

IT Akademija

Smjer: PHP Web Development

Projekat: **Web aplikacija za iznajmljivanje automobila**

Autor: Halima Potur

Datum: 10.09.2019.god.

UVOD

Tehnologije koje sam koristila u svojoj aplikaciji za frontend:

- Bootstrap(CSS, JS);
- JQuery JS;
- HTML;
- CSS;

Tehnologije koje sam koristila u svojoj aplikaciji za backend:

- PHP;
- MySQL;

Aplikacija Car Rental ima za cilj da olakša komunikaciju između poslodavca i kupca na način da komuniciraju online kako bi bio ostvaren brži kontakt.

Aplikacija ima dva korisnika ADMIN i USER.

Korištenje aplikacije kao User podrazumijeva registraciju jer samo registrovani korisnici mogu koristiti naše usluge i izvršiti rezervaciju željenog automobila.

Admin ima privilegije u smislu uređivanja, dodavanja, brisanja sadržaja aplikacije.

U narednim stranicama će bit detaljno uputstvo za korištenje aplikacije od strane korisnika, admina i developera te upute za razvoj aplikacije.

UPUTSTVO ZA KORISNIKE

Korisnik:

Prva stranica aplikacije ima kratak opis poslovanja firme, kontakt i adresu te kratke upute za korištenje. Prikazane su opcije login i register.

Aplikacija za iznajmljivanje automobila koristi se tako što korisnik treba da se prvo registruje. Nakon izvršene registracije koristi svoj password i username da bih se prijavio (login) na svoj račun.

Prijavom korisniku je prikazana lista dostupnih automobila. Klikom na jedan automobil izbacuje se prozor u kojem korisnik može izvršiti rezervaciju sa porukom za admina, odabir željenog datuma, vrijeme za koje rezerviš. Klikom na dugme reserved dobije poruku da je rezervacija uspješno izvršena i da će biti uskoro kontaktiran. Korisnik može obrisati svoju rezervaciju i pregledati sve svoje rezervacije.

Korisnik također može da uredi svoj račun, da ga obriše i pregleda.

Admin:

Aplikacija ima samo jednog admina. Admin može da izmijeni i vidi svoje podatke, ali ne može da obriše svoj račun.

Admin dodaje novi automobil, briše postojeći, mijenja podatke o automobilima. Ima na uvid sve registrovane korisnike.

Admin može da pregleda sve rezervacije od svih korisnika sa svim korisničkim podacima.

Opcija "logout" koristi se za izlazak iz aplikacije.

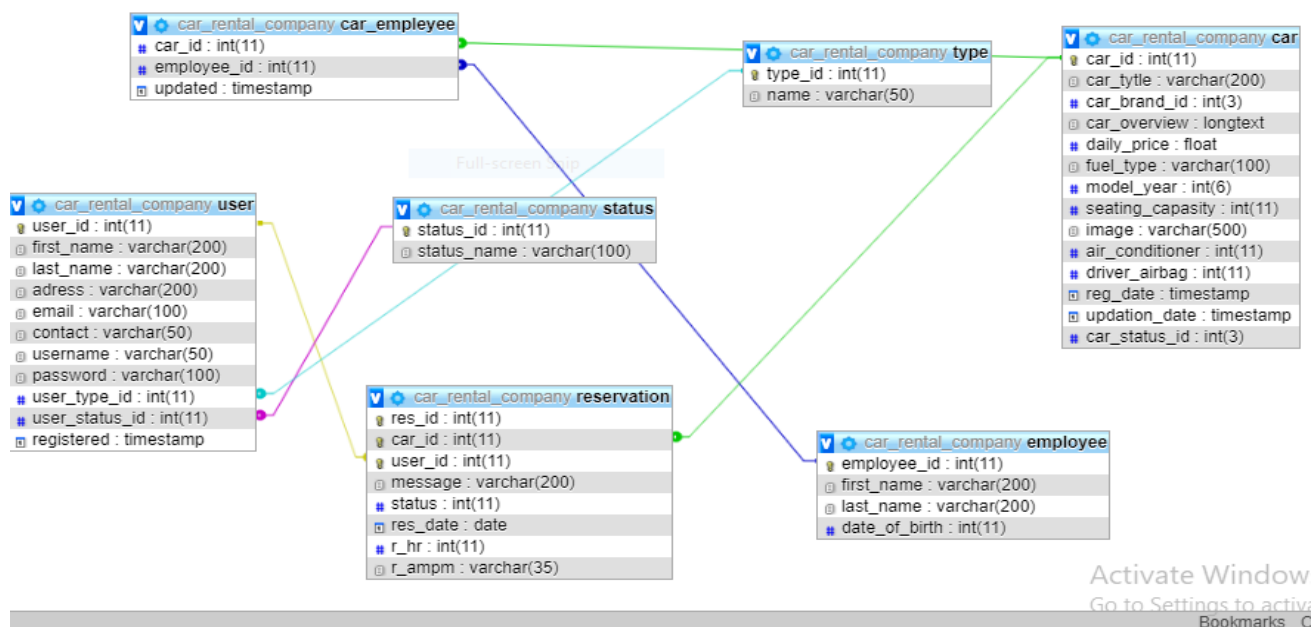
UPUTSTVO ZA DEVELOPERE

Aplikacija CarRental je izrađena u programskom jeziku PHP . Pošto je web aplikacija osim backend jezika, korištene su i frontend tehnologije HTML5, CSS3, Java Script i framework Bootstrap kako bi User Interface (UI) dobio neki standardni izgled kakav bi se očekivao od ovakve aplikacije. Što se tiče rada sa bazom korišten je SQL i MYSQL baza podataka.

Aplikaciju je najbolje pokrenuti u Chrome browseru. Sama aplikacija je slijedila model OOP odnosno objektno orijentisanog programiranja. A konekcija koja se koristila za bazu je PDO (PHP Data Object). Korištenje novijih pristupa jeziku omogućilo je pravilnije korištenje jezika i izbjegavanja kompliciranog i teško održivog koda.

Također da bi se izvršilo pokretanje aplikacije na lokalnom serveru potrebno je imati XAMPP, WAMP, LAMP ili MAMP zavisno od operativnog sistema(OS-a).

Daljni tok uputa bit će vezan prvenstveno za pojašnjenja o interakciji baze, backenda pisanog u php-u i naravno frontenda. Što se tiče baze broj tabela koji postoji je osam. Tabele su user, car, type, status, reservation, employee(nije u upotrebi), car_employee(nije u upotrebi), likes(služi za ocjenjivanje/rating-nije u upotrebi). Baza je potpuno relaciona jer strani ključevi omogućavaju pravilan prikaz kroz SQL upite.



Slika br.1

Klasa Database služi za komunikaciju sa bazom podataka. Sadrži sve funkcije koje su potrebne za rad sa bazom.

```
include_once('connection.php');  
class Database extends connection{  
public function __construct(){  
parent::__construct();
```

Ova klasa nasljeđuje Connection klasu koja služi za povezivanje sa bazom. . Za pokretanje roditeljskog konstruktora potreban je poziv parent :: __construct () unutar podređenog konstruktora. U ovom slučaju podređeni konstruktor je “class Database” Ako dijete ne definira konstruktor, tada ga može naslijediti od nadređene klase poput uobičajene metode klase (ako nije proglašeno privatnom).

Fajl config.php, u koji je uključena Database klasa (samim tim i connection), se uključuje u sve ostale fajlove kako bi bila ostvarena konekcija, mogli pozivati metode, kreirati nove objekte i pokrenuli sesije:

```
include_once(' ../config.php');  
$db = new Database();
```

U config.php je kod za pokretanje sesija:

```
if(session_status() == PHP_SESSION_NONE)  
{  
session_start();//pokreni sesiju ako već nije pokrenuta  
}//end if  
require_once('DatabaseClass/Database.class.php');
```

Registracija korisnika

Admin nalog je napravljen, može da se mijenja i ne može da se obriše. Da se registruje može samo obični korisnik, to je definisano u SQL upitu koji se nalazi u registration.php:

```
INSERT INTO user(first_name, last_name, adress, contact, username, password,  
user_type_id)  
VALUES(?,?,?,?,?, 1);
```

Kolonom “user_type_id” u tabeli user provjeravamo koji je tip korisnika admin ili user. Admin je “user_type_id=2” dok je user “user_type_id=1”. Tabela user je stranim ključem povezana na tabelu type .

```
ADD CONSTRAINT `user_ibfk_2` FOREIGN KEY (`user_type_id`) REFERENCES  
`type` (`type_id`);
```

Prijava korisnika

Korisnik se loguje koristeći svoje korisničko ime(username) i password. Ako je korisnik sa sesijom userID odnosno user_type_id=1 šalje ga na User/index.php, a ako je sesija logged odnosno user_type_id=2 onda je admin i šalje se na Admin/index.php.

```
if($result){  
$r = $result['user_type_id'];  
if($r == '2'){  
$_SESSION['logged'] = '2';  
header('location: Admin/index.php');  
}else{  
$_SESSION['logged'] = '1';  
$_SESSION['userID'] = $result['user_id'];  
header('location: User/index.php');
```

U svaki fajl je uključen i logout.php koji služi za izlaz iz aplikacije i šalje na početnu stranicu index.php. U ovom fajlu se nalazi poništavanje sesije:

```
include_once('config.php');
```

```
session_destroy();
```

```
header("location: ../First_Project_Car_Rental/index.php");
```

Aplikacija uglavnom sadrži osnovne CRUD (create-read-update-delete) operacije gdje se u User folderu nalaze funkcije gdje korisnik može da rezerviše svoj automobil(Create), može da mijenja svoj profil(Update), može da vidi svoj profil(Read). Također korisnik ima opciju da obriše>Delete) svoj račun, ali aplikacija je urađena tako da se iz baze ne može ništa obrisati. To sam definisala u SQL upitu:

```
$sql = "UPDATE user SET user_status_id = 2 WHERE user_id = ?";
```

Tabela status u bazi podataka predstavlja 4 statusa korisnika:

- Active (Aktivan);
- Deleted (Obrisan);
- Blocked (Blokiran);
- Forbidden (Zabranjen)

Tako da kada User obriše svoj račun samo ga postavi na status “2” a ne trajno obriše. Nakon toga se korisnik više ne može prijaviti ni koristiti aplikaciju izuzev ponovne registracije.

SQL upit u login.php gdje se vidi da se mogu logovati samo “Aktive” korisnici odnosno u tabeli user kolona user_status_id = 1.

```
$sql = 'SELECT * FROM user WHERE username = ? AND password = ? AND user_status_id=1';
```

Tabela “user” kolona user_status_id je referencirana na tabelu status odnosno kolonu status_id:

```
ADD CONSTRAINT `user_ibfk_1` FOREIGN KEY (`user_status_id`) REFERENCES `status` (`status_id`).
```

U “Admin/carUpdate.php” se nalazi upload slika. Kada novu sliku postavljamo, stara se briše i dodaje nova u folder “Uploads”:

```
$new_image_name = 'image_' . date('Y-m-d-H-i-s') . '_' . uniqid() . '  
    move_uploaded_file($_FILES["car_image"]["tmp_name"],  
    "../car_image/".$new_image_name);  
    $new_image_name = '../car_image/'.$new_image_name;
```

U primjeru iznad nova slika dobiva naziv image_trenutniDatum (PHP funkcija date koja vraća trenutni datum i vrijeme) i jedinstveni id broj (image_2019-09-12-11-56-20_15).

move_uploaded_file() - premješta prenesenu datoteku na novo mjesto.

Ova funkcija provjerava da li je datoteka koju filename je označio valjana datoteka za prijenos (što znači da je bila prenesena putem PHP-ovog mehanizma za prijenos HTTP POST). Ako je datoteka valjana, bit će premještena u ime datoteke koja je definisana.

Cijena automobila postavljena je if-else uslovom ako je automobil sa brojem sjedišta 4 onda je cijane 50,00 KM, ako je sa brojem sjedišta 9 cijena na dnevnom nivou je 100,00 KM ili je cijena 70 KM.

```
$carPrice = null;  
if($carCapacity == '4 Persons'){  
$carPrice = 50;  
}else if($carCapacity == '9 Persons'){  
$carPrice = 100;  
}else{  
$carPrice = 70;
```

Koristeći \$.ajax(jQuery) metod, koji znači učitavanje podataka sa servera koristeći HTTP POST zahtjev, sam uradila rezervaciju automobila od strane korisnika. U fajlu reservedProcess je postavljen PHP kod pomoću kojeg uspostavljam komunikaciju sa bazom i sql upit:

```
$sql = "SELECT COUNT(res_date) as rdt FROM reservation WHERE car_id = ? AND  
res_date = ?";
```


Zatim u User/index.php:

```
var message= $('#message'+str).val();  
var car_id = str;  
var user_id = '<?php echo $_SESSION['userID']; ?>';  
var message = $('#message'+str).val();  
var res_date = $('#res_date'+str).val();  
var hr = $('#r_hr').val();  
var ampm = $('#r_ampm').val();  
var datas =  
  
"car_id="+car_id+"&user_id="+user_id+"&message="+message+"&res_date="+res_  
date+"&r_hr="+hr+"&r_ampm="+ampm;
```

Ovo je skraćena funkcija Ajax koja je ekvivalentna:

```
$.ajax({  
    type: "POST",  
    url: "reservedProcess.php",  
    data: datas  
}).done(function( data ) {  
    $('#info').html(data);  
});
```

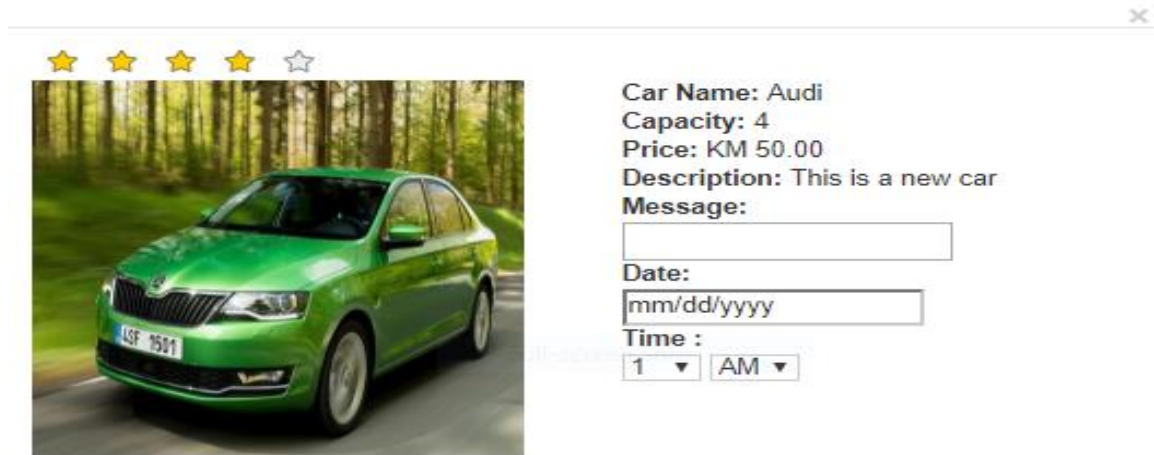
Ovdje dobivamo traženi HTML isječak i ubacujemo ga na stranicu.

SMJERNICE ZA BUDUĆI TOK RAZVOJA APLIKACIJE

Svaka aplikacija ima prostora za poboljšanje, tako i ova. Mnoge stvari nedostaju i to smatram vlastitim izazovom gdje ću raditi na poboljšanju. Aplikaciji nedostaje verifikacija email-a, zaboravljena lozinka i promjena lozinke što se može riješiti upotrebom nekih php email biblioteka da bi se ostvarila komunikacija potrebna za reset opcija.

Zatim u bazi su napravljene i povezane tabele “employee, car_employee, likes” koje nisu iskorištene još uvijek, a na čemu ću raditi u budućnosti. Tabela “employee” koja je povezana na “car_employee”(veza između zaposlenika i automobila) bit će korištena da bi se zaposlenicima dodijelili automobili za koje će biti odgovorni (servisi, registracije, održavanje uglavnom).

Tabela likes koristit će za ocjenjivanje automobila i kompletne aplikacije. HTML i CSS sam uradila za “star rating”(ocjenjivanje) još nedostaje PHP kod.



The screenshot displays a web form for car rental. At the top, there is a star rating system with five stars; the first four are yellow and the fifth is grey. Below the stars is a photograph of a green Skoda car parked on a road in a forest. To the right of the image, the following information is displayed: Car Name: Audi, Capacity: 4, Price: KM 50.00, Description: This is a new car. Below this is a text input field labeled 'Message:'. Further down is a 'Date:' label followed by a date input field with the placeholder 'mm/dd/yyyy'. At the bottom, there is a 'Time:' label followed by a dropdown menu showing '1' and another dropdown menu showing 'AM'.

Slika broj 2 prikaz zvjezdica za ocjenjivanje od strane korisnika.

Ocjene se trebaju prikazati adminu u tabeli sa prikazanim automobilima i na početnoj stranici aplikacije First_Project_Car_Rental/index.php.

Ovo su bile neke osnovne smjernice i moje trenutne ideje, dok se nadam da će se u budućem vremenu i tokom rada javljati još bolje ideje i razvijati moj osjećaj za programiranjem.