

Web sustav za zadavanje i prikupljanje ponuda za putovanja

Dubovečak, Denis

Undergraduate thesis / Završni rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:346028>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-02-23**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA**

Stručni studij, smjer Informatika

**WEB SUSTAV ZA ZADAVANJE I PRIKUPLJANJE
PONUDA ZA PUTOVANJA**

Završni rad

Denis Dubovečak

Osijek, 2016.

Sadržaj

1. UVOD	1
1.1. Zadatak završnog rada.....	1
2. PRIPREMA.....	2
2.1. Tehnologije.....	2
2.2. Baza podataka	3
2.3. Struktura aplikacije	5
2.4. Predlošci	7
3. FUNKCIONALNOST	9
3.1. Naslovnica.....	9
3.2. Odjava	11
3.3. Registracija.....	11
3.4. Nove lokacije.....	11
3.5. Oglas.....	11
3.6. Nova ponuda	13
3.7. Pregled i prihvaćanje ponuda	13
3.8. Status ponuda	16
4. DIZAJN.....	17
5. ZAKLJUČAK	24
LITERATURA.....	25
SAŽETAK.....	26
ŽIVOTOPIS	28

1. UVOD

Oglašavanje i prikupljanje ponuda za putovanja je već generacijama vrlo bitan posao, bilo da se radi o školi koja želi organizirati izlet za svoje učenike, bilo da se radi o turističkim agencijama koje planiraju privući pažnju turista bogatim programom ili ako se možda radi i o samim autoprijevoznicima koji žele jednostavan način za pregledavanje mogućih natječaja i odabira onih na koje bi se mogli javiti.

Kod izrada takvih sustava, najvažniju ulogu ima sama funkcionalnost aplikacije. Njezin temelj u bazi podataka i naposljetku dizajn korisničkog sučelja. Ima i manje očitih stvari koje osobi kojoj se dopadne raditi ovakav posao izuzetno puno znače. To su prije svega tehnologije koje su mu dane na izbor.

Tako je zadatak ovog završnog rada napraviti web sustav za zadavanje i prikupljanje ponuda za putovanja. Rad opisuje cijeli proces izrade aplikacije u tu svrhu, od postavljanja radnog okruženja i odabira tehnologija, preko opisa tablica u bazi i izrade predložaka, sve do same funkcionalnosti programa i dizajna njenog korisničkog sučelja. Na kraju razrade je i poglavlje posvećeno testiranju aplikacije kao dokaz u njezinu funkcionalnost.

1.1. Zadatak završnog rada

Objasniti način raspisivanja (zadavanja) uvjeta ponuda za putovanja kao i način prikupljanja istih. Dizajnirati bazu podataka pomoću koje će se moći voditi sustav zadavanja i prikupljanja ponuda. Predvidjeti najmanje dva korisnička profila: tražitelj usluge i ponuditelj. Izraditi web sučelje prema korisnicima kao i funkcionalnosti za pristup bazi podataka. Objasniti način rada cijele aplikacije kao i način testiranja funkcionalnosti.

2. PRIPREMA

Prvi korak u izradi bilo kojeg projekta, pa tako i projektiranja ovakve web aplikacije, jest stvaranje radnog okruženja i odabir tehnologija za rad. Tako ovo poglavlje služi kao svojevrsna platforma na kojoj će se obrađivana web aplikacija graditi.

Prvo predstavljamo i opisujemo tehnologije koje su korištene u izradi, pa krećemo na proces izrade i opis same baze podataka koju aplikacija koristi da bi dolazila do podataka. Nakon toga je objašnjena struktura aplikacije, odnosno njezina fizička logika – koja se zbog praktičnosti treba podijeliti na više dijelova. Na poslijetku dolazi izrada predložaka za statične blokove koda kako bi se smanjila njegova fragmentacija.

2.1. Tehnologije

Korištene tehnologije mogu se podijeliti na dvije grane: programske i pomoćne. Programske tehnologije su one koje se koriste kao materijal za izradu aplikacije, a to su:

- MySQL – baza podataka koja se koristi za spremanje podataka
- PHP 5.6.15 – serverski jezik u kojem je napisana kompletna logika aplikacije i koja služi kao posrednik između same aplikacije baze podataka
- Javascript – klijentski jezik, koji se u ovom slučaju koristi isključivo za standardiziranje korisničkog sučelja
- HTML5 – jezik kojim definiramo kostur aplikacije
- CSS3 – koji u kombinaciji sa Javascript-om, služi za izradu dizajna korisničkog sučelja
- Bootstrap/jQuery – fokus ovog rada nije na dizajnu pa se koristi najpopularniji okvir za dizajniranje standardiziranih web stranica

Pomoćne tehnologije, s druge strane, su analogizirane kao alat koji se koristi za rad s materijalom (programskim tehnologijama). To su:

- XAMPP – kako se dinamičke aplikacije (ova radi na PHP-u) testiraju jedino na serveru, ovaj program se koristi za emuliranje serverskog okruženja lokalno.
- Google Chrome - HTML , CSS i Javascript nemaju navedena ograničenja, ali da se pristupi aplikaciji potreban je pretraživač.
- Sublime Text 3 - veoma brz i fleksibilan tekst editor koji se koristi za pisanje koda.

2.2. Baza podataka

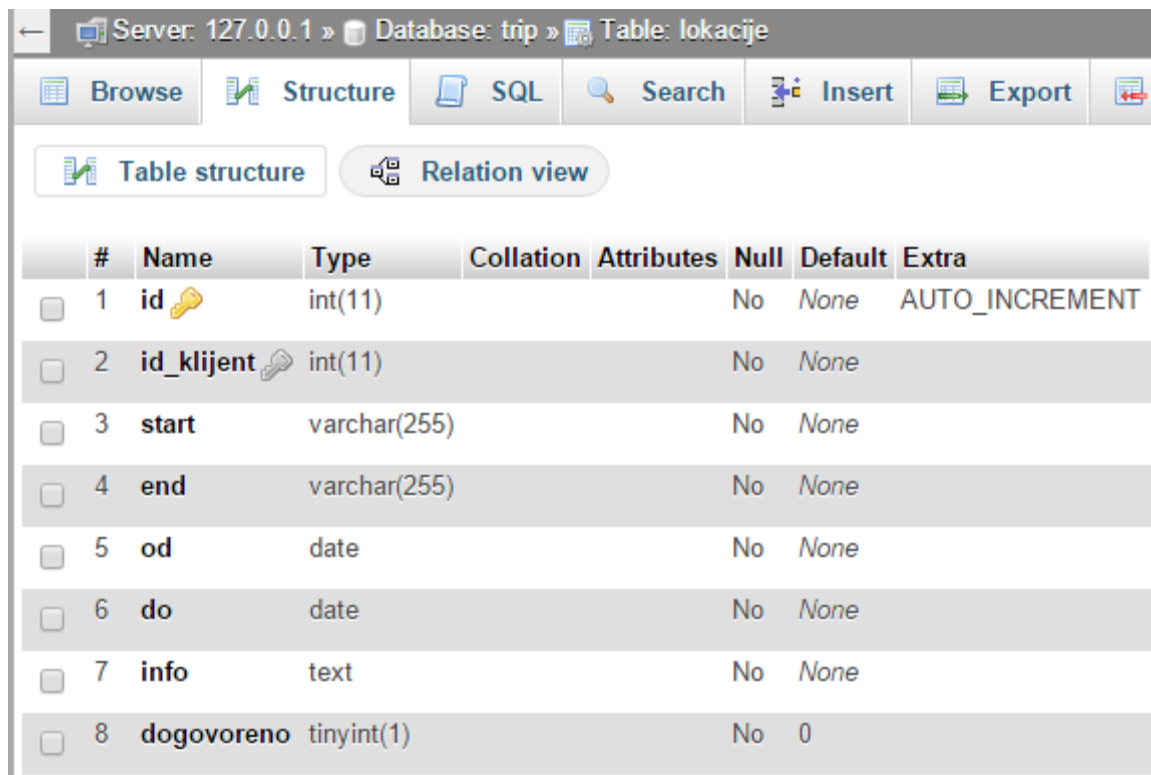
Kompletna baza podataka se izrađuje putem *phpmyadmin* sučelja koji je sastavni dio XAMPP-a. Za potrebe ovoga rada nije nužno raditi kompleksnu bazu podataka jer sadrži svega 3 tablice. Prva na tom popisu je tablica `users` (Sl. 2.1.) koja sadrži pristupne podatke i informacije o registriranim korisnicima.

#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	1	id	int(11)		No	None	AUTO_INCREMENT
<input type="checkbox"/>	2	user	varchar(25)		No	None	
<input type="checkbox"/>	3	pass	varchar(64)		No	None	
<input type="checkbox"/>	4	role	enum('0', '1')		No	1	
<input type="checkbox"/>	5	naziv	varchar(255)		No	None	

Sl. 2.1. Struktura tablice `users`

Tablica neće postati komplicirana za korištenje, jer sadrži samo najnužnije informacije, a to su: `id` (identifikacijska oznaka) svakog korisnika, `user` i `pass` koji služe kao parametri za prijavu korisnika u sustav. Atribut `role` označava vrstu korisnika i `naziv` u koji se sprema službeno ime te registrirane pravne osobe.

Iduća tablica je tablica `lokacije` (Sl. 2.2.) koja je kompleksnija i sadrži podatke o objavljenim oglasima za koje se traži prijevoznik.



#	Name	Type	Collation	Attributes	Null	Default	Extra
1	id	int(11)			No	None	AUTO_INCREMENT
2	id_klijent	int(11)			No	None	
3	start	varchar(255)			No	None	
4	end	varchar(255)			No	None	
5	od	date			No	None	
6	do	date			No	None	
7	info	text			No	None	
8	dogovoreno	tinyint(1)			No	0	

Sl. 2.2. Struktura tablice `lokacije`

Od podataka koji se nalaze u ovoj tablici su dvije identifikacijske oznake – prvi je indeks korisnika koji je objavio oglas, a drugi je identifikator same lokacije. Atributi `start`, `end`, `od` i `do` su glavni parametri oglasa (ruta i planirano vrijeme), dok je `info` tekst polje u kojem se nalaze sve ostale informacije i zahtjevi. Ovdje postoji i statusno polje `dogovoreno` koje označava je li prijevoz za određenu lokaciju ugovoren.

Zadnja tablica je tablica `ponude` (Sl. 2.3.). Točnije, za svaku lokaciju za koju korisnik objavi oglas, autoprijevoznik može poslati svoju ponudu koja dolazi na uvid potražitelju te na temelju usporedbe svih ponuda odlučuje kojeg prijevoznika će angažirati.

#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	1 id	int(11)			No	None	AUTO_INCREMENT
<input type="checkbox"/>	2 id_ponudac	int(11)			No	None	
<input type="checkbox"/>	3 cijena	float			No	None	
<input type="checkbox"/>	4 opis	mediumtext			No	None	
<input type="checkbox"/>	5 id_lok	int(11)			No	None	
<input type="checkbox"/>	6 ok	enum('0', '1', '2')			No	0	

Sl. 2.3. Struktura tablice `ponude`

Osim identifikatora prijevoznika ponuda također sadrži: očekivanu cijenu, detalje ponude i id lokacije, odnosno oglasa, za koji je ponuda namijenjena. Polje `ok` je statusna varijabla koja ovisno o svojoj vrijednosti određuje da li je ponuda na razmatranju, prihvaćena ili odbijena.

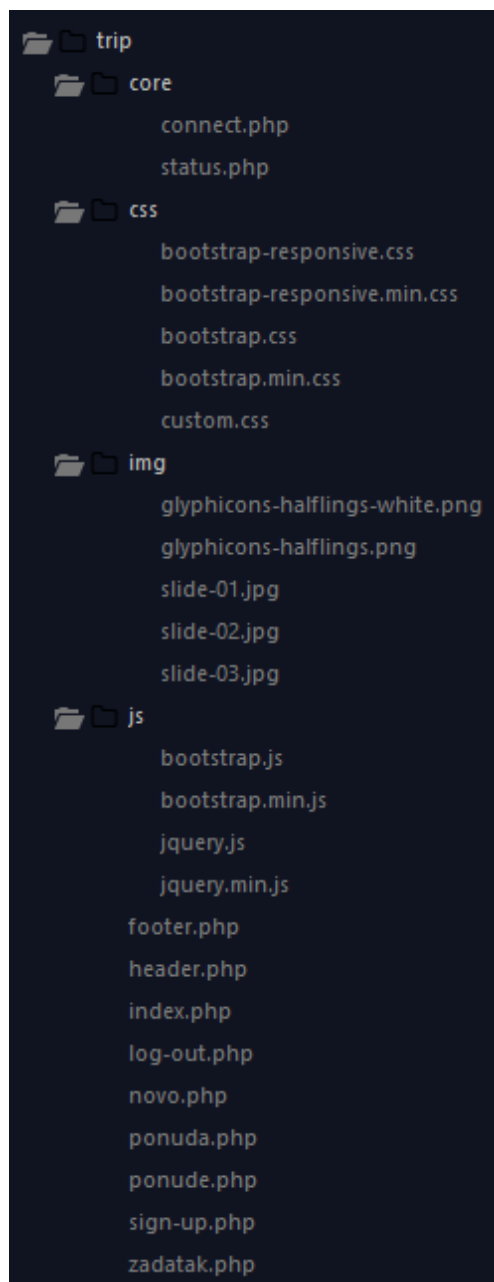
2.3. Struktura aplikacije

Nakon izrade baze podataka na redu je definiranje strukture same aplikacije. Struktura aplikacije je zapravo raspoređivanje dijelova aplikacije u mape i logičke odjeljke kako bi se u slučaju izmjenjivanja lakše locirali. Struktura ove aplikacije izgleda:

- /
 - Root mapa aplikacije, gdje su smještene sve PHP stranice i predlošci, kao i ostali folderi.
- /core
 - Skripte potrebne za pokretanje aplikacije.
- /js
 - Bootstrap i jQuery biblioteke.

- /css
 - Datoteka sa stilovima specifičnima za aplikaciju, te Bootstrap CSS.
- /img
 - Slike i simboli koji su korištene u aplikaciji (Glyphs, Slideshow)

Sl. 2.4. prikazuje završni izgled strukture web aplikacije:



Sl. 2.4. Struktura aplikacije

2.4. Predlošci

Općenito je dobra ideja da se prije pisanja funkcionalnosti naprave svojevrсни predlošci, odnosno da se izdvoje blokovi koda koji će se pojavljivati na svakoj stranici da se izbjegne nakupljanje. Kako je ovo prejednostavna aplikacija da bi se opravdalo pisanje objektno orijentiranog okvira ili pratilo smjernice MVC paradigme, ovdje će se koristiti samo izdvojeni blokovi koda koji će kasnije biti ubačeni na odgovarajuća mjesta na stranici.

Prvi na redu su dizajnerski predlošci, odnosno zaglavlje (Sl. 2.5.) i podnožje stranice koje će biti iste bez obzira na kojem dijelu aplikacije se našli. Njih pišemo samo u HTML-u te ih spremamo kao .php datoteku i kasnije se pozivaju u aplikaciji naredbom `include('ime_datoteke.php');`

Sl. 2.5. header.php

Ovdje se uz HTML koristi i komadić PHP koda (svjetlija pozadina). Njegova uloga u ovom slučaju je da kondicionalno (u ovisnosti o statusu korisnika – prijavljen da/ne) određuje koji linkovi će postojati na zaglavlju stranice.

Connect.php (Sl. 2.6.) sadrži podatke za prijavu u bazu podataka tj. to je datoteka koja je potrebna za funkcioniranje dinamičke web stranice, te status .php (Sl. 2.7.) koja se koristi za provjeru statusa prijavljenosti korisnika i postavljanje korisničke sesije.

```
connect.php
1 <?php
2 //podaci za spajanje sa bazom
3 $hostname='localhost';
4 $username='root';
5 $password='';
6 $database='trip';
7 //spajamo se
8 $db = mysqli_connect($hostname, $username, $password, $database);
9 //provjeravamo da li radi
10 if (mysqli_connect_errno()) {
11     echo mysqli_connect_error();
12     exit();
13 } else {
14     // echo "RADIIIIII :D";
15 }
16 ?>
```

Sl. 2.6. connect.php

Prve dvije naredbe u skripti postavljaju sesiju i uspostavljaju konekciju s bazom, naredne tri su inicijalizacije varijabli. Funkcija `provjeri()` služi kao validator korisničkog stanja, a skripta ispod provjerava postoji li aktivna sesija te iz nje uzima varijable i provjerava ih naredbom `preg_replace()`, koja općenito briše sve ne-alfanumeričke znakove, te poziva funkciju `provjeri()` s njima kao parametrima.

```
status.php
1 <?php
2 session_start();
3 require_once("connect.php");
4
5 $user_ok = false;
6 $log_username = "";
7 $log_password = "";
8
9 function provjeri($db, $u, $p) {
10     $sql = "SELECT * FROM users WHERE user='$u' AND pass='$p' LIMIT 1";
11     $query = mysqli_query($db, $sql);
12     $numrows = mysqli_num_rows($query);
13     if ($numrows > 0) {
14         return true;
15     }
16 }
17
18 if (isset($_SESSION["user"]) && isset($_SESSION["pass"])) {
19
20     $log_username = preg_replace('#[^a-z0-9]#i', '', $_SESSION['user']);
21     $log_password = preg_replace('#[^a-z0-9]#i', '', $_SESSION['pass']);
22
23     $user_ok = provjeri($db, $log_username, $log_password);
24 }
25
26 ?>
```

Sl. 2.7. status.php

3. FUNKCIONALNOST

Nakon pripreme, počinje slaganje funkcionalnosti aplikacije. Ovo poglavlje će biti fokusirano na PHP blokove u svim stranicama koje se nalaze na 'root' folderu aplikacije.

Treba naglasiti da svaka naredna stranica treba započinjati s naredbom `include_once('core/status.php');` kako bi se inicirala sesija te uspostavila konekcija s bazom bez koje je ova aplikacija bezvrijedna.

3.1. Naslovnica

Naslovnica je početna stranica aplikacije i jedina kojoj neprijavljeni korisnici mogu pristupiti (izuzev one koja bas i služi za registraciju).

Sastoji se isključivo od toga da pri učitavanju provjerava status prijavljenosti (i klase) korisnika pa u odnosu na to, prikazuje određeni sadržaj ili linkove (Sl. 3.1.). Provjera prijavljenosti se obrađuje preko prethodno opisane funkcije (`provjeri()`) te u slučaju da je posjetitelj prijavljen, iz baze se uzimaju podaci o korisniku koji potom određuju vrstu sadržaja ovisno o ulozi korisnika, te se korisniku nudi mogućnost odjave sa stranice.

U slučaju da korisnik nije prijavljen, jedine opcije koje stranica nudi je pregledavanje popisa aktualnih oglasa, te mogućnosti prijave odnosno registracije.

Prijava (Sl. 3.2.) se također nalazi u ovoj datoteci. Ona provjerava je li ispunjena forma za prijavu putem `POST` metode, te ukoliko je, uzima upisane podatke i traži postoji li u tablici ``korisnici`` jedinstveni unos koji se slaže s onim što je poslano preko forme. Ako postoji, skripta postavlja varijable sesije, a ako ne postoji a forma je poslana, onda piše poruku upozorenja.

```

35
36 /*****
37 ***** RENDER - ostali
38 *****/
39 if(!isset($_SESSION['role'])){
40     $rend="Trenutno nema potraživanja.";
41     $i = 1;
42     $op=mysqli_query($db, "SELECT * FROM lokacije WHERE dogovoreno=0");
43     if($ct=mysqli_num_rows($op)){
44         $rend='
45             <thead>
46             <tr>
47                 <th>#</th>
48                 <th>RUTA</th>
49                 <th>INTERVAL</th>
50             </tr>
51             </thead>
52         ';
53     }
54     while($row=mysqli_fetch_array($op, MYSQLI_ASSOC)){
55         $id=$row['id'];
56         $start=$row['start'];
57         $end=$row['end'];
58         $od=$row['od'];
59         $do=$row['do'];
60
61         $rend .= '
62             <tr>
63                 <td>' . $i . '</td>
64                 <td>' . $start . ' <span class="glyphicon glyphicon-resize-horizontal"></span>
65                 <b style="color:red;">' . $end . '</b> .
66                 |</td>
67                 <td>' . $od . ' <span class="glyphicon glyphicon-minus"></span> ' . $do . '
68                 </td>
69             </tr>
70         ';
71         $i++;
72     }
73 }
74

```

Sl. 3.1. Logika za prikaz – neprijavljeni korisnici

```

3
4 /*****
5 ***** LOGIN
6 *****/
7 if (isset($_POST["user"])) {
8
9     $username = mysqli_real_escape_string($db, $_POST['user']);
10    $password = mysqli_real_escape_string($db, $_POST['pass']);
11
12    $sql = "SELECT * FROM users WHERE user='$username' AND pass='$password' LIMIT 1";
13    $query = mysqli_query($db, $sql);
14    $row = mysqli_fetch_row($query);
15
16    $id = $row[0];
17    $user = $row[1];
18    $pass = $row[2];
19    $uloga = $row[3];
20
21    if ($password != $pass || $username != $user) {
22        echo 'Greska: korisnik ne postoji! ';
23        echo '<a href="login.php">Vratite se natrag.</a>';
24        exit();
25    } else {
26        $_SESSION['id'] = $id;
27        $_SESSION['user'] = $user;
28        $_SESSION['pass'] = $pass;
29        $_SESSION['role'] = $uloga;
30    }
31
32    header("location: index.php");
33    exit();
34 }
35

```

Sl. 3.2. Logika za prijavu

3.2. Odjava

Odjava korisnika je vrlo jednostavna i njen kod se sastoji od par linija koje pokreću sesiju (`session_start();`), inicijaliziraju je na prazan niz (`$_SESSION = array();`), te ju brišu (`session_destroy();`) i vraćaju prethodnu stranicu (`header("location: index.php");`).

3.3. Registracija

Ukoliko korisnik koji nema račun poželi uživati u punim blagodatima aplikacije mora se registrirati kao član ove stranice. `Sign-up.php` je datoteka u kojoj se nalazi HTML forma i jednostavna skripta za registraciju.

Skripta pri učitavanju provjerava je li forma ispunjena, ako je, onda preuzima podatke putem POST metode. Zatim ih putem SQL `INSERT` naredbe unosi u bazu te postavlja sesiju na isti način kao i sistem za prijavu (Sl. 3.3.).

Tada je novo-registrirani korisnik preusmjeren na naslovnu stranicu s već potvrđenim statusom prijave koja mu ovisno o klasi predstavlja dodatne opcije.

3.4. Nove lokacije

Potraživači u lijevom vertikalnom izborniku imaju opciju izraditi svoje oglase. Funkcionalnost ovoga je gotovo identičan onoj za registraciju, samo se podaci iz forme unose u tablicu ``lokacije`` a ne u ``users``. Također, nakon kreiranja oglasa korisnik je preusmjeren na naslovnu stranicu gdje može vidjeti svoj oglas na listi aktualnih.

3.5. Oglas

Klikom na neki oglas na naslovnici, korisnik se preusmjerava na stranicu s detaljima tog oglasa. Ovisno o svojoj klasi dostupne su mu različite opcije. Ono što je identično za svakog korisnika su detalji tog oglasa odnosno uvjeti koje kreator oglasa postavlja mogućim zainteresiranim strankama. Također se nalazi skripta koja koristi podatke iz baze (parametre rute) da bi generirala interaktivnu kartu (Sl. 3.4.) kako bi zainteresirani imali bolji uvid u zahtjeve potražitelja.

```

36 $detalji_lokacije=mysqli_query($db, "SELECT * FROM lokacije WHERE id='$id_lok' LIMIT 1");
37 $row=mysqli_fetch_array($detalji_lokacije, MYSQLI_ASSOC);
38     $id=$row['id'];
39     $start=$row['start'];
40     $end=$row['end'];
41     $od=$row['od'];
42     $do=$row['do'];
43     $info=$row['info'];
44     $id_klijent=$row['id_klijent'];
45
46     $info = nl2br(htmlentities($info, ENT_QUOTES, 'UTF-8'));
47
48 $potrazivac=mysqli_query($db, "SELECT * FROM users WHERE id='$id_klijent' LIMIT 1");
49 $rr=mysqli_fetch_array($potrazivac, MYSQLI_ASSOC);
50 $skola=$rr['naziv'];
51

```

Sl. 3.3. Dohvaćanje podataka o oglasu

```

182 <script>
183 /*****
184 *** skripta za generiranje mape
185 *****/
186 $(document).ready(function(){
187     $("address").each(function(){
188         var embed =
189             "<iframe width='320' height='300' frameborder='0'"
190             + " scrolling='no' marginheight='0' marginwidth='0'"
191             + " src='https://maps.google.com/maps?&saddr="
192             + encodeURIComponent('<?php echo $start; ?>')
193             + "&daddr="+ encodeURIComponent( $(this).text() )
194             + "&z=7&output=embed'></iframe>"
195         ;
196         $(this).html(embed);
197     });
198 });
199 </script>
200 </body>
201 </html>

```

Sl. 3.4. Javascript skripta za generiranje mape

(Sl. 3.4.) prikazuje i jedinu Javascript skriptu u cijeloj aplikaciji jer se u potpunosti odvija klijentskim putem. Skripta je inicijalizirana putem jQuerya nakon što se kompletna stranica učita. Nakon toga jQuery za svaki element označen tag-om <address> stvara varijablu 'embed'.

Varijabla zapravo sadržava <iframe> tag s mnogobrojnim parametrima:

- Width/height – širina i visina interaktivne mape
- Frameborder – jačina ruba kontejnera mape
- Marginheight/marginwidth – horizontalne i vertikalne margine mape

- Src – poveznica na Google maps s dodatnim parametrima
 - Saddr – početna adresa kod navigacije
 - Daddr – krajnja adresa kod navigacije

Parametri za početnu i završnu adresu nalaze se unutar poveznice koje se šalju Google-u, tada se iz sigurnosnih razloga prvo mora prebaciti u HTML niz znakova, odnosno preko ključne riječi `jQuery` `this` kojom se preuzima sadržaj trenutno skeniranog tag-a te ga preko funkcije `encodeURIComponent()` prebacujemo u željeni format.

3.6. Nova ponuda

Ukoliko se na prethodno spomenutoj stranici oglasa nađe ponuđač, on ima mogućnost poslati svoju ponudu klikom na dugme 'STVORI PONUDU' koja preusmjerava korisnika na stranicu s praznim obrascem koji mora ispuniti da bi se ponuda generirala. Formular je obična HTML forma s `POST` metodom koja traži očekivanu cijenu u kunama za ponuđača, te opis ponude. Klikom na dugme 'POTVRDI' pokreće se skripta koja skuplja podatke iz forme te ih putem `INSERT` naredbe unosi u tablicu `ponude`.

```

10  if(isset($_POST['potvrdi'])){
11
12      $cijena=mysqli_real_escape_string($db, $_POST['cijena']);
13      $opis=mysqli_real_escape_string($db, $_POST['opis']);
14      $id_lo=mysqli_real_escape_string($db, $_POST['id_lo']);
15
16      $query=mysqli_query($db, "INSERT INTO ponude(id_ponudac, cijena, opis, id_lo)
17                                  VALUES ('$id_user', '$cijena', '$opis', '$id_lo')");
18

```

Sl. 3.5. Unos ponude

3.7. Pregled i prihvaćanje ponuda

Ako se na stranici s detaljima oglasa nađe korisnik koji je taj oglas i objavio, onda on uz detalje oglasa ima mogućnost pregledati sve ponude (sa svim njenim detaljima) korisnika koji spadaju pod klasu prijevoznika. Prikaz zaprimljenih ponuda je prikazan na (Sl. 3.7.). Kada se korisnik odluči za neku od ponuda, odobrava ju klikom na istoimeno dugme, što će automatski taj oglas učiniti neaktivnim, a prihvaćenu ponudu pobjedničkom (Sl. 3.6.).


```

4  /*****
5  ***** ODOBRI
6  *****/
7  if(isset($_POST['odobri'])){
8      $id_po = mysqli_real_escape_string($db, $_POST['id_po']);
9      $id_lok = mysqli_real_escape_string($db, $_POST['id_lo']);
10     //odbija sve ponude
11     $qc = mysqli_query($db, "SELECT id FROM ponude WHERE id_lok='$id_lok'");
12     while($counter = mysqli_fetch_array($qc)){
13         $odbij = mysqli_query($db, "UPDATE ponude SET ok='1' WHERE id_lok='$id_lok'");
14     }
15     //prihvaca pobjednicku ponudu
16     $odobri =mysqli_query($db, "UPDATE ponude SET ok='2' WHERE id='$id_po'");
17     //oglas je neaktivan
18     $dogovori=mysqli_query($db, "UPDATE lokacije SET dogovoreno=1 WHERE id='$id_lok'");
19
20     header('location: index.php');
21     exit();
22 }
23

```

Sl. 3.6. Prihvaćanje ponude

Bez obzira koju ponudu korisnik odabere svim ponudama se status mijenja iz 'na razmatranju' u 'odbijeno' koristeći SQL naredbu UPDATE, izuzev one prihvaćene koja ima pobjednički status.

```

62 /***** RENDEROVANJE PONUDA *****/
63 ***** RENDER - potrazivac
64 *****
65 $query=mysqli_query($db, "SELECT * FROM ponude WHERE id_lok='$id_lok'"); //traži sve ponude za trenutni oglas
66 if($ct=mysqli_num_rows($query)){ //ako postoji barem jedna ponuda na taj oglas
67 //stvaranje outputa #1
68 $rend = '
69     <div class="row">
70         <div class="span12">
71             <ul class="nav nav-tabs">
72             ';
73 //oslobodimo rezultat
74 mysqli_free_result($query);
75
76 $query=mysqli_query($db, "SELECT * FROM ponude WHERE id_lok='$id_lok'"); //traži detalje tih ponuda
77 while ($row1=mysqli_fetch_array($query, MYSQLI_ASSOC)) {
78 //baci u varijable za daljnje korištenje
79 $id_po=$row1['id'];
80 $cijena=$row1['cijena'];
81 $opis=$row1['opis'];
82 $id_ponudac=$row1['id_ponudac'];
83 //formatiranje prikaza - enter je novi red
84 $opis = nl2br(htmlentities($opis, ENT_QUOTES, 'UTF-8'));
85
86 $q=mysqli_query($db, "SELECT * FROM users WHERE id='$id_ponudac' LIMIT 1"); //traži detalje ponuđača
87 $r=mysqli_fetch_array($q);
88 $pname=$r['naziv']; //naziv firme koa nudi oglas
89
90 //stvaranje outputa #2
91 $rend .= '
92     <div id="tab-' . $i . '" class="tab-pane fade ' . $cl . '">
93         <table class="table table-striped">
94             <tbody>
95                 <tr>
96                     <th>Ponuđač</th>
97                     <td>' . $pname . '</td>
98                 </tr>
99                 <tr>
100                    <th>Cijena</th>
101                    <td>' . $cijena . '</td>
102                </tr>
103                <tr>
104                    <th>Opis</th>
105                    <td>' . $opis . '</td>
106                </tr>
107                <tr>
108                    <td><form method="POST" action="zadatak.php">
109                        <input type="hidden" name="id_po" value="' . $id_po . '">
110                        <input type="hidden" name="id_lo" value="' . $id_lo . '">
111                        <button type="submit" name="odobri" class="btn btn-primary">ODOBRI</button>
112                    </form></td>
113                    <td><td>
114                </tr>
115            </tbody>
116        </table>
117    </div>';
118    $i++;
119 }
120 //stvaranje outputa #3
121 $rend .= '
122     </div>
123 </div>
124 </div>
125

```

Sl. 3.7. Generiranje liste zaprimljenih ponuda

3.8. Status ponuda

Autoprijevoznici imaju mogućnost pregledati sve svoje ponude prema njihovom statusu. Primjer se nalazi na (Sl. 3.8.).

Lijevi vertikalni izbornik sadrži poveznicu na statusnu stranicu gdje se sve stvorene ponude razvrstavaju po bojama u ovisnosti o statusu.

- Zelena – prihvaćene ponude
- Plava – ponude još uvijek na razmatranju
- Crvena – odbijene ponude

```
5
6 /*****
7 ***** RENDER - prihvaceno
8 *****/
9     $d_rend="";
10    $p_rend="";
11    $o_rend="";
12
13 //traži sve uspjesne ponude
14 $detalji_ponuda=mysqli_query($db, "SELECT * FROM ponude WHERE ok='2' AND id_ponudac='$id_user'");
15 while($dr=mysqli_fetch_array($detalji_ponuda, MYSQLI_ASSOC)){
16     $cijena=$dr['cijena'];
17     $opis=$dr['opis'];
18     $id_oglasa=$dr['id_lok'];
19
20     //traži detalji trenutnog oglasa
21     $detalji_oglas=mysqli_query($db, "SELECT * FROM lokacije WHERE id='$id_oglasa'");
22     while($row1=mysqli_fetch_array($detalji_oglas, MYSQLI_ASSOC)){
23         $id=$row1['id'];
24         $start=$row1['start'];
25         $end=$row1['end'];
26         $od=$row1['od'];
27         $do=$row1['do'];
28
29         //generiranje outputa
30         $d_rend .= '
31         <tr><form action="zadatak.php" method="POST">
32             <input type="hidden" name="id" value="' . $id . '">
33             <td>' . $start . ' -> <b style="color:red;">' . $end . '</b> .</td>
34             <td>' . $od . ' -> ' . $do . '</td>
35             <td>' . $cijena . '</td>
36             <td>' . $opis . '</td>
37         </form></tr>
38         ';
39     }
40 }
41
```

Sl. 3.8. Generiranje popisa uspješnih ponuda

Taj se ispis kasnije u HTML dijelu ubacuje u tablicu gdje se formatira izvještaj zajedno s odbijenim ponudama kao i onima koje su još uvijek na čekanju te im se status nije promijenio.

4. DIZAJN

S gotovim stražnjim dijelom ostalo je još samo opisati proces izrade sučelja. Kako navodimo još u prvom poglavlju, za dizajn korisničkog sučelja koristimo vanjske biblioteke (Bootstrap, jQuery) uz minimalne preinake u obliku ručno definiranih CSS stilova, na elementima na kojima je to bilo potrebno.

Kao shema stranice korišten je klasično zaglavlje-2stupca-podnožje raspored, odnosno izgled svake stranice se u pravilu sastoji od zaglavlja i podnožja (objašnjenih ranije u sekciji 'Predložci'), te tijela stranice koje rastavljamo na dva nejednaka stupca – manji sadržava vertikalni izbornik, a drugi prikazuje sam sadržaj aplikacije.

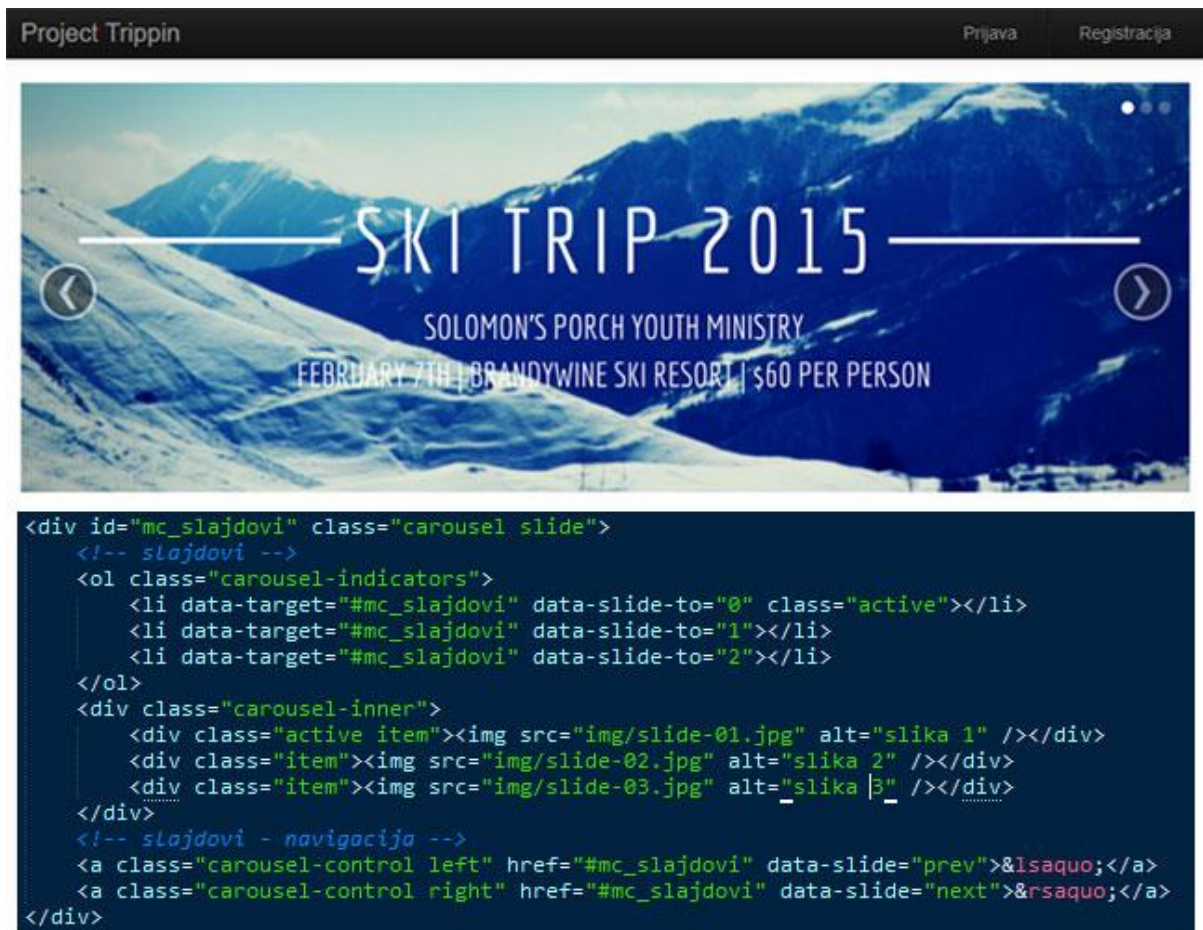
Osim generalnog predloška koje stranice uglavnom prate, ponekad je potrebno ubaciti neki element koji je vezan isključivo za određenu stranicu kao što je slideshow koji vidimo na (Sl. 4.1.).

Slideshow se postiže uz pomoć Bootstrap klase: Unutar kontejnera (`div class="carousel slide"`) koji služi kao omot za slideshow, definira se novi kontejner (`div class="carousel-inner"`) koji sadrži sve slike koje taj slideshow prikazuje. Slike su deklarirane klasično preko `` tag-a unutar `<div>` elementa klase *item*.

Indikatori za redni broj slike u projekciji se deklarira kao posložena lista (*eng. ordered list*) te koristimo dva parametra:

- `data-target`: pridodaje se vrijednost vanjskog kontejnera dijaprojekcije
- `data-slide-to`: redni broj slike ako se koristi *on-click* ili ako se slika u projekciji sama promijeni

Navigaciju u obliku strjelica na horizontalnim rubovima slike dobijemo koristeći `<a>` tag. Klasa dodijeljena linku (`carousel-control left/right`) definira poziciju tipke za navigaciju, `href` atribut kao metu dobiva identifikator vanjskog kontejnera slideshow-a, a `data-slide` parametar određuje smjer kretanja projekcije. Sama strelica nastaje koristeći entitet `&lsaquo/&rsaquo` ovisno o smjeru strelice.



Sl. 4.1. Izgled i kod slideshow-a na naslovnoj stranici

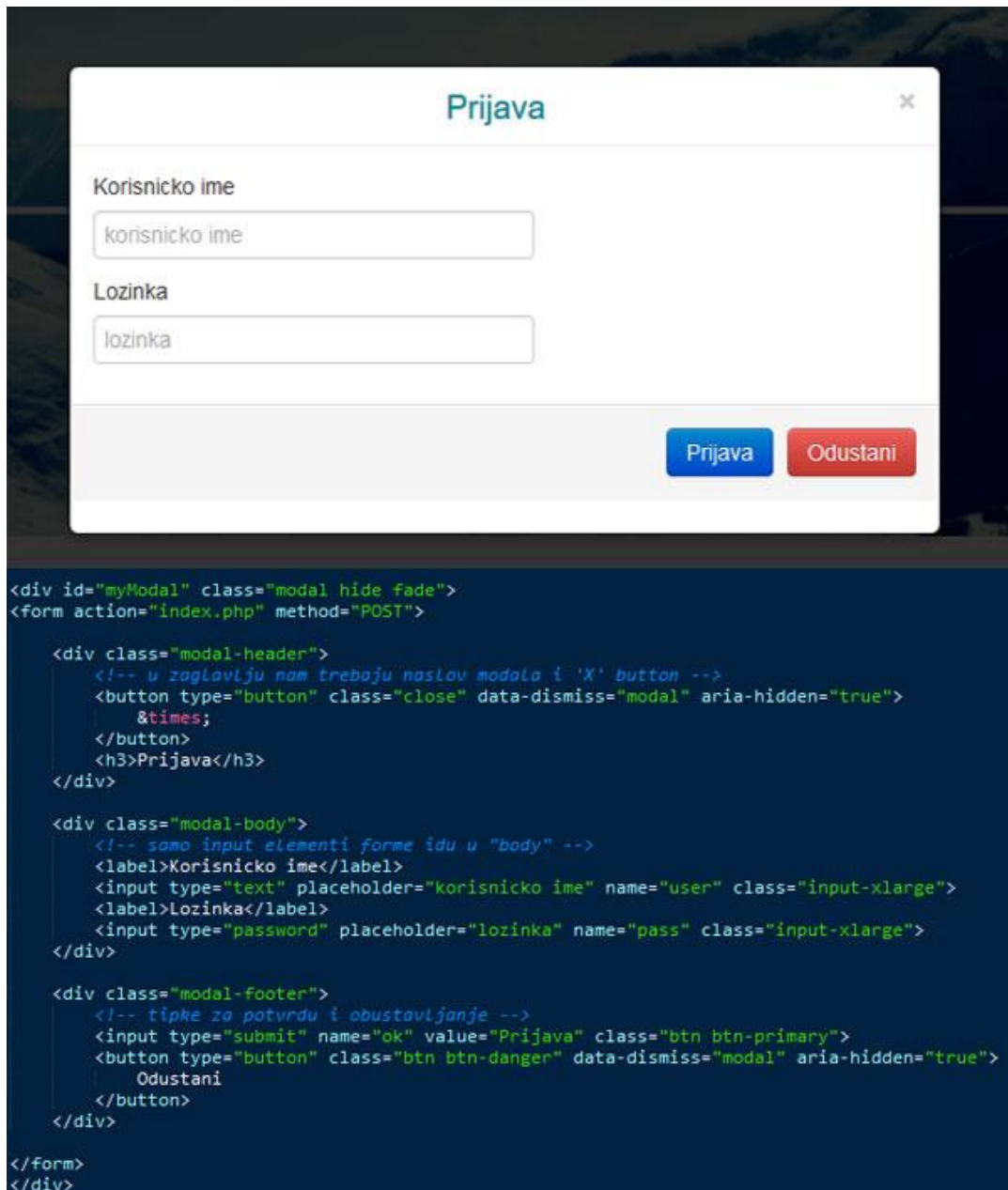
Nakon sistema dijaprojeksije, na redu je još jedna novina na naslovnoj stranici aplikacije a to je modalni dijalog sistema za prijavu. Na (Sl. 4.2.) je usporedno prikazan dizajn dijaloga i sam kod koji ga definira.

Kao i svi elementi dizajnirani putem Bootstrap biblioteke, i modalni dijalog se definira unutar nekog vanjskog kontejnera klase *modal*. Klasa *hide* osigurava da dijalog ostane sakriven sve dok se ne klikne poveznica kojom je vrijednost *href* parametra identifikatorska oznaka vanjskog kontejnera.

Kao što se može vidjeti na slici, sam dijalog sadrži 3 dijela: zaglavlje, tijelo dijaloga i njegovo podnožje.

- Zaglavlje – kontejner klase *modal-header* sadrži elemente na samom vrhu dijaloga (naslov i tipka za prekid)

- Tijelo – kontejner klase modal-body sadrži input elemente obrasca za prijavu (korisničko ime i zaporka)
- Podnožje - kontejner klase modal-footer sadrži završne elemente forme (tipka za prijavu) i dijaloga (tipka za prekid operacije).



Sl. 4.2. Izgled i kod modalnog dijaloga sistema za prijavu

Nakon prijave, korisnik pristupa naslovnoj stranici ali s mogućnošću pregleda aktualnih potraživanja. Oglasi su prikazani u obliku tablice (Sl. 4.3.) sa vanjskim kontejnerom klase `table-responsive` koja je zadužena za dinamičko određivanje širine elementa ovisno o veličini ekrana na kojem se prikazuje.

Unutar kontejnera deklarirana je sama tablica i zadane su joj klase:

- *Table* – osnovna klasa za tablice (horizontalne granice i lagano povećanje)
- *Table-hover* – klasa omogućuje 'lebdjenje' na redove tablice
- *Table-condensed* – klasa još više čini tablicu kompaktnom tako što povećanje ćelija smanjuje za pola



The screenshot displays a web interface with a table titled "AKTUALNE PONUDE". The table has three columns: "# RUTA", "INTERVAL", and "AKCIJE". It lists three offers with their respective routes and dates. Below the table, there is a copyright notice and a code block showing the HTML structure used to render the table.

#	RUTA	INTERVAL	AKCIJE
1.	Osijek, Hrvatska ↔ Požega, Hrvatska	2016-04-24 ↔ 2016-04-30	PREGLEDAJ
2.	Osijek, Hrvatska ↔ Zagreb, Hrvatska	2016-11-15 ↔ 2016-11-19	PREGLEDAJ
3.	Osijek, Hrvatska ↔ Varaždin, Hrvatska	2016-11-15 ↔ 2016-11-19	PREGLEDAJ

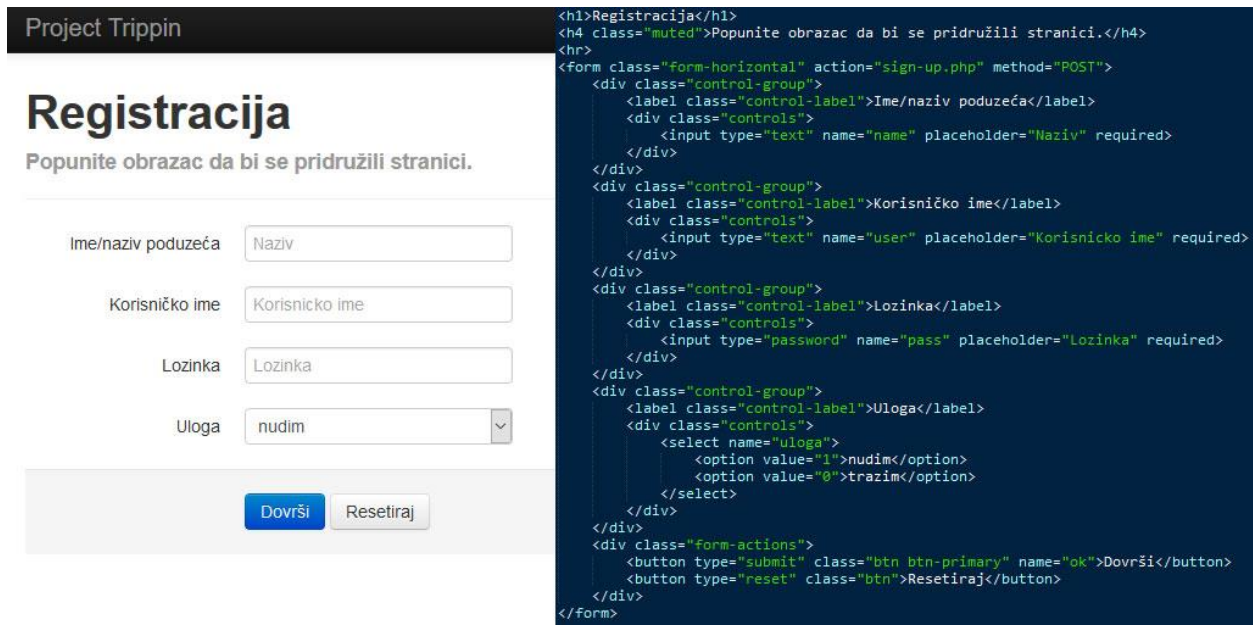
```
<h3>AKTUALNE PONUDE</h3>
<div class="table-responsive">
  <table class="table table-hover table-condensed">
    <?php echo $rend; ?>
  </table>
</div>
```

Sl. 4.3. Izgled i kod tablice oglasa

Kako veći dio aplikacije otpada na raznorazne predloške i obrasce, oni se rade na način kao što se može vidjeti na (Sl. 4.4.).

Obrascu se dodaje klasa `form-horizontal` koja osigurava da elementi obrasca budu poredani jedan ispod drugoga i poravnani kao kada se koristi tablica. Svaki „red“ obrasca stavljamo u poseban kontejner s klasom `control-group`, a svaki takav kontejner sadrži dva elementa – kontejner u kojem je smješten element obrasca i njegova oznaka.

Oznaka sama za sebe ima klasu `control-label` s kojom definiramo poziciju prvog stupca obrasca odnosno stupca s opisima elemenata. Kontejner je ponovno `<div>` element s klasom `controls` i on definira širinu i visinu drugog stupca obrasca u odnosu na broj elemenata u pojedinom kontejneru.



Sl. 4.4. Izgled i kod forme za registraciju

Svaki tekstualni element obrasca za parametar sadrži `required` koji osigurava da se obrazac ne može poslati ukoliko je to polje prazno. Dok polje za unos zaporke ima poseban parametar `password` koji sakriva unos zamjenjujući ga nizom točkica.

Završni dio obrasca se razlikuje od ostatka po tome što njegov kontejner nosi klasu `form-actions` koji definira boju pozadine i margine za dio obrasca u kojem se nalaze tipke za slanje i obustavljanje procesa. Klase na samim tipkama određuju boju tipke na svim njihovim stanjima, a parametar `type` njihovu akciju.

Dio aplikacije koji služi da bi potraživači mogli pregledati sve ponude upućene za neki od njihovih oglasa je gotovo u potpunosti generiran dinamički, ali i dalje koristi Bootstrap elemente.

Za kreiranje izgleda ovog sistema koristi se sustav tab-ova unutar Bootstrap-a i on se sastoji od dva dijela a to su:

- Neporedana lista
- Kontejner sa sadržajem

Neporedana lista zapravo služi kao popis linkova na identifikatore unutarnjih kontejnera sa sadržajem. Njoj pridodajemo klase `nav` i `nav-tabs` da bi se inicijalizirala kao navigacijska lista za ispod deklarirani kontejner i kako bi poprimila dizajn horizontalnog izbornika sa svim stanjima. Također se svakom pojedinom linku daje parametar `data-toggle='tab'` da bi se poveznice natjeralo da se ponašaju isključivo u ovisnosti na trenutno aktivni tab.

Vanjski kontejner sadržaja kao u slučaju obrasca ima nekoliko unutarnjih kontejnera ovisno o broju poveznica u listi. Pridodaje mu se klasa `tab-content` koji određuje poziciju elemenata unutar njega u odnosu na elemente navigacije.

Unutarnji kontejneri se uglavnom sastoje od dva dijela a to su naslov odlomka i njegov tekst. Samom kontejneru dajemo unikatni identifikator i klasa `tab-pane` i prvom odlomku se još dodaje klasa `active` kako bi odredili koji će od tab-ova biti prvi prikazan.



1. PONUDA
2. PONUDA

Ponudač	Belisce-bus
Cijena	1200
Opis	<ul style="list-style-type: none"> - opisati ponudeni paket usluga - 3x noćenje/doručak - Hotel 3* u zoni javnog prijevoza - Većernji polazak, više vremena za razgledavanje - uključen Beč u povratku - odrediti rutu - via Bizovac - Nasice (1h 7min)

ODOBRI

Od:

```

while ($row=mysqli_fetch_array($query, MYSQLI_ASSOC)) {
    $id_po=$row['id'];
    $cijena=$row['cijena'];
    $opis=$row['opis'];
    $id_ponudac=$row['id_ponudac'];

    $opis = nl2br(htmlentities($opis, ENT_QUOTES, 'UTF-8'));

    $sql=mysqli_query($db, "SELECT * FROM users WHERE id='$id_ponudac' LIMIT 1");
    $row=mysqli_fetch_array($sql);
    $ime=$row['naziv'];

    $cl='';
    $i<2 ? $cl='in active' : $cl='';

    $srand.=' '
    <div id="tab- . $i . '" class="tab-pane fade . $cl . '">
        <table class="table table-striped">
            <tbody>
                <tr>
                    <th>Ponudač</th>
                    <td> . $ime . '</td>
                </tr>
                <tr>
                    <th>Cijena</th>
                    <td> . $cijena . '</td>
                </tr>
                <tr>
                    <th>Opis</th>
                    <td> . $opis . '</td>
                </tr>
                <tr>
                    <td><form method="POST" action="zadatak.php">
                        <input type="hidden" name="id_po" value=" . $id_po . '">
                        <input type="hidden" name="id_lo" value=" . $id_lo . '">
                        <button type="submit" name="odobri" class="btn btn-primary">ODOBRI</button>
                    </form></td>
                    <td><td>
                </tr>
            </tbody>
        </table>
    </div>
    $i++;
}
    $srand.=' '
</div>
</div>
';
}

```

Sl. 4.5. Izgled i kod sistema za ponude

U aplikaciji se ovaj sistem koristi na način da se dinamički generira lista aktivnih ponuda za zadani oglas sa svim njihovim detaljima, te se kroz petlju generira ispis ponuda formatiran tako da odgovara sistemu – posebno se prikazuje HTML za listu a posebno za sadržaj.

Svaka od navedenih ponuda u sebi ima obrazac s nekoliko skrivenih polja koja služi da bi objavljivač oglasa mogao prihvatiti neku određenu ponudu. Format „opis“ sekcije – uključujući nove redove, dobiva preko PHP funkcija:

```
nl2br( htmlentities($opis, ENT_QUOTES, 'UTF-8'));
```

gdje *htmlentities()* uzima tri parametra:

- string koji zahtjeva provjeru
- zastave (eng. flags) – određuje kako će se funkcija odnositi prema navodnicima (ENT_QUOTES enkodira i navodnike i apostrofe)
- set znakova (UTF-8)

5. ZAKLJUČAK

Ovaj rad prikazuje proces izrade funkcionalnog, web-baziranog sustava za potražnju i ponudu rješenja prijevoza za planirana putovanja. Iako možda zvuči komplicirano, činjenica da postojanje slobode za odabir programskih rješenja, ali i načina izvedbe ove teme, kompleksnost je svedena na razinu koncepta da bi se pokazalo na koji način takvi sustavi funkcioniraju u stvarnom svijetu i koji koraci su potrebni za njegovu realizaciju.

Naravno, ovo je grana u kojoj ima bezbroj rješenja za svaki problem, ali u tome i jest čar ovakvih projekata. Počelo je s definiranjem radnog okruženja, a tu je PHP/MySQL očiti pobjednik na fakultativnoj razini zbog slobodne licence, ostali korišteni alati također su besplatni.

Na početku rada je ustanovljeno da je program s namjerom napravljen jednostavno da bi se izbjegla potreba za korištenjem paradigme, koja je i suviše apstraktna za dokazivanje koncepta rada neke aplikacije, ali kako se projekt primicao kraju, postajalo je sve očitije da bez obzira koliko pojednostavili ideju za rješenje, ono će na kraju ispasti kompleksno.

No, zadatak je uspješno priveden kraju. Funkcionalnost je rastavljena na cjeline, te ju nadograđivali kako se program širio, ne dopuštajući da se aplikacija izrodi.

LITERATURA

- [1] <http://php.net/> 16.04.2016.
- [2] <http://getbootstrap.com/getting-started/> 18.04.2016.
- [3] <http://www.w3schools.com/> 22.05.2016.
- [4] <https://www.codecademy.com/learn> 25.05.2016.

SAŽETAK

Cilj ovog rada je izrada sustava za potraživanje i prihvaćanje ponuda za putovanja, temeljen na internetskim tehnologijama što je u konačnici i postignuto korištenjem kombinacijom PHP/MySQL.

Nakon opisa korištenog radnog okruženja i tehnologija, navedena je uloga baze podataka u ovoj web aplikaciji. Fokus je kasnije stavljen na funkcionalnost programa i dizajn korisničkog sučelja.

Testiranje aplikacije potvrdilo je da je ovaj sustav iako rađen s idejom da bude što jednostavniji, ipak vrlo kompleksan, ali svejedno izvediv.

Ključne riječi: ponude, web, sustav, aplikacija, putovanje

ABSTRACT

WEB SYSTEM FOR SETTING AND COLLECTING OFFERS FOR EXCURSIONS

This graduate paper explains designing and developing web application which will be able to require and accept travel offers. Application is developed using PHP/MySQL web technologies.

After description of framework and development environment role of database in this web application is explained. Focus is then moved to the application functionality and designing user interface.

This type of web application can be easilly designed and developed but its functionality might look complex.

Key words: offers, web, system, program, travel

ŽIVOTOPIS

Denis Dubovečak rođen je 14.10.1994 u Osijeku. 2009 godine je završio osnovnu školu „Franjo Krežma“ s vrlo dobrim uspjehom te nakon toga upisuje srednju školu „Ekonomska i upravna škola“ koju završava 2013. godine. Iste godine upisuje stručni studij na Elektrotehničkom fakultetu u Osijeku, smjer Informatika.

Aktivno se služi engleskim jezikom, računalom i MS paketima (Word, Excel, PowerPoint).