://flask.palletsprojects.com), 2022). Zvolil jsem ho z důvodu, že dokáže využívat knihovny potřebné pro predikci ceny u mého modelu. Vytvořil jsem v něm RestAPI, které z URL adresy extrahuje zvolené hodnoty. Ty následně zpracuji do pole pomocí knihovny `numpy` a vložím do načteného modelu. Výsledek z modelu je hodnota, kterou zakóduji do formátu JSON a odešlu jako odpověď uživateli.

```
// https://api.petrbohac.eu:5001/car/price/7/52/1/3/2004/325000

{
  "price": [
    135000.0
  ]
}
```

Obrázek 2: Predikce ceny auta Škoda Fabia, palivo Diesel s manuální převodovkou z roku 2004 a s nájzdem kilometrů 325000

Jako druhé RestAPI jsem vytvořil ověření platnosti dálniční známky pro vozidla s českou poznávací značkou. Toto API odkazuje na koncový bod rozhraní edalnice.cz, které zpátky pošle JSON s začátkem a koncem platnosti dálniční známky.

```
// https://api.petrbohac.eu:5001/edalnice/6AN6002

{
  "charges": [
    {
      "fuelType": "OTHER",
      "isCurrentlyValid": true,
      "validSince": "2023-01-31T00:00:00.000+01:00",
      "validUntil": "2024-01-30T23:59:59.999+01:00"
    }
  ],
  "isGivenExemption": false
}
```

Obrázek 3: Kontrola dálniční známky pro vozidlo s českou SPZ

## 2.5. Apache

Apache je nejrozšířenější open-source HTTP webový server. Má řadu výhod, mezi které patří například:

- flexibilita, která umožňuje vývojářům upravovat a personalizovat svůj webový server pomocí modulů
- stabilita je u Apache primární vlastností, což zaručuje dostupnost stránek po dlouhou dobu bez problémů
- podporuje také řadu bezpečnostních funkcí jako je šifrování SSL a TLS nebo kontrolu přístupu na stránku

Na webový server jsem umístil doménu, na které jsem vytvořil 2 subdomény. Na první subdoméně je spuštěna Laravel aplikace pro simulaci nasazení projektu do produkce a také k testování, se kterým mi převážně pomohla rodina, nebo spolužáci. Na druhé subdoméně

jsem spustil RestAPI společně s dokumentací, aby se mohl například můj vytrénovaný machine learningový model používat i v jiných aplikacích a nejenom v té mé.

## 2.6. Virtual Private Server – Linux

Pro automatizaci služeb a nebo také pro provoz již zmíněného webového serveru je zapotřebí vlastnit nějaký server. Jelikož koupě vlastního fyzického serveru je příliš drahá a pro tyto účely bych řekl až zbytečná, tak jsem zvolil volbu VPS. Virtual Private Server (VPS) je typ serveru, který umožňuje uživatelům sdílet fyzický hardware, ale s vlastním izolovaným prostředím, které se chová jako samostatný server. Jedná se o jednu z nejpoužívanějších forem cloudového hostingu a poskytuje uživatelům plnou kontrolu nad svým prostředím.

Výhody VPS serverů jsou mnohé. Jednou z nich je cenová dostupnost, protože uživatelé neplatí za celý fyzický server, ale pouze za část zdrojů, které používají. To znamená, že výdaje jsou o mnoho nižší než u dedikovaného serveru. Další výhodou je flexibilita, protože uživatelé mohou snadno upravit své prostředí podle svých potřeb.

Většina VPS serverů běží na operačním systému Linux, který je vhodný pro tuto technologii kvůli své otevřenosti a flexibilitě. Linux poskytuje uživatelům možnost instalovat a konfigurovat aplikace podle svých potřeb, a to bez omezení, která mohou být kladená jinými operačními systémy. Kromě toho je Linux dobře znám pro svou stabilitu a spolehlivost, což jsou důležité faktory pro servery, které mají běžet nepřetržitě. V mém případě jsem zvolil distribuci Linuxu, Debian.

Na server jsem nainstaloval Python společně s knihovnou Scrapy, abych mohl automatizovat již vytvořený kód web crawleru. Pro automatizaci jsem využil službu Linuxu, CRON. Což je plánovač úloh, který umožňuje uživatelům spouštět příkazy v určitých časových intervalech. Uživatelé mohou nastavit spouštění příkazů libovolný den, týden, měsíc nebo rok. CRON se mimo jiné hojně využívá pro automatizaci opakujících se úloh, jako je zálohování dat, aktualizace systému, a nebo v mém případě spouštění web crawleru ([linuxconfig.org](http://linuxconfig.org), 2023). Můj vytvořený CRON spočívá v automatickém spouštění Scrapy programu jednou za měsíc s následnou aktualizací machine learningového modelu. Díky této funkci model využívá stále nová a relevantní data, která následně zvyšují přesnost predikce.

## 2.7. Databáze – MariaDB

Databázi, kterou jsem použil pro tuto aplikaci je SQL databáze MariaDB. MariaDB je open-source relační databázový management systém, který vznikl jako náhrada za MySQL. Tato databáze má řadu vylepšení oproti MySQL, jako jsou například lepší funkce pro správu databází, výkonnější dotazy a rychlejší zpracování dat. MariaDB jsem nainstaloval na VPS a pro správu jsem zvolil nejpoužívanější nástroj phpMyAdmin. Pro správu tabulek v databázi se postarali migrace v Laravelu. Migrace spočívají v přehledné správě tabulek a údržbou databázových schémat v jednotném formátu a bez nutnosti ručního upravování SQL příkazů.

```

Schema::create( table: 'car_info', function (Blueprint $table) {
    $table->id();
    $table->string( column: 'engine_type', length: 255)->nullable( value: false);
    $table->integer( column: 'engine_volume_cm3')->nullable( value: false);
    $table->integer( column: 'engine_power_kw')->nullable( value: false);
    $table->integer( column: 'trunk_space')->nullable( value: false);
    $table->integer( column: 'fuel_tank_space')->nullable( value: false);
    $table->integer( column: 'max_speed')->nullable( value: false);
    $table->integer( column: 'weight')->nullable( value: false);
    $table->integer( column: 'length')->nullable( value: false);
    $table->integer( column: 'width')->nullable( value: false);
    $table->integer( column: 'height')->nullable( value: false);
    $table->string( column: 'vin', length: 17)->nullable( value: false);
    $table->smallInteger( column: 'user_has_vin_car')->nullable();
    $table->smallInteger( column: 'decibel')->nullable();
    $table->date( column: 'valid_since')->nullable();
    $table->date( column: 'valid_until')->nullable();
    $table->string( column: 'license_plate')->nullable();
    $table->integer( column: 'price')->nullable( value: false);
    $table->integer( column: 'year')->nullable( value: false);
    $table->integer( column: 'km')->nullable( value: false);
    $table->unsignedInteger( column: 'car_brand_id')->nullable( value: false);
    $table->unsignedInteger( column: 'car_model_id')->nullable( value: false);
    $table->unsignedInteger( column: 'car_fuel_id')->nullable( value: false);
    $table->unsignedInteger( column: 'car_transmission_id')->nullable( value: false);

    $table->foreign( columns: 'car_brand_id')
        ->references( columns: 'car_brand_id')->on( table: 'car_brands');

    $table->foreign( columns: 'car_model_id')
        ->references( columns: 'car_model_id')->on( table: 'car_models');

    $table->foreign( columns: 'car_fuel_id')
        ->references( columns: 'car_fuel_id')->on( table: 'car_fuels');

    $table->foreign( columns: 'car_transmission_id')
        ->references( columns: 'car_transmission_id')->on( table: 'car_transmissions');

    $table->softDeletes();
    $table->timestamp( column: 'created_at')->useCurrent();
    $table->timestamp( column: 'updated_at')->useCurrent();
});

```

Obrázek 4: Ukázkový kód migrace v Laravelu včetně referencí na další tabulky

### 3. Realizace vývoje projektu

Abych mohl docílit výsledné aplikace, která má být opravdu užitečná, musel jsem se nejprve poradit s experty, kteří se v této branži pohybují už dost dlouho a nasbírali za tu dobu nějaké zkušenosti. Na získané informace jsem posléze cílil a implementoval je do aplikace, aby i pro běžné zákazníky byly užitečné.



## 4. Požadavky pro spuštění

Tato webová aplikace není nějak náročná na chod na webovém serveru. Proto u většiny případů může aplikace běžet pod subdoménou webu autobazaru. Podmínkou webového serveru je nainstalování PHP, Pythonu včetně potřebných knihoven a databáze. Laravel si s databází umí poradit a spustí uložené migrace téměř na jakémkoliv typu, ať už je to MySQL, SQLite nebo SQL server. Toto zajišťuje framework prostřednictvím nainstalovaných databázových ovladačů. Pro připojení do databáze se používají různé konfigurační soubory, ten hlavní se nazývá „.env“. Zde se vyplní mimo jiné parametry pro připojení jako je například adresa serveru, název databáze, uživatelské jméno a heslo.

Samotná instalace aplikace spočívá vložením souborů do rootovské složky, odkud se web má spouštět a modifikovat tuto cestu do „public“ složky Laravelu. Dále musíme zajistit, aby přístup do těchto složek měl i uživatel a nejen root, jinak bychom se do aplikace nedostali kvůli přístupovým právům. Nesmíme zapomenou vytvořit již zmíněný konfigurační soubor „.env“, bez kterého je aplikace nespustitelná. Po úspěšném vyplnění hodnot v tomto souboru nám už jen stačí vygenerovat aplikační klíč. Aplikační klíč v souboru „.env“ je automaticky generovaný důležitý prvek pro zabezpečení aplikace a ochranu dat. Tento klíč slouží k šifrování dat a tokenů v aplikaci, což pomáhá předcházet útokům zvenčí a zlepšuje zabezpečení celé aplikace.

Po nasazení této aplikace zbylo na admina už jen vytvořit zaměstnanecké účty, pod kterými se budou zaměstnanci přihlašovat.

A nakonec, pro využití této aplikace bude potřeba, aby zaměstnanec měl jakýkoliv počítač, který má přístup k internetu a stažený webový prohlížeč

## 5. Základní funkce aplikace

### 5.1. Přihlášení do systému

Přihlašovací formulář je u většiny webových aplikací nezbytnou součástí, obzvlášť pokud se jedná o nějaký interní systém.

Formulář slouží k ověření totožnosti uživatele a umožňuje mu získat přístup k aplikaci. Do aplikace se přihlašuje pomocí emailu a hesla. Email slouží k identifikaci uživatele a heslo k ověření jeho identity.

Uložení hesla je šifrováno, aby bylo dodrženo zabezpečení uživatelských účtů a minimalizace rizika úniku citlivých dat. Hashování znamená, že heslo uživatele je převedeno na řetězec znaků, který je matematicky odvozen z původního hesla a nelze jej snadno převést zpět na původní heslo. Tento algoritmus mimo jiné zajišťuje, že hesla nebudou uložena v čitelné podobě v databázi. Kromě hashování hesel při ukládání do databáze je také důležité hashovat heslo při přihlášení. To znamená, že uživatelské heslo je při zadání do přihlašovacího formuláře okamžitě převedeno na hash a porovnává se s hashem uloženým v databázi. Pokud se shodují, uživatel je přihlášen.

Vítejte zpět!  
Přihlašte se prosím

E-mail

Heslo

[Přihlásit se](#)

[Zapomenuté heslo?](#)

Ještě nemáte účet? [Vytvořte si ho!](#)

© 2022–2023 Petr Boháč.

Obrázek 5: Ukázka přihlašovacího formuláře

## 5.2. Správa uživatelů

Při přihlašovacím systému v aplikaci nesmí chybět také správa uživatelů. Ta spočívá v zobrazení všech systémových uživatelů, jejich snadné upravy, smazání nebo změny hesla. Pro vytvoření nového uživatele bude potřeba vyplnit formulář s povinnými položkami jako je jméno, email a role. Pokud si uživatel nezadá sám heslo, nebo mu je účet vytvořen přes správce, je zde možnost si nechat poslat odkaz na formulář na uvedený email. Tento odkaz je zabezpečený hashem v url adrese, abychom předešli podvodům například při uhádnutí url adresy atp. Samotný odkaz je také platný pro určitou dobu a nelze vytvořit na tomto odkazu heslo podruhé. Každý uživatel v seznamu bude moci být snadno editován nebo vymazán. Tato funkce bude sloužit pro úpravu údajů, jako je již zmíněné jméno, nebo uživatelská role. Pokud uživatel zapomene své heslo, bude mu nebo správci systému umožněno resetovat jej. Po kliknutí na odkaz pro reset hesla bude uživateli na uvedený email zaslán odkaz také s hash url pro vytvoření nového hesla.

### 5.3. Feedback

Feedback ve webové aplikaci je funkcionalita, která by za mě neměla chybět ani v těch nejmenších aplikacích. Jeho implementace je pro uživatele výhodná, protože umožňuje rychlejší řešení problémů a zlepšení zážitků z aplikace. Vývojáři zase získají cennou informaci o chování uživatele a mohou tak aplikaci dále vylepšovat. Existuje mnoho různých typů feedbacku, které mohou být implementovány do webové aplikace. Například formuláře na hodnocení, možnost přidání komentáře nebo tlačítka na hlášení chyb. Já zvolil možnost jednoduchého formuláře, ve kterém uživatel napíše problém do textového pole a po odeslání ho dostane na email správce systému.

### 5.4. Evidence aut

Evidence aut je klíčovou součástí aplikace. Umožňuje zaměstnancům přidávat, upravovat a mazat informace o automobilech, jako je rok výroby, najeté kilometry, označení motoru, zvihový objem, kW, VIN, cena, velikost nádrže, velikost kufru, maximální rychlost, hmotnost, délka, šířka, výška, převodovka, palivo, SPZ, model a značka. Zákazníci mohou tyto informace snadno získat díky QR kódu, který je umístěn přímo na prodávaném autě. Stačí pouze naskenovat QR kód pomocí aplikace, která následně přesměruje uživatele na stránku s kompletními informacemi o daném vozidle. Obrázek QR kódu je navíc při stažení ze stránky ve formátu SVG, což dává volnost roztahování a zvětšování podle potřeby bez ztráty kvality.



Obrázek 6: Ukázka funkčního QR kódu

## Přehled informací o vozidle

Škoda Kodiaq 6AN6002  
TMBLJ7NS4KN401036

 Obsah motoru <b>1989cm<sup>3</sup></b>	 Výkon motoru <b>180kW</b>	 Velikost kufru <b>610l</b>	 Velikost nádrže <b>55l</b>
 Maximální rychlost <b>185km/h</b>	 Hmotnost vozidla <b>1690kg</b>	 Délka vozidla <b>2010mm</b>	 Šířka vozidla <b>1690mm</b>
 Výška vozidla <b>1780mm</b>	 VIN <b>TMBLJ7NS4KN401036</b>	 Palivo <b>Benzín</b>	 Převodovka <b>Automatická</b>

[Navštívit stránku prodejce](#)

Obrázek 7: Výsledná stránka s informacemi o vozidle po načtení QR kódu

V aplikaci je také implementováno měření hlučnosti uvnitř vozidla. Zaměstnanci stačí před vyjetím z autobazaru položit notebook s otevřenou stránkou pro měření hlučnosti vedle sebe na bezpečné místo a po celou jízdu se o to nemusí starat. Aplikace sama pozná nejvyšší naměřenou hlučnost, kterou následně dočasně uloží do proměnné. Až řidič zastaví vozidlo, může se na tuto hodnotu podívat a uložit ji ke specifikacím vozidla. Pokud by náhodou nastala chyba, je možné naměřené decibely resetovat a začít test znovu.

Při zadávání ceny auta hraje velikou roli situace na trhu, proto jsem za pomoci machine learningového modelu a velké spousty dat schopen odhadnout cenu auta pouze podle nejdůležitějších parametrů, jako je například značka, model, palivo, převodovka, rok auta a najeté kilometry. Pro vozový park to znamená jednoduché ověření ceny, aby auta neprodávali pod cenou nebo naopak za příliš vysokou cenu.

## 5.5. Tvorba faktur

Vystavení faktury u vozového parku je nezbytná věc, a proto i v tomto systému nechybí tvorba faktur. Při tvorbě faktury uživatel zadává pouze základní údaje o kupci, jako je například jméno, příjmení, email, telefon a adresa. Následuje přidání položek na fakturu přes políčka, která jsou dynamicky generována. To znamená, že pokud zaměstnanec zaplní poslední volné políčko, automaticky se mu tam přidá jedno další volné. Toto dynamické přidávání políček se náramně hodí u formuláře, kde nevíme, kolik políček bude potřeba, jako například zde. U každého políčka lze zadat název produktu, popis, kvantitu, cenu za kus a slevu v procentech, která je nepovinná.

[← Zpět](#)

**Údaje u kupci**

Jméno a příjmení	Email	Telefon	Adresa
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

**Položky**

Název	Popis	Kvantita	Cena za 1 kus	Sleva v %	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="0"/>	<input type="button" value="Delete"/>

Obrázek 8: Formulář pro vystavení faktury

V aplikaci jsou také k dispozici další funkce pro tvorbu faktur, jako například možnost nastavení předvoleného textu pro fakturu nebo možnost tisku či uložení faktury ve formátu PDF.

Jelikož aplikace ukládá tyto faktury a také jejich cestu nebo informace o uživateli do databáze, je možné pro zaměstnance se kdykoliv zpětně vrátit k prodaným položkám bez nějaké větší námahy.

Tento systém by mohl v budoucí fázi kladně prospět k vytvoření CRM modelu, který se zaměřuje na správu vztahů se zákazníky a je často používán pro zlepšení obchodních vztahů s klienty. Informace o fakturách a jejich položkách by byly propojeny se zákaznickými záznamy. CRM model by pomohl udržovat a zlepšovat vztahy se zákazníky tím, že by poskytoval nástroje pro sledování a analyzování interakcí s nimi. Například by bylo možné sledovat, jak často si zákazníci prohlížejí nabízené vozy, jaké jsou jejich preference a zájmy nebo také analyzovat zakoupená zboží. Na základě toho by bylo možné nabídnout cílenější marketing nebo speciální nabídky. Využití CRM modelu v aplikaci pro evidenci aut by mohlo výrazně zlepšit komunikaci se zákazníky, zvýšit prodej a zlepšit vztahy s prodeji.

## **5.6. Přínos aplikace pro uživatele a firmy**

Tato webová aplikace přináší uživatelům mnoho výhod a zlepšení při správě vozového parku. Díky aplikaci mohou uživatelé snadno a pohodlně přidávat nové automobily do svého vozového parku a udržovat přehled o aktuálním stavu vozidel. Funkce generování QR kódu s technickými specifikacemi automobilů umožňuje zákazníkům snadno a rychle získat veškeré informace o vozidlech, což zvyšuje transparentnost a důvěryhodnost prodeje. Aplikace také nabízí správu uživatelů, což zjednodušuje a usnadňuje přidělování rolí a oprávnění v rámci firmy. Díky predikci ceny automobilů pomocí umělé inteligence mohou zaměstnanci lépe plánovat své náklady a optimalizovat svůj vozový park.

## 6. Závěr

Cílem mé práce bylo vytvořit uživatelsky příjemnou aplikaci, která by sloužila jak majitelům vozových parků, tak i zákazníkům, kteří mají zájem o koupi či pronájem automobilu. Ve webové aplikaci se promítly zkušenosti i přání osob, kteří se v tomto oboru pohybují, proto věřím, že vytvořená aplikace bude přínosem pro obě strany. Aplikace by měla pomoci k ověření dat nabízeného automobilu a tím i k větší poctivosti prodeje.

Celkově lze říct, že aplikace přináší mnoho výhod jak pro firmy, tak pro uživatele. Usnadňuje proces správy vozového parku, zvyšuje transparentnost a důvěryhodnost prodeje automobilů a umožňuje díky predikci ceny aut také plánovat a optimalizovat náklady na provoz vozového parku. Funkce generování QR kódu s technickými specifikacemi zvyšuje kvalitu služeb a poskytuje zákazníkům přesné a relevantní informace.

Výhody aplikace vycházejí ze spolupráce se specialisty v oboru prodeje automobilů, kteří přispěli k rozšíření funkcionalit aplikace. Díky nim je aplikace přizpůsobena potřebám zákazníků a zaručuje vysokou kvalitu služeb. Věřím, že tato aplikace bude úspěšná a poskytne uživatelům a firmám mnoho výhod a zlepšení.

## 7. Použitá literatura

- [1] PHP: PHP Manual - Manual . PHP: Hypertext Preprocessor [online]. Copyright © [cit. 18.02.2023]. Dostupné z: <https://www.php.net/manual/en/>
- [2] CSRF Protection - Laravel - The PHP Framework For Web Artisans. Laravel - The PHP Framework For Web Artisans [online]. Copyright © 2011 [cit. 18.02.2023]. Dostupné z: <https://laravel.com/docs/8.x/csrf#main-content>
- [3] 1.10. Decision Trees — scikit-learn 1.2.1 documentation. scikit-learn: machine learning in Python — scikit-learn 0.16.1 documentation [online]. Copyright © 2007 [cit. 19.02.2023]. Dostupné z: <https://scikit-learn.org/stable/modules/tree.html>
- [4] Welcome to Flask — Flask Documentation (2.1.x). 2022 [online]. Copyright © 2011 [cit. 20.02.2023]. Dostupné z: <https://flask.palletsprojects.com/en/2.1.x/>
- [5] Linux Crontab Reference Guide [cit. 26.03.2023]. Dostupné z: <https://linuxconfig.org/linux-crontab-reference-guide>

## 8. Přílohy

---

### About the project

---

This project is a Laravel-based web application designed for the administration of a vehicle fleet, as well as for customers who want to create their own accounts, store their invoices, and access information about the cars they have purchased from the fleet.

The application also features a unique predictive pricing tool that allows users to estimate the value of their car based on various factors, such as brand and model, fuel, transmission, year, and mileage. With this tool, anyone can get a better idea of how much their car is worth and make informed decisions about selling or trading it in.

Whether you're managing a fleet of vehicles or simply looking for a convenient way to keep track of your car's information and value, this Laravel web application has everything you need to streamline your processes and make better decisions.

### Prerequisites

---

- Version of [PHP 8.1](#) or higher
- [Composer](#)
- Any database (tested on MySQL - [MariaDB](#))

*Příloha 1: README.md 1/3*

---

### Installation

---

This section is focusing on how to set up this project locally.

1. Clone the GitHub repository:

```
git clone git@github.com:W0lf1n/CRM.git
```

2. Install dependencies:

```
composer install
```

3. Create a copy of the '.env.example' file and rename it to '.env':

```
cp .env.example .env
```

4. Generate an application key:

```
php artisan key:generate
```

*Příloha 2: README.md 2/3*



5. Edit the '.env' file to set your database credentials, mail server and other settings:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=your_database_name
DB_USERNAME=your_database_username
DB_PASSWORD=your_database_password

API_KEY_SKLONOVANI=your_api_key on sklonovani-jmen.cz

MAIL_MAILER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=7076f64e887f14
MAIL_PASSWORD=dc92af1e281603
MAIL_ENCRYPTION=tls
MAIL_FROM_ADDRESS=null
MAIL_FROM_NAME="${APP_NAME}"
```

6. Run database migrations:

```
php artisan migrate
```

7. Serve the application:

```
php artisan serve
```

8. Run the unit tests

```
php artisan test
```

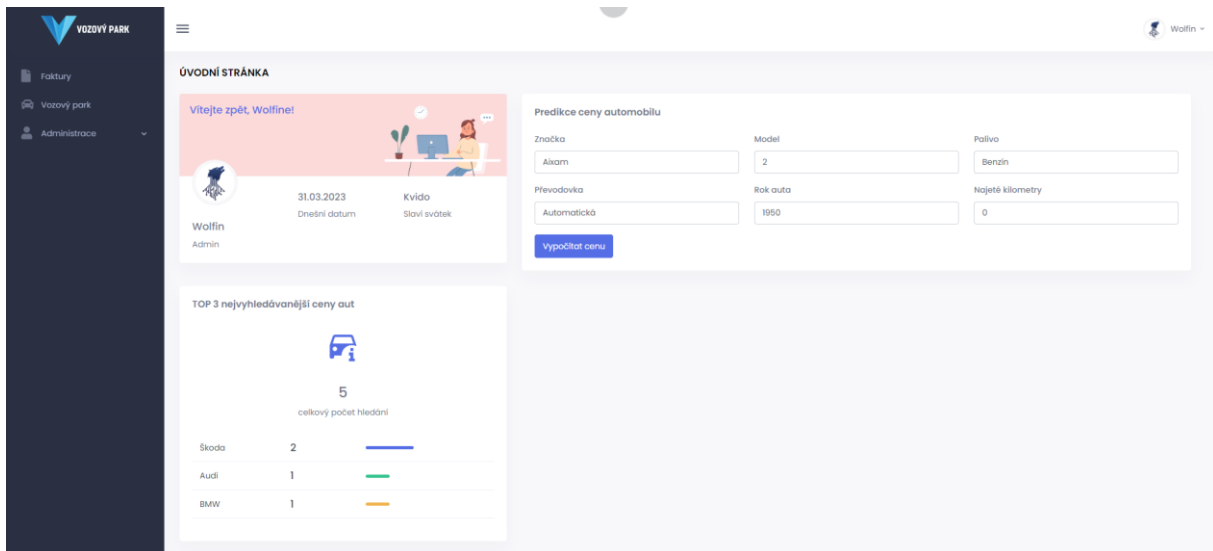
*Příloha 3: README.md 3/3*

```

@foreach( \App\Models\SideMenu::active()->onlyParent()->get() as $menu )
@if( Route::has( $menu->route) && (auth()->user()->role == $menu->for_role) || (auth()->user()->role == 1) )
    @if( $menu->children->count() == 0)
        @if( $menu->route == '' )
            <li>
                <a href="javascript: void(0);" class="has-arrow waves-effect">
                    <i class="{{ $menu->icon }}"></i>
                    <span>{{ __( $menu->name ) }}</span>
                </a>
            </li>
        @else
            <li {{ request()->routeIs($menu->route) ? 'active':'' }}>
                <a href="{{ route($menu->route) }}" class="waves-effect">
                    <i class="{{ $menu->icon }}"></i>
                    <span>{{ __( $menu->name ) }}</span>
                </a>
            </li>
        @endif
    @else
        <li {{ request()->routeIs($menu->route) ? 'active':'' }}>
            <a href="javascript: void(0);" class="has-arrow waves-effect">
                <i class="{{ $menu->icon }}"></i>
                <span>{{ __( $menu->name ) }}</span>
            </a>
            <ul class="sub-menu" aria-expanded="false">
                @foreach( $menu->children as $child)
                    <li>
                        <a href="{{ route($child->route) }}">
                            {{ __( $child->name ) }}
                        </a>
                    </li>
                @endforeach
            </ul>
        </li>
    @endif
@endif
@endforeach

```

*Příloha 4: Sidebar vykreslený pomocí hodnot z databáze*



**VOZOVÝ PARK**

Faktury  
Vozový park  
Administrace

ÚVODNÍ STRÁNKA

Vítejte zpět, Wolfin!

Wolfin Admin

31.03.2023 Dnešní datum  
Kvído Slavi svátek

**Predikce ceny automobilu**

Značka:  Model:  Palivo:

Převodovka:  Rok outa:  Najojeté kilometry:

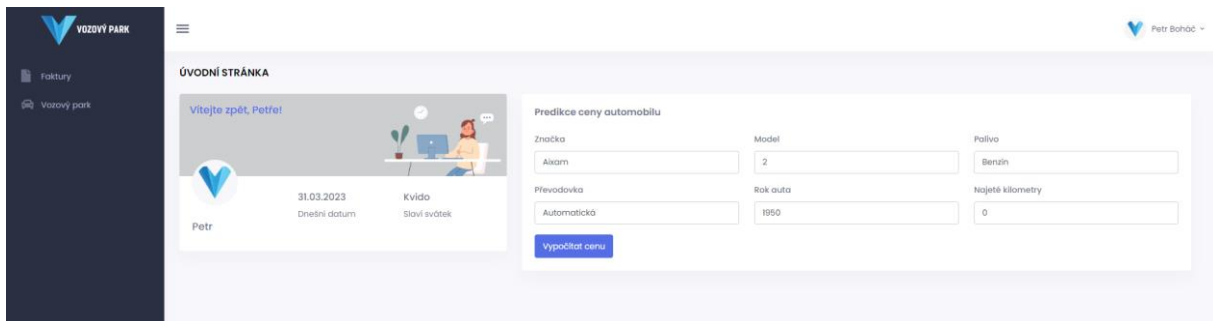
[Vypočítat cenu](#)

TOP 3 nejvyhledávanější ceny aut

5 celkový počet hledání

Škoda	2	
Audi	1	
BMW	1	

Příloha 5: Stránka z pohledu administrátorského účtu



**VOZOVÝ PARK**

Faktury  
Vozový park

ÚVODNÍ STRÁNKA

Vítejte zpět, Petr!

Petr

31.03.2023 Dnešní datum  
Kvído Slavi svátek

**Predikce ceny automobilu**

Značka:  Model:  Palivo:

Převodovka:  Rok outa:  Najojeté kilometry:

[Vypočítat cenu](#)

TOP 3 nejvyhledávanější ceny aut

5 celkový počet hledání

Škoda	2	
Audi	1	
BMW	1	

Příloha 6: Stránka z pohledu zákaznického účtu

## FAKTURA

Serial No. **INNJ-00001**

Invoice date: **11.03.2023**

### Seller

**Ukázkový vozový park**

Adresa: Zámecká 216-215, 570 01, Litomyšl

VAT: 95175386

Telefon: 123456789

### Buyer

**Petr Boháč**

Adresa: Bryksova 941/33

Email: petr.bohac61@gmail.com

Telefon: +420607648456

Description	Qty	Price	Discount	Sub total
Škoda Kodiaq	1	980000.00 Kč	49000.00 Kč	931000.00 Kč
Total discount				49000.00 Kč
Taxable amount				931000.00 Kč
Tax rate				21%
Total taxes				195510.00 Kč
Total amount				<b>1126510.00 Kč</b>

Amount in words: One million one hundred twenty-six thousand five hundred ten CZK and zero ct.

Please pay until: 18.03.2023

Příloha 7: Ukázka vygenerované faktury