

Střední průmyslová škola elektrotechniky a informatiky, Ostrava

# 40. Webová aplikace – půjčovna aut

## Maturitní práce

Autor práce: Jonatan Lepík  
Vedoucí práce: RNDr. Rostislav Miarka, Ph. D.  
Třída: I4C  
Školní rok: 2022/2023

Zadání práce – vložte listy se zadáním podepsaným ředitelem školy

Prohlašuji, že předložená práce je mým původním dílem, které jsem vypracoval samostatně. Veškerou literaturu a další zdroje, z nichž jsem při zpracování čerpal, v práci řádně cituji a jsou uvedeny v seznamu použité literatury a zdrojů informací.

V Ostravě 2. 4. 2023

podpis:

### **Licenční ujednání**

Ve smyslu §60 autorského zákona č. 121/2000 Sb. poskytuji, Střední průmyslové škole elektrotechniky a informatiky, Ostrava, příspěvkové organizaci, Kratochvílova 1490/7, 702 00 Ostrava, bezplatně oprávnění k výkonu práva (licenci) ke školnímu dílu (maturitní práci) užít v rozsahu a způsoby uvedenými v §12 až 23 autorského zákona.

Souhlasím se zveřejněním díla v rámci školní počítačové sítě pro potřeby studentů a zaměstnanců školy a pro potřeby výuky v souladu s §35(3) autorského zákona.

Souhlasím s použitím práce k propagaci školy.

V Ostravě 2. 4. 2023

podpis:

### **Anotace**

Tato práce byla napsána za účelem vytvoření dokumentace k maturitní práci s tématem webové aplikace – půjčovna aut. Popisuje postup vývoje, části napsaného kódu a návod k ovládání aplikace.

# Obsah

1	Úvod .....	5
1.1	Plánovaný postup řešení.....	5
2	Rozbor funkcionality webové aplikace.....	6
2.1	Stylování elementů pomocí frameworku Tailwind CSS.....	7
2.2	Řešení hlavní stránky .....	8
2.3	Stránka o vozidle a možnosti rezervace .....	11
2.4	Registrace a přihlašování.....	15
2.5	Uživatelské rozhraní.....	17
2.6	Administrační rozhraní .....	18
2.7	Header, navigační menu a footer.....	20
3	Závěr .....	22
4	Seznam příloh.....	23
5	Použité zdroje.....	24

# 1 Úvod

V dnešní době je auto nezbytnou součástí našeho každodenního života. Ale ne každý jej vlastní, nebo si ho může dovolit. Na tento problém existují tzv. autopůjčovny – půjčovny, kde si každý může vybrat a vypůjčit jeho zvolené auto. V poslední době se autopůjčovny stávají čím dál tím více populární a využívá je velká část lidí. Nevyužívají se pouze na jednodenní vypůjčení vozidla například na stěhování, ale i na několikátýdenní výlet do zahraničí.

Žijeme v moderní době, kde téměř každé podnikání vyžaduje vlastní webovou stránku. Nejenom proto, aby si vás mohli zákazníci rychleji dohledat, ale také pro jednodušší objednání daných služeb.

Cílem práce je vytvořit webovou aplikaci pro půjčovnu aut. Bude zde možné se zaregistrovat a zarezervovat si libovolné auto na určitý termín. Kromě uživatelského rozhraní zde bude také rozhraní pro správce stránky, kde bude možnost přidávat a upravovat vozidla, kategorie a uživatele. Celá webová aplikace bude uživatelsky přívětivá a jednoduchá na ovládání. Bude vyvíjena v jazyce HTML, CSS, PHP a Javascript (jQuery).

## 1.1 Plánovaný postup řešení

1. Vytvoření celkového nápadu a všech funkcí, která bude aplikace obsahovat
2. Navrnutí rozložení, designu a vybrání barev které se budou používat
3. Vytvoření databázového modelu
4. Naprogramování samotné aplikace
  - a. Vytvoření úvodní stránky
  - b. Vytvoření stránek na registraci a přihlášení
  - c. Vytvoření administračního rozhraní
  - d. Vytvoření uživatelského rozhraní
5. Finální úprava aplikace - přidání animací, doprovodných textů a stránek

## 2 Rozbor funkcionality webové aplikace

Celá webová aplikace je postavena na jazycích HTML, CSS, PHP a také z malé části Javascriptu.

Její princip závisí na rezervačním systému, který umožňuje pouze přihlášeným uživatelům si nezávazně zarezervovat vozidla, která jsou v zadaný termín dostupná. Po úspěšném odeslání se rezervace uloží do databáze se statusem *čekající*. Tato rezervace nyní čeká na potvrzení od administrátora stránky, který mezitím ověří čas vyzvednutí a celkovou realizaci objednávky s klientem. Nyní ji správce stránky může označit jako *potvrzenou*, a po konci rezervace jako *dokončenou*. V opačném případě ji může smazat, nebo označit jako *zrušenou*. Administrátor stránky přehledně na nástěnce uvidí, kolik nových rezervací čeká na potvrzení, celkovou tržbu a podobně. Každý klient si může taktéž zobrazit své rezervace a podle potřeby je upravit nebo smazat. Kromě toho si každý uživatel může změnit heslo.

Webová aplikace obsahuje také všechny základní informace potřebné pro půjčovnu aut, jako jsou jejich podmínky, výhody, kontakt a často kladené dotazy.

## 2.1 Stylování elementů pomocí frameworku Tailwind CSS

Vzhled a stylování je velice důležitou částí každé webové aplikace. Stylování elementů pomocí klasického CSS může být poněkud zdlouhavé a po delší době také velmi nepřehledné. Kvůli tomu je celá aplikace postavena na frameworku s názvem Tailwind CSS, který nám umožní jednoduché stylování elementů pomocí tříd. Vše potřebné, co lze udělat v klasickém CSS, jde udělat i zde.

Jak se styluje pomocí Tailwind CSS? Je to jednodušší, než to vypadá. Nejdříve u elementu, který chceme stylovat, přidáme třídu. Následně všechny styly, které chceme na elementu použít, píšeme do ní. Například pokud bychom chtěli nastavit odsazení zprava, použijeme třídu *mr-x* (*margin-right*). Místo x se dosadí potřebná hodnota v číslech. Po nastavení odsazení shora se použije *mt-x* (*margin-top*). Všechny třídy lze najít v dokumentaci na oficiálních stránkách Tailwind CSS.

Použitím tohoto frameworku se celkově urychlí práce se stylováním, zadáváním různých ID s unikátními jmény a podobně. Disponuje také několika přednastavenými styly například pro stíny nebo pro velikosti písma. Kromě těchto stylů má také velmi velkou výhodu v jednoduchosti responzivního designu.



```
1 <div class="mr-5 mt-2 shadow-xl text-lg text-red-500">
```

Obrázek 1: Základní styly pomocí Tailwind CSS

Responzivní design se dělá podobným způsobem – Tailwind má již předdefinované tzv. breakpointy, které můžeme využít při vytváření responzivního designu. Je jich hned několik, například small (sm) = 640 px, medium (md) = 768 px, large (lg) = 1024 px. Pokud je chceme využít, vždy napíšeme jejich zkratku uvedenou v závorce následovanou dvojtečkou, za kterou se napíše styl, který chceme použít. Opět je tento způsob oproti klasickému CSS mnohem jednodušší a přehlednější na použití.

V níže uvedeném příkladu je defaultně nastavená šířka na 100 %, pokud velikost obrazovky rovna nebo větší *medium*, nastaví se šířka na 50 %. Zda-li je velikost obrazovky rovna nebo větší *large*, nastaví se šířka na 25 %.



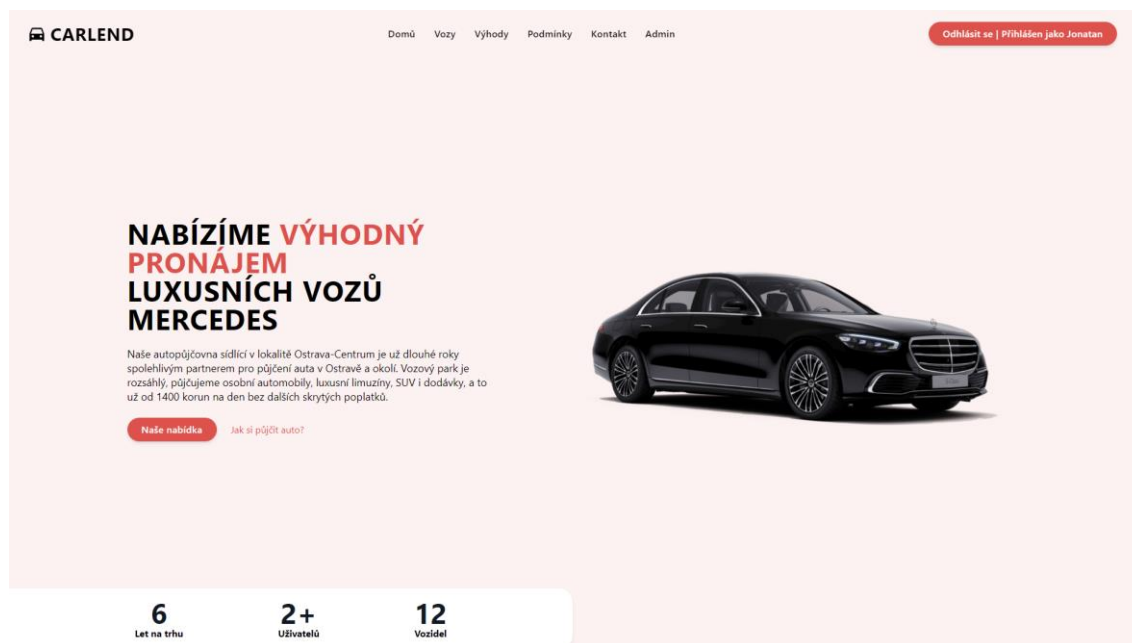
```
1 <div class="w-full md:w-1/2 lg:w-1/4"></div>
```

Obrázek 2: Základní styly pomocí Tailwind CSS - breakpointy

## 2.2 Řešení hlavní stránky

Hlavní stránka se skládá z větší části doprovodného textu, ale také ze samotného výběru vozidel.

Po prvotním zobrazení stránky se nám ukáže uvítací obrazovka se statistikami půjčovny. Tyto statistiky jsou dynamické a mění se v reálném čase podle údajů z databáze.



Obrázek 3: Vzhled úvodní stránky

Počet uživatelů se spočítá v níže uvedeném SQL dotazu, kde se využívá agregační funkce `COUNT()`. Stejně funkce se využije u dotazu na počet vozidel.

```
1 $sql = "SELECT COUNT(*) AS pocet FROM uzivatele";
2 $pocetUzivatelu = mysqli_query($pripojeni, $sql)->fetch_array();
```

Obrázek 4: Kód aplikace na získání počtu registrovaných uživatelů



Pokud se posuneme o jednu obrazovku níže, nalezneme seznam vozidel k vypůjčení.

Toto zobrazení vozidel se skládá z cyklu `while()` a funkce `fetch_array()` které se využívá v celé aplikaci. Tato funkce nám rozloží záznam z databáze přehledně do dvourozměrného asociativního pole, pomocí kterého lze data z databáze jednoduše na stránce zobrazit. V cyklu se pro každé vozidlo vyberou z databáze také všechny údaje o jeho kategorii, ve které je aktuálně zařazeno. Zobrazují se pouze ta vozidla, která u sebe mají status *dostupná*.

```
1 while (($vozidlo = mysqli_fetch_array($seznamVozidel)) != NULL) {
2     if($vozidlo["dostupnost"]) {
3         $sql = "SELECT * FROM kategorie WHERE id_k=" . $vozidlo["kategorie"];
4         $kategorie = mysqli_query($pripojeni, $sql)->fetch_array();
5
6         echo "Vozidlo: " . $vozidlo["nazev"] . " " . $kategorie["nazev"];
7     }
8 }
```

Obrázek 5: Kód aplikace na zobrazení všech vozidel z databáze

Dále se zde také nachází několik podmínek `if()`, které nám jednoduše vyřeší zformulování slova u počtu míst.

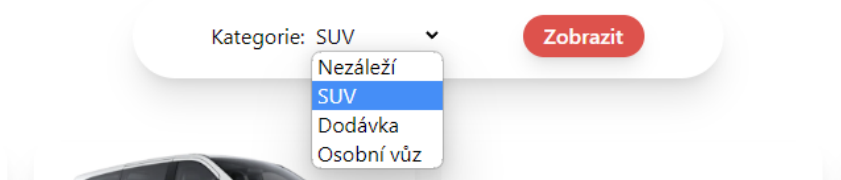
```
1 if($vozidlo["pocetmist"] == 1) {
2     echo "<i class='ri-group-fill'></i> " . $vozidlo["pocetmist"] . " Místo<br>";
3 }
4 else if($vozidlo["pocetmist"] <= 4) {
5     echo "<i class='ri-group-fill'></i> " . $vozidlo["pocetmist"] . " Místa<br>";
6 }
7 else {
8     echo "<i class='ri-group-fill'></i> " . $vozidlo["pocetmist"] . " Míst<br>";
9 }
```

Obrázek 6: Kód aplikace na zformulování slova u počtu míst

Seznam vozidel k rezervaci je dále doplněn o filtrování pomocí kategorie, ve které jsou vozidla zařazena.

## Naše nabídka vozů

Podívejte se na naši širokou nabídku užitkových i osobních vozů k zapůjčení



Obrázek 7: Vzhled filtrování kategorií u výběru aut na hlavní stránce

Tento filtr je udělaný pomocí formuláře s metodou GET, která nám danou kategorii vloží do URL adresy a s prvkem *select*, který nám umožní vybrat jednu možnost z rozbalovacího seznamu. Zprvu byla dána formuláři metoda POST, nicméně se narazil na problém – po kliknutí na vozidlo a poté na tlačítko zpět prohlížeč ukazoval znovu odeslání formuláře. Proto se zde použila metoda GET, která vybranou kategorii uloží.

Pro zobrazení všech dostupných kategorií v rozbalovacím seznamu nám poslouží níže uvedený kód. Kromě základní podmínky a cyklu *while* se muselo vyřešit, která položka bude defaultně vybraná. Tento problém se vyřešil pomocí podmínky, ve které se porovnává id kategorie z databáze a id kategorie, kterou zadal uživatel. Pokud se id shoduje, do prvku *option* se přidá atribut *selected*, který nám určí, která z položek bude defaultně označená.

```
1 echo '  
2 <option value="" . $kategorie["id_k"] . ''';  
3     if(isset($_POST["kategorie"])) {  
4         if($_POST["kategorie"] == $kategorie["id_k"]) {  
5             echo 'selected';  
6         }  
7     }  
8     echo '  
9 >' . $kategorie["nazev"] . '</option>';
```

Obrázek 8: Kód aplikace pro zobrazení všech dostupných kategorií ve formuláři

## 2.3 Stránka o vozidle a možnosti rezervace

Po rozkliknutí každého z vozidel se zobrazí detailní stránka o vozidle a možnosti zarezervování.

Kromě podrobnějších údajů o vozidle můžeme také vidět jeho ceník. Cena na den se vždy odvíjí od jeho celkové ceny a slevy u jeho kategorie. Každá půjčovna bude mít cenový model jiný, byl zvolen tento.

Pro výpočet ceny na určitý počet dnů se vždy začíná u jeho celkové ceny. Ta se vydělí tisícem a vynásobí dvěma. Pomocí této rovnice byla zjištěna optimální cena u každého vozidla. Pokud bude chtít půjčovna cenu vozidla zvýšit, jednoduše zvýší celkovou cenu a naopak. Kromě celkové ceny vozidla zde figuruje sleva u jednotlivých kategorií. Každá kategorie vozidel bude jinak oblíbená, proto je zde tato sleva. Pokud bude chtít půjčovna hromadně snížit cenu u vozidel kategorie SUV, zvýší slevu. Čím vyšší počet dnů zapůjčení, tím nižší bude cena za jeden den rezervace.


```
1  $cena = ($vozidlo["cena"] / 1000) * 2;
2
3  <div class="grid grid-cols-2 grid-rows-5 border p-2 rounded w-full md:w-1/2 lg:w-4/5">
4    <div class="font-semibold">1-3 dny</div>
5    <div>' . $cena . ' Kč/den</div>
6    <div class="font-semibold">4-6 dnů</div>
7    <div>' . $cena - $kategorie["sleva"] . ' Kč/den</div>
8    <div class="font-semibold">1 týden</div>
9    <div>' . $cena - ($kategorie["sleva"] * 1.5) . ' Kč/den</div>
10   <div class="font-semibold">2 týdny</div>
11   <div>' . $cena - ($kategorie["sleva"] * 2) . ' Kč/den</div>
12   <div class="font-semibold">1 měsíc</div>
13   <div>' . $cena - ($kategorie["sleva"] * 3) . ' Kč/den</div>
14   <div class="text-gray-400">Kauce</div>
15   <div class="text-gray-400">30 000 Kč</div>
16 </div>
```

Obrázek 9: Kód aplikace pro vypočítání ceny za určitý počet dnů

Pod údaji o vozidle se nachází rezervační formulář, který se zobrazí pouze přihlášeným uživatelům. V opačném případě je zde informace o nutnosti přihlášení a odkazem na přihlašovací stránku.

Formulář se skládá z datumu začátku a konce rezervace, času vyzvednutí a popisu objednávky. Po kliknutí na tlačítko *Nezávazně rezervovat* se nejprve zkontroluje, zda jsou datумы validní, a poté zdali je vozidlo na daný termín dostupné.


Validní datum musí mít začátek před koncem rezervace a zároveň nesmí být v minulém čase. Pokud datum validní není, nastaví se proměnná *\$odeslano* na hodnotu 2. V případě, že je vozidlo v daném termínu již zarezervované, tato proměnná se nastaví na hodnotu 3. Pokud je vše správně zkontrolováno, nastaví se na hodnotu 1. Pomocí této proměnné se dále zobrazí uživateli zpráva o úspěšném odeslání rezervace, nebo chybě, kvůli které nebyla rezervace odeslána.



```
1  if(($_POST["rezervace_zacatek"] > $_POST["rezervace_konec"]) ||
2      ($_POST["rezervace_zacatek"] < date('Y-m-d')) ||
3      ($_POST["rezervace_konec"] < date('Y-m-d')) {
4      $odeslano = 2;
5  }
```

Obrázek 10: Kód aplikace pro zjištění správného výběru termínu rezervace [1]

Jak lze zkontrolovat dostupnost daného vozidla? Pokud již existuje nějaká rezervace s tímto vozidlem, zkontroluje se, zdali je začátek zadané rezervace menší nebo rovno datumu konce již existující rezervace. Dále ještě zdali je konec zadané rezervace větší nebo rovno začátku již existující rezervace. Pomocí této podmínky se zajistilo to, že vozidlo nepůjde půjčit v jednom termínu vícekrát.



```
1  while (($rezervace = mysqli_fetch_array($seznamRezervaci)) != NULL) {
2      if(($_POST["rezervace_zacatek"] <= $rezervace["rezervace_konec"]) &&
3          ($_POST["rezervace_konec"] >= $rezervace["rezervace_zacatek"])) {
4          $kontrola = 0;
5          break;
6      }
7  }
```

Obrázek 11: Kód aplikace pro zjištění správného výběru termínu rezervace [2]

Pokud zadaná rezervace prošla přes všechny kontroly, uživateli se zobrazí informace o úspěšném odeslání a rezervace se uloží do databáze se statusem *čekající*.

```
1 if($deslano == 1) {
2     echo "<h2 class='font-bold text-center text-xl my-5 py-5 px-5 bg-green-500 text-white'>Rezervace byla úspěšně odeslána na potvrzení, děkujeme.</h2>";
3 }
4 else if($deslano == 2){
5     echo "<h2 class='font-bold text-center text-xl my-5 py-5 px-5 bg-red-500 text-white'>Zadejte, prosím, platné datum rezervace.</h2>";
6 }
7 else if($deslano == 3){
8     echo "<h2 class='font-bold text-center text-xl my-5 py-5 px-5 bg-red-500 text-white'>Vybraný termín již není dostupný.</h2>";
9     $sql = "SELECT * FROM rezervace";
10    $seznamRezervaci = mysqli_query($pripojeni, $sql);
11    echo "<div class='flex flex-col items-center pb-6'>";
12    echo "<h3 class='font-bold'>Vozidlo je již zarezerováno v níže uvedených termínech:</h3>";
13    while (($rezervace = mysqli_fetch_array($seznamRezervaci)) != NULL) {
14        if(($_POST["rezervace_zacatek"] <= $rezervace["rezervace_konec"]) && ($_POST["rezervace_konec"] >= $rezervace["rezervace_zacatek"])) {
15            echo "<p> . $rezervace["rezervace_zacatek"] . " - " . $rezervace["rezervace_konec"] . "</p>";
16        }
17    }
18    echo "</div>";
19 }
```

Obrázek 12: Kód aplikace pro zobrazení úspěšnosti zarezerování

Ještě před odesláním se ale musí vypočítat celková cena objednávky, aby měl klient přehled o tom, kolik ho bude rezervace stát. Pro toto je nejprve nutné vypočítat počet dnů, na které je vozidlo rezervováno.

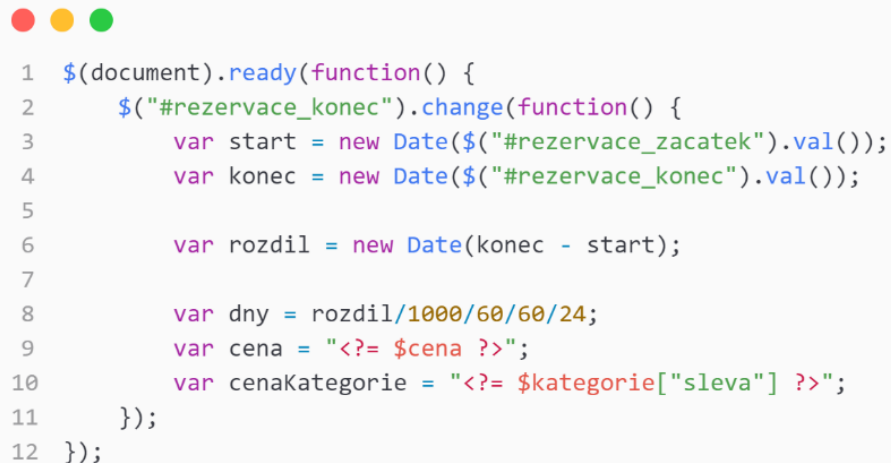
Tento počet se zjistí odečtením začátku rezervace od konce. Datum z databáze je ale jako string, takže je nutné použít funkci *strtotime()*, která nám převede textový řetězec na formát datumu. Toto datum ještě vydělíme počtem minut za den a zaokrouhlíme.

Tímto výpočtem získáme počet dnů mezi daty, pomocí kterého dále vypočteme celkovou cenu za rezervaci.

```
1 $rozdilDat = strtotime($_POST["rezervace_konec"]) - strtotime($_POST["rezervace_zacatek"]);
2 $rozdilDat = round($rozdilDat / (60*60*24));
```

Obrázek 13: Kód aplikace pro výpočet dnů v zadaném termínu

Aplikace byla ke konci vývoje doplněna také o aktuální zobrazení ceny ještě před tím, než klient rezervaci odešle. Bylo využito knihovny JQuery. Kód obsahuje funkci `$("#rezervace_konec").change()`, která se spustí pokaždé, když klient upraví konečné datum rezervace. Samotný výpočet počtu dnů je velice podobný jako v jazyce PHP.



```
1 $(document).ready(function() {
2     $("#rezervace_konec").change(function() {
3         var start = new Date($("#rezervace_zacatek").val());
4         var konec = new Date($("#rezervace_konec").val());
5
6         var rozdil = new Date(konec - start);
7
8         var dny = rozdil/1000/60/60/24;
9         var cena = "<?= $cena ?>";
10        var cenaKategorie = "<?= $kategorie["sleva"] ?>";
11    });
12 });
```

Obrázek 14: Kód aplikace s využitím knihovny JQuery na předběžné zobrazení ceny rezervace

## 2.4 Registrace a přihlašování

Registrační stránka se skládá ze základního formuláře a jeho prvků jako je *input type text*, *number* a *password*. Je zde i telefonní číslo s možností volby slovenské a české předvolby pomocí prvku *select a option*. Defaultní hodnota se může určit pomocí atributu *selected*. U telefonního čísla je i určitý vzor (*pattern*), který musí zadání splňovat. Parametr *[0-9]* nám určí, jaká čísla je zde možno zadat, *{9}* slouží k definování počtu čísel.

```
1 <div class="flex items-center px-4 border-2 shadow rounded-lg w-full">
2   <i class="ri-phone-fill text-xl mr-1"></i>
3   <select name="telcislopredvolba" id="telcislopredvolba">
4     <option value="+420" selected>(+420)</option>
5     <option value="+421">(+421)</option>
6   </select>
7   <input type="tel" name="telcislo" pattern="[0-9]{9}" placeholder="Telefonní číslo *"
8   class="w-full p-2 focus:outline-none" required>
9 </div>
```

Obrázek 15: Kód aplikace pro zadání telefonního čísla při registraci

Po kliknutí na tlačítko *Registrovat se*, se musí zkontrolovat hned několik věcí. Jako první je zde kontrola hesla.

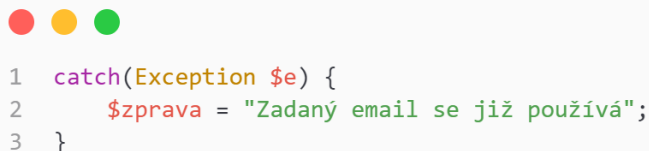
Obě hesla musí být totožná, a splňovat několik požadavků. Pomocí funkce *preg\_match()* se zkontroluje, zda heslo obsahuje námi určené znaky. Heslo musí obsahovat alespoň jedno číslo, velké a malé písmeno. Dále musí mít minimálně 6 znaků.

```
1 if($_POST["heslo"] == $_POST["heslo2"]) {
2     $cislo = preg_match('@[0-9]@', $_POST["heslo"]);
3     $pismeno = preg_match('@[a-z]@', $_POST["heslo"]);
4     $pismenoVelke = preg_match('@[A-Z]@', $_POST["heslo"]);
5 }
```

Obrázek 16: Kód aplikace pro kontrolu síly hesla

Pokud heslo všechny výše uvedené podmínky splňuje, vytvoří se jeho hash pomocí funkce `sha1()` který se následně s ostatními údaji vloží do databáze.

V databázi má každý uživatel email, který je jako unikátní klíč (může se nacházet v databázi pouze jednou). Je proto nutné použít klíčová slova `try{} a catch{}`  které nám odchytávají chyby. V případě zadání již používaného emailu se nám ukáže chyba a registrace se neprovede.



```
1 catch(Exception $e) {  
2     $zprava = "Zadaný email se již používá";  
3 }
```

Obrázek 17: Kód aplikace pro vyhození chyby při špatném zadání emailu

Další je přihlašování uživatelů. K tomu se využívá superglobální proměnná `$_SESSION`. Po použití funkce `session_start()` lze do této proměnné ukládat jakákoliv data o přihlášeném uživateli. Po úspěšném přihlášení se do proměnné uloží ID uživatele, jméno, email a role. Pomocí těchto prvků lze dále například autorizovat a odlišit běžného uživatele od administrátora.



```
1 if(sha1($_POST["heslo"]) == $radek["heslo"]) {  
2     $_SESSION["id"] = $radek["id_u"];  
3     $_SESSION["jmeno"] = $radek["jmeno"];  
4     $_SESSION["uzivatel"] = $radek["email"];  
5     $_SESSION["role"] = $radek["role"];  
6     header("Location: ../index.php");  
7     exit;  
8 }
```

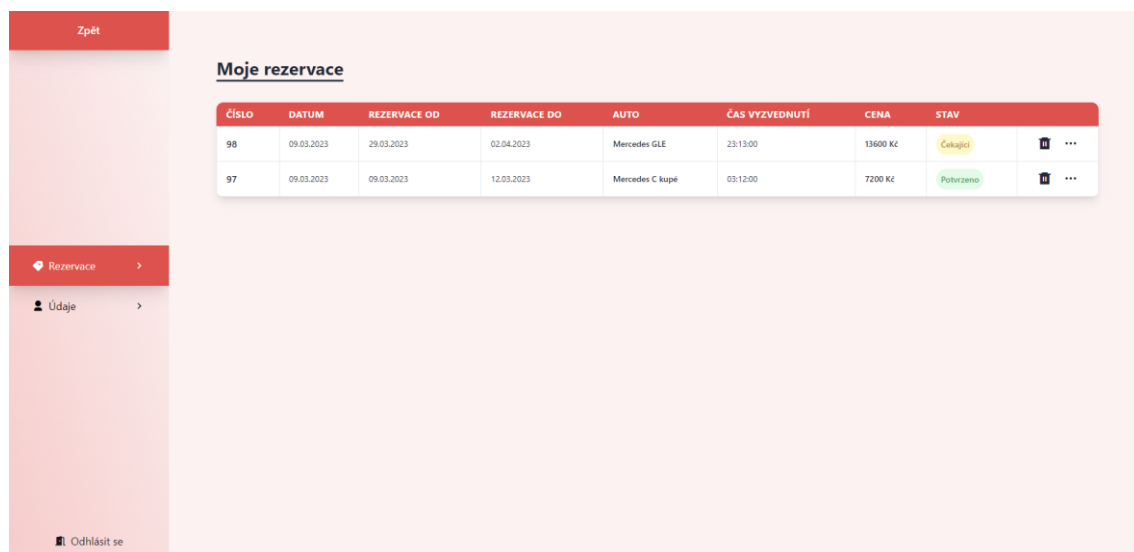
Obrázek 18: Kód aplikace pro uložení údajů o přihlášeném uživateli do `SESSION`

Při odhlášení se vymaže obsah této proměnné pomocí funkcí `session_unset()` a `session_destroy()`.



## 2.5 Uživatelské rozhraní

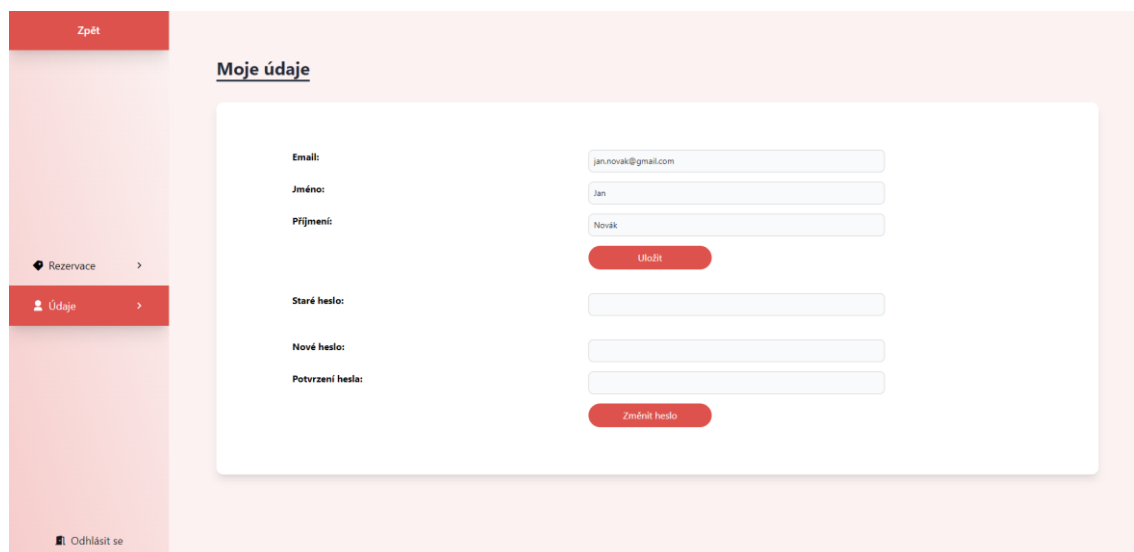
Po kliknutí na *Můj účet* v navigačním menu se zobrazí uživatelské rozhraní. Zde je možnost si zobrazit, upravit a smazat rezervace, popřípadě změna údajů.



Obrázek 19: Vzhled uživatelského rozhraní – Moje rezervace

Uživatel u rezervace může upravit pouze čas vyzvednutí a poznámku. Zbytek údajů k rezervaci je pevný – tohoto lze dosáhnout pomocí atributu *readonly*, který znemožní úpravu prvků ve formuláři. Pokud uživatel změní čas vyzvednutí nebo dopíše poznámku, stav rezervace se změní na čekající. Kromě úpravy objednávky je může také zcela smazat.

Jako další je zde změna údajů, kde lze v prvním formuláři změnit jméno a příjmení po kliknutí na tlačítko *Uložit*. Druhý formulář slouží na změnu hesla. Pro ni je nejprve nutné zadat své stávající (staré) heslo. Po kliknutí na tlačítko *Změnit heslo* se zkontroluje, zda zadané stávající heslo odpovídá. Pokud heslo souhlasí, stejně jako u registrace se u nového hesla zkontroluje bezpečnost.

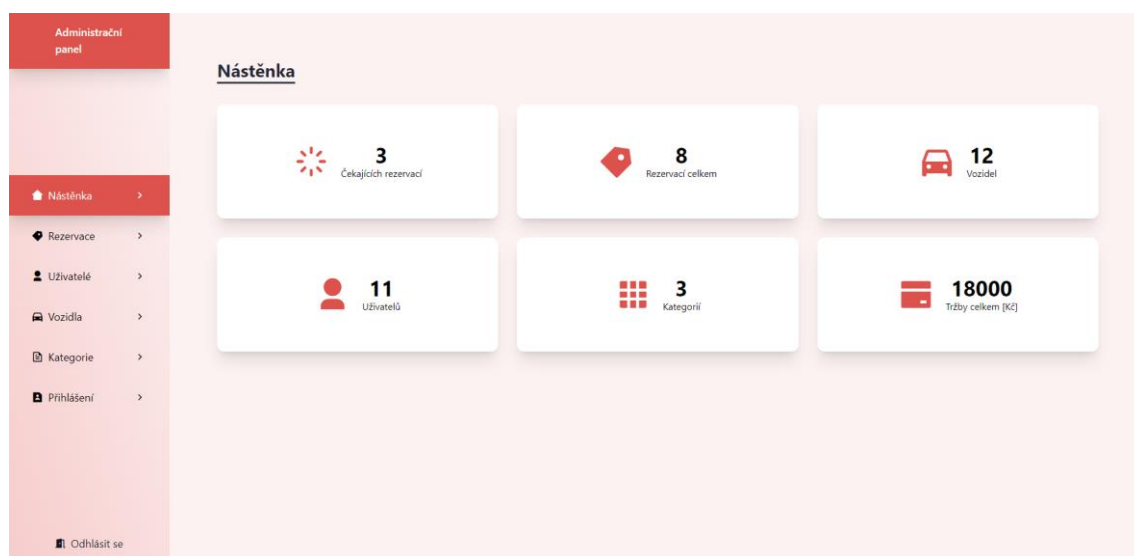


Obrázek 20: Vzhled uživatelského rozhraní – Údaje

## 2.6 Administrační rozhraní

Pokud se přihlásí uživatel, který má práva administrátora, místo tlačítka *Můj účet* se zobrazí tlačítko *Admin*. Po kliknutí na toto tlačítko se zobrazí úvodní nástěnka s různými statistikami, které rychle a přehledně zobrazí aktuální informace o půjčovně. Celá tato stránka se zobrazí pouze zdali je uživatel administrátor, pokud není, zobrazí se chybová hláška o neoprávněném přístupu. Obyčejný uživatel by se sice na stránku admina dostat neměl, nicméně pokud zná URL adresu a sám si ji změní, stránka se mu zobrazí. Na tuto kontrolu se využívá již výše uvedená superglobální proměnnou `$_SESSION`.

Administrační rozhraní slouží k úpravě, přidání a smazání vozidel, kategorií, uživatelů a rezervací. Nachází se zde i stránka pro změna hesla administrátora.



Obrázek 21: Vzhled administračního rozhraní - Nástěnka

Na stránce rezervace se zobrazí všechny uložené rezervace z databáze, včetně možností je smazat a upravit. Po kliknutí na určitého uživatele nebo vozidlo nás aplikace přesměruje na jejich detailní informace s možností úpravy.

Stavy rezervací jsou přehledně zobrazeny a zbarveny podle jejich aktuálního stavu. Toho bylo docíleno pomocí níže uvedeného kódu.

```
1 if($rezervace["stav"] == 1) {
2     echo '<span class="p-2 rounded-full font-semibold text-green-600 bg-green-100 text-sm">Potvrzeno</span>';
3 }
4 else if($rezervace["stav"] == 2) {
5     echo '<span class="p-2 rounded-full font-semibold text-yellow-600 bg-yellow-100 text-sm">Čekající</span>';
6 }
7 else if($rezervace["stav"] == 3) {
8     echo '<span class="p-2 rounded-full font-semibold text-gray-600 bg-gray-100 text-sm">Dokončeno</span>';
9 }
10 else {
11     echo '<span class="p-2 rounded-full font-semibold text-red-600 text-sm">Zrušeno</span>';
12 }
```

Obrázek 22: Kód aplikace pro zobrazení stavu rezervace včetně odpovídající barvy

U každé rezervace se dále nachází tlačítko na smazání a úpravu dané rezervace. Všechny tyto odkazy a tlačítka fungují na metodě GET, která nám umožní do URL adresy vložit námi zvolený parametr. Po kliknutí například na tlačítko upravit nás aplikace přesměruje na stránku, která nám podle parametru v URL adrese (id rezervace) umožní zobrazit a upravit detailní informace o každé rezervaci.

Velmi podobně nebo stejně jsou řešeny stránky o uživateli, vozidlech a kategoriích. Problém nastává v ten moment, kdy bychom chtěli smazat vozidlo, které je již obsaženo v nějaké rezervaci. Nebo smazat kategorii, která již obsahuje určitá vozidla. Na toto nám opět poslouží klíčová slova `try{} a catch{}.`



```
1 try {
2     mysqli_query($pripojeni, $sql);
3     header("Location: vozidla.php");
4 } catch (Exception $e) {
5     header("Location: vozidla.php?chyba=1");
6 }
```

Obrázek 23: Kód aplikace pro odchyčení chyb v případě mazání dat

Zdali nám query vyhodí chybovou hlášku, odkážeme admina zpět na stránku se seznamem. Tentokrát ale s parametrem chyba. Pokud tento parametr existuje, ukáže se administrátorovi alert box, který funguje pomocí JavaScriptu.



```
1 if(isset($_GET["chyba"])) {
2     echo '
3     <script>
4         alert("Chyba při smazání vozidla");
5     </script>';
6 }
```

Obrázek 24: Kód aplikace pro zobrazení alert boxu v případě chyby při mazání dat

## 2.7 Header, navigační menu a footer

Všechny tyto prvky jsou rozdělené v jednotlivých souborech, které se pomocí funkce *require\_once* vloží kdekoli je potřeba. Díky uložené hodnoty role v proměnné *\$\_SESSION* lze zjistit, zdali je přihlášený uživatel admin nebo nikoliv. Podle role se v navigačním menu zobrazí odkaz na *administrační panel* nebo *můj účet*.

```
1 if(isset($_SESSION["role"])) {
2     if($_SESSION["role"] == 1) {
3         echo '<li><a href="' . $cesta . 'admin/admin.php" class="font-medium tracking-wide
4             transition-colors duration-200 hover:text-red-500">Admin</a></li>';
5     }
6 }
```

Obrázek 25: Kód aplikace pro zobrazení menu podle oprávnění uživatele

Rozbalovací responzivní menu je řešené zvlášť, pomocí stylů se zobrazí pouze na malých obrazovkách. Toto menu je udělané pomocí Javascriptu a metody *addEventListener()*, která nám s pomocí parametru *click* zjistí, zdali uživatel klikl na určitý element na stránce. Pokud na menu klikl, přidá se nebo odebere styl *hidden*, který menu zviditelní a naopak. To stejné se provede u jednotlivých odkazů, po každém kliknutí na jakýkoliv odkaz se celé menu schová. Toto se provádí v cyklu *for-of()*.

```
1 const navigaceResponzivni = document.getElementById("navigace-responzivni");
2 const tlacitkoNavigace = document.getElementsByClassName("nav-tlacitko")[0];
3 const tlacitkoMobil = document.getElementsByClassName("nav-tlacitko")[1];
4
5 tlacitkoNavigace.addEventListener('click', () => {
6     navigaceResponzivni.classList.toggle('hidden');
7 })
8
9 tlacitkoMobil.addEventListener('click', () => {
10     navigaceResponzivni.classList.toggle('hidden');
11 })
12
13 const odkaz = document.getElementsByClassName('odkaz');
14
15 for (const element of odkaz) {
16     element.addEventListener('click', () => {
17         navigaceResponzivni.classList.toggle('hidden');
18     })
19 }
```

Obrázek 26: Kód aplikace pro rozbalovací hamburger menu

Navigační menu je na úvodní obrazovce průhledné, ale po přejetí o něco níže se zbarví pozadí do bílé barvy pro lepší viditelnost. Toto nám umožní opět Javascript. Kromě parametru *click* je zde i *scroll* který zjistí, o kolik bodů se uživatel po stránce posunul.



```
1 window.addEventListener('scroll', () => {
2     if(window.scrollY >= 50) {
3         nav.classList.add('bg-white');
4         nav.classList.add('shadow-md');
5     }
6     else{
7         nav.classList.remove('bg-white');
8         nav.classList.remove('shadow-md');
9     }
10 })
```

Obrázek 27: Kód aplikace pro zbarvení navigačního menu

### 3 Závěr

V maturitní práci jsem využil všechny získané zkušenosti z předešlých ročníků, hlavně z předmětů Webové aplikace a Databázových systémů. I přesto, že jsem většinu věcí na tuto práci již uměl, musel jsem řešit hned několik problémů, u kterých jsem získal mnoho nových znalostí.

Během postupu jsem hned několikrát musel kompletně změnit celou strukturu webové aplikace, hlavně z důvodu objevení nových způsobů a řešení, které práci ulehčily. Pokud bych tuto webovou aplikaci dělal znovu, už bych se vyvaroval chyb, které jsem na začátku a v průběhu práce dělal.

Pokud bych si měl vybrat věc, co webové aplikaci chybí, určitě to bude responzivita administračního a uživatelského rozhraní, ze které jsem nakonec upustil.

Nicméně pokud bych měl celkovou práci zhodnotit z mého pohledu, tak si myslím, že se povedla. Obsahuje vše, co bylo zadáno v zadání, plus i dalších pár věcí navíc. Očekávání bylo samozřejmě ze začátku o něco větší než finální produkt, ale i přesto jsem s prací nadmíru spokojený.

## 4 Seznam příloh

### 4.1 Obrázky

Obrázek 1: Základní styly pomocí Tailwind CSS .....	7
Obrázek 2: Základní styly pomocí Tailwind CSS - breakpointy.....	7
Obrázek 3: Vzhled úvodní stránky .....	8
Obrázek 4: Kód aplikace na získání počtu registrovaných uživatelů.....	8
Obrázek 5: Kód aplikace na zobrazení všech vozidel z databáze .....	9
Obrázek 6: Kód aplikace na zformulování slova u počtu míst.....	9
Obrázek 7: Vzhled filtrování kategorií u výběru aut na hlavní stránce .....	10
Obrázek 8: Kód aplikace pro zobrazení všech dostupných kategorií ve formuláři .....	10
Obrázek 9: Kód aplikace pro vypočítání ceny za určitý počet dnů .....	11
Obrázek 10: Kód aplikace pro zjištění správného výběru termínu rezervace [1].....	12
Obrázek 11: Kód aplikace pro zjištění správného výběru termínu rezervace [2].....	12
Obrázek 12: Kód aplikace pro zobrazení úspěšnosti zarezervování .....	13
Obrázek 13: Kód aplikace pro výpočet dnů v zadaném termínu.....	13
Obrázek 14: Kód aplikace s využitím knihovny JQuery na předběžné zobrazení ceny rezervace.....	14
Obrázek 15: Kód aplikace pro zadání telefonního čísla při registraci.....	15
Obrázek 16: Kód aplikace pro kontrolu síly hesla .....	15
Obrázek 17: Kód aplikace pro vyhození chyby při špatném zadání emailu .....	16
Obrázek 18: Kód aplikace pro uložení údajů o přihlášeném uživateli do SESSION.....	16
Obrázek 19: Vzhled uživatelského rozhraní – Moje rezervace.....	17
Obrázek 20: Vzhled uživatelského rozhraní – Údaje .....	17
Obrázek 21: Vzhled administračního rozhraní - Nástěnka.....	18
Obrázek 22: Kód aplikace pro zobrazení stavu rezervace včetně odpovídající barvy .....	18
Obrázek 23: Kód aplikace pro odchycení chyb v případě mazání dat.....	19
Obrázek 24: Kód aplikace pro zobrazení alert boxu v případě chyby při mazání dat.....	19
Obrázek 25: Kód aplikace pro zobrazení menu podle oprávnění uživatele.....	20
Obrázek 26: Kód aplikace pro rozbalovací hamburger menu .....	20
Obrázek 27: Kód aplikace pro zbarvení navigačního menu .....	21

## 5 Použité zdroje

Tailwind CSS - Rapidly build modern websites without ever leaving your HTML.. Tailwind CSS - Rapidly build modern websites without ever leaving your HTML. [online] [cit. 26.03.2023]. Dostupné z: <https://tailwindcss.com/>

PHP: Documentation. PHP: Hypertext Preprocessor [online]. Copyright © 2001 [cit. 26.03.2023]. Dostupné z: <https://www.php.net/docs.php>

HTML Tutorial. W3Schools Online Web Tutorials [online] [cit. 26.03.2023]. Dostupné z: <https://www.w3schools.com/html/default.asp>

JavaScript Tutorial. W3Schools Online Web Tutorials [online] [cit. 26.03.2023]. Dostupné z: <https://www.w3schools.com/js/default.asp>

CSS Tutorial. W3Schools Online Web Tutorials [online] [cit. 26.03.2023]. Dostupné z: <https://www.w3schools.com/css/default.asp>

JavaScript | MDN. [online]. Copyright ©1998 [cit. 26.03.2023] [cit. 26.03.2023]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

Tailblocks — Ready-to-use Tailwind CSS blocks. Tailblocks — Ready-to-use Tailwind CSS blocks [online] [cit. 26.03.2023]. Dostupné z: <https://tailblocks.cc/>

Osobní vozy Mercedes-Benz. Osobní vozy Mercedes-Benz [online]. Copyright © 2023. Mercedes [cit. 26.03.2023]. Dostupné z: <https://www.mercedes-benz.cz/>