Web aplikacija za vođenje evidencije iznajmljivanja objekata

Knezović, Božo

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://urn.nsk.hr/urn:nbn:hr:200:667719

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-25**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
ELEKTROTEHNIČKI FAKULTET

Stručni studij
Smjer informatika

WEB APLIKACIJA ZA VOĐENJE EVIDENCIJE IZNAJMLJIVANJA OBJEKATA

Završni rad

Student: Božo Knezović

Mentor : Doc.dr.Krešimir Nenadić

Osijek, 2016.

Sadržaj

1. UVOD 1
1.1. Zadatak završnog rada 1
2. RAZRADA ZADATKA 2
2.1. Baza podataka 2
2.2. Datotečna struktura 5
3. IZRADA APLIKACIJE 7
3.1. Sustav prijave 7
3.2. Naslovnica 10
3.3. Pretraga 11
3.4. Detalji proizvoda 14
3.5. Rezervacije 19
3.6. CRUD oglasa 21
4. ZAKLJUČAK 24
5. LITERATURA 25
SAŽETAK 26
ABSTRACT 27
ŽIVOTOPIS 28

1. UVOD

Zadatak ovog završnog rada je izrada web aplikacije za vođenje evidencije iznajmljivanja objekata. U sljedećem poglavlju razrađen je plan izvedbe. Također su navedeni alati i tehnologije koji su se koristili za izradu web aplikacije i njezine baze podataka, te je opisana datotečna struktura samog projekta. Kod izbora tehnologija razmatrat će se isključivo softver dostupan za besplatno korištenje.

Nakon izvedene arhitekture aplikacije, opisan je proces njezine izrade. Kako aplikacija podržava više klasa korisnika u tom dijelu rada prvo će biti obrađen iz perspektive jedne vrste korisnika, a zatim i druge. No, najprije će biti opisano ono što je svima njima zajedničko, a to je sustav za prijavu/registraciju te pregledavanje objavljenih oglasa.

Nakon završene obrade procesa izrade ovog sustava nekoliko će stranica biti posvećeno testiranju programa.

1.1. Zadatak završnog rada

Objasniti način iznajmljivanja nekog objekta uz pojašnjenje svi relativnih parametara za taj proces. Dizajnirati bazu podataka pomoću koje će se voditi evidencija kratkoročnih zakupa nekog objekta. Predvidjeti mogućnost zadavanja vrste i svrhe objekta, različitih uloga i prava pristupa korisnika te vođenje detaljne evidencije iznajmljivanja objekta. Napraviti web sučelje prema korisnicima i funkcionalnost za pristup bazi podataka. Objasniti način rada cijele aplikacije kao i način testiranja funkcionalnosti.

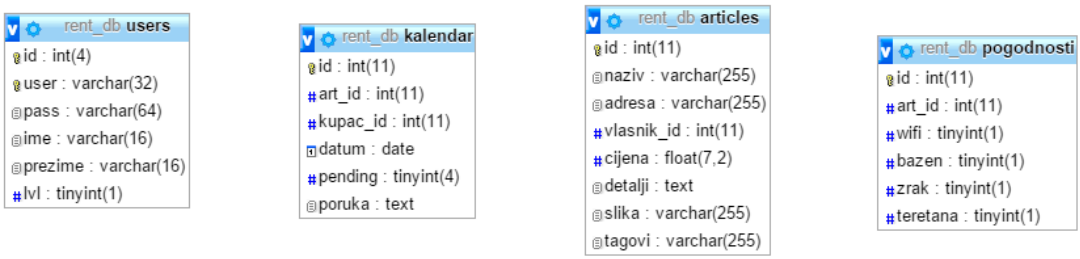
2. RAZRADA ZADATKA

Rješavanje bilo kojeg zadatka počinje tako što se odaberu alati za njegovu izvedbu. Budući da je naslov zadatka„ Web aplikacija za evidenciju iznajmljivanja objekata“, može se zaključiti da će njezina konstrukcija zahtijevati vrlo specifične tehnologije. Pojam web aplikacija govori da će biti potreban neki poslužiteljski jezik kako bi se napravila sama jezgra aplikacije. PHP se u ovome iskazao kao najbolji zbog svoje licence, a u tom će se pravcu razmišljati i o svim drugim idealnim rješenjima. *MySQL* kao baza podataka za ovu aplikaciju također se nameće sama od sebe. Dobra stvar je što se i jedno i drugo može dobiti u paketu sa lokalnim serverom u obliku XAMPPA te su tako riješeni svi problemi s backend strane.

Za pisanje HTML-a, CSS-a i Javascripta nije potrebno nikakvo razvojno okruženje, ali ipak, zbog jasnoće čitanja koda u budućnosti, kao uređivač teksta koristit će se SublimeText 3 zbog lakoće snalaženja i fleksibilnosti u radu.

2.1. Baza podataka

Nakon instalacije svih potrebnih programa, prvo na redu jest izrada baze kako bi se potpuna pozornost mogla posvetiti isključivo kodiranju. Baza koja je zamišljena imat će 4 tablice koje su detaljno objašnjene u nastavku.



Sl. 2.1. Pogled na tablice

Gore prikazane tablice nisu međusobno povezane jer se korištena baza vrti na MYISAM *engineu* koji nema podršku za relacije, no kako baza sama po sebi nije

kompleksna ne postoji potreba za njima. Prva tablica je korisnička tablica „users“(slika 2) koja bi trebala sadržavati osobne podatke o korisnicima, njihove zaporke i korisnička imena, kao i razinu privilegija koje će imati, odnosno o kojoj vrsti korisnika se radi. Za tipove podataka za sva polja (osim identifikatorskog koji je uvijek INT) stavljen je VARCHAR, osim na polju koje određuje vrstu korisnika, kod kojeg je korišten. BOOLEAN. BOOLEAN je u suštini mali INT kojem se 0 tumači kao laž, a bilo koji broj veći od nule kao istina. Kako ova aplikacija ima samo dvije vrste korisnika, ovo je sasvim dovoljno.

	#	Name	Type	Collation	Attributes	Null	Default	Extra
	1	**id**	int(4)			No	*None*	AUTO_INCREMENT
	2	**user**	varchar(32)			No	*None*	
	3	**pass**	varchar(64)			No	*None*	
	4	**ime**	varchar(16)			No	*None*	
	5	**prezime**	varchar(16)			No	*None*	
	6	**lvl**	tinyint(1)			No	1	

Sl. 1.2. Tablica korisnika

Nakon korisničke tablice, na redu je tablica koja bi katalogizirala nekretnine, a to je tablica „articles“ koja sprema podatke o nekretninama koje su stavljene na tržište kratkoročnih zakupa.

Polja u toj tablici su naziv nekretnine i njezina adresa, od kojih su oboje pojmovi s malim brojem znakova pa se stoga stavlja VARCHAR kao tip podatka. Nakon njih ide identifikacijska oznaka vlasnika zgrade, koja je kao i svako ID polje – INT. Zatim slijedi još jedno numeričko polje tipa FLOAT koje sprema cijenu najma po danu. FLOAT kao tip podatka zahtijeva da mu se odredi veličina preko dva parametra – prvi je ukupni maksimalni broj znamenaka koje može spremiti, a drugi koliko od tih znamenaka je rezervirano za decimalne brojeve. U ovom primjeru to znači da se u nju sprema realni broj čija je maksimalna vrijednost 9999, 99.

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	**id**	int(11)			No	*None*	AUTO_INCREMENT
2	**naziv**	varchar(255)			No	*None*	
3	**adresa**	varchar(255)			No	*None*	
4	**vlasnik_id**	int(11)			No	*None*	
5	**cijena**	float(7,2)			No	*None*	
6	**detalji**	text			No	*None*	
7	**slika**	varchar(255)			No	*None*	
8	**tagovi**	varchar(255)			No	*None*	

Sl. 2.2. Tablica nekretnina

Naredni atribut u spomenutoj tablici je opis zgrade koju piše iznajmljivač. Kako nitko ne može predvidjeti koliko bi velik bio taj opis, tip ovog polja definirale kao TEXT koji je tip najveće vrijednosti u string kategorije podataka. Nakon opisa nalaze se još dva VARCHAR polja od kojih prvo imenovano:„slika“ koje zapravo sprema url adresu učitane slike za naknadno korištenje, a drugo nosi ime „tagovi“ i sprema niz stringova odvojenih zarezom koji će se kasnije koristiti za filtriranje ponuda po ključnim riječima.

Tablica pogodnosti jest pomoćna tablica tablici articles jer ona zapravo samo sprema identifikatore zgrada i neke osnovne pogodnosti koje ta zgrada nudi a moderno tržište zahtijeva.

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	**id**	int(11)			No	*None*	AUTO_INCREMENT
2	**art_id**	int(11)			No	*None*	
3	**wifi**	tinyint(1)			No	*None*	
4	**bazen**	tinyint(1)			No	*None*	
5	**zrak**	tinyint(1)			No	*None*	
6	**teretana**	tinyint(1)			No	*None*	

Sl. 2.3. Tablica *pogodnosti*

Tablica svaku od tih pogodnosti sadrži pod tipom BOOLEAN jer određena pogodnost ili postoji ili ne postoji.

Zadnja tablica u ovoj bazi je tablica „kalendar“ u koju se zapisuju zahtjevi ljudi koji bi neku zgradu htjeli iznajmiti i te zahtjeve razlaže na sljedeće dijelove: prvi su dio dva identifikatora: id korisnika koji je uputio zahtjev te id zgrade koju želi iznajmiti.. „datum“ sadrži željeni termin jednodnevnog najma, a „poruka“ službenu molbu. Polje „pending“ status je molbe sa tri stanja - 0, 1 i 2 - koje redom označuju molbu u procesu, koju vlasnik zgrade još nije pogledao. Jedinica se dodjeljuje uspješnim molbama, dok je dvojka znak da je ista odbijena.

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	**id**	int(11)			No	*None*	AUTO_INCREMENT
2	**art_id**	int(11)			No	*None*	
3	**kupac_id**	int(11)			No	*None*	
4	**datum**	date			No	*None*	
5	**pending**	tinyint(4)			No	*None*	
6	**poruka**	text			No	*None*	

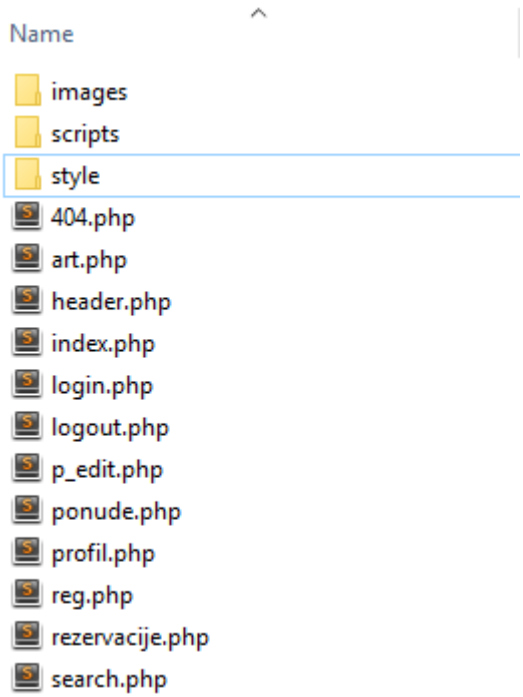
Sl. 2.4. Kalendarska tablica

2.2. Datotečna struktura

Kako je loša praksa cijelu aplikaciju definirati unutar jedne datoteke, ona je u sklopu ovog projekta, rastavljena na nekoliko manjih cjelina. Spomenuta struktura je grafički prikazana na sl 2.6.. Na njoj se vidi da je aplikacija podijeljena na 4 dijela, i to na :

1. / - root datoteka sadrži sve ostale datoteke i mape te se vidi da su direktno na njoj postavljene sve datoteke stranice, tj.sve stranice ove aplikacije. Tu su naslovnica, stranica za 404 grešku, stranice za prijavu/odjavu/registraciju te za CRUD operacije.
2. /images/ – mapa u kojoj se spremaju sve uploadane slike koje su vezane za iznajmljivane objekte, zato će url za svaku sliku počinjati sa „/images/<ime_slike>.<ekstenzija>“

3. /scripts/ - mapa jest ona koja sadrži datoteke važne za funkcioniranje stranice, ali manje opsežne nego normalna stranica kakve se nalaze na '/'. Tu su npr.skripte za povezivanje s bazom, inicijalizaciju sesije, provjera prijavljenosti, renderiranje kalendara, itd.- U ovoj je mapi sve ono što se ponavlja na svakoj stranici ili služi kao jedan dio neke veće skripte.
4. /style/ - mapa skladišti CSS stilove i razne elemente i ikonice korištene u svrhu dizajniranja izgleda same aplikacije



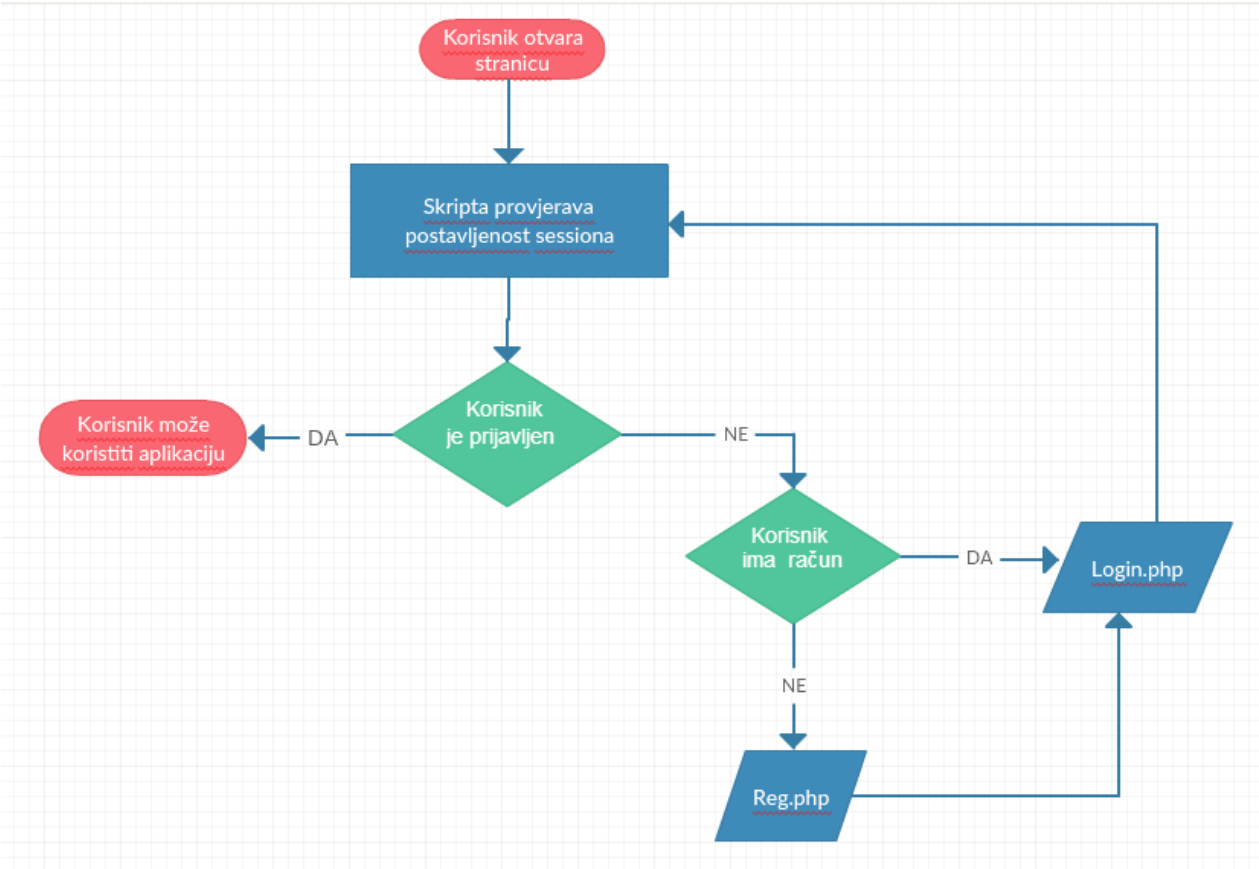
Sl. 2.5. Datotečna struktura

3. IZRADA APLIKACIJE

Sa postavljanjem baze i arhitekture završava onaj niži dio izrade ovog projekta. Tako da će se ovaj dio rada biti posvećen izradi logike stranice.

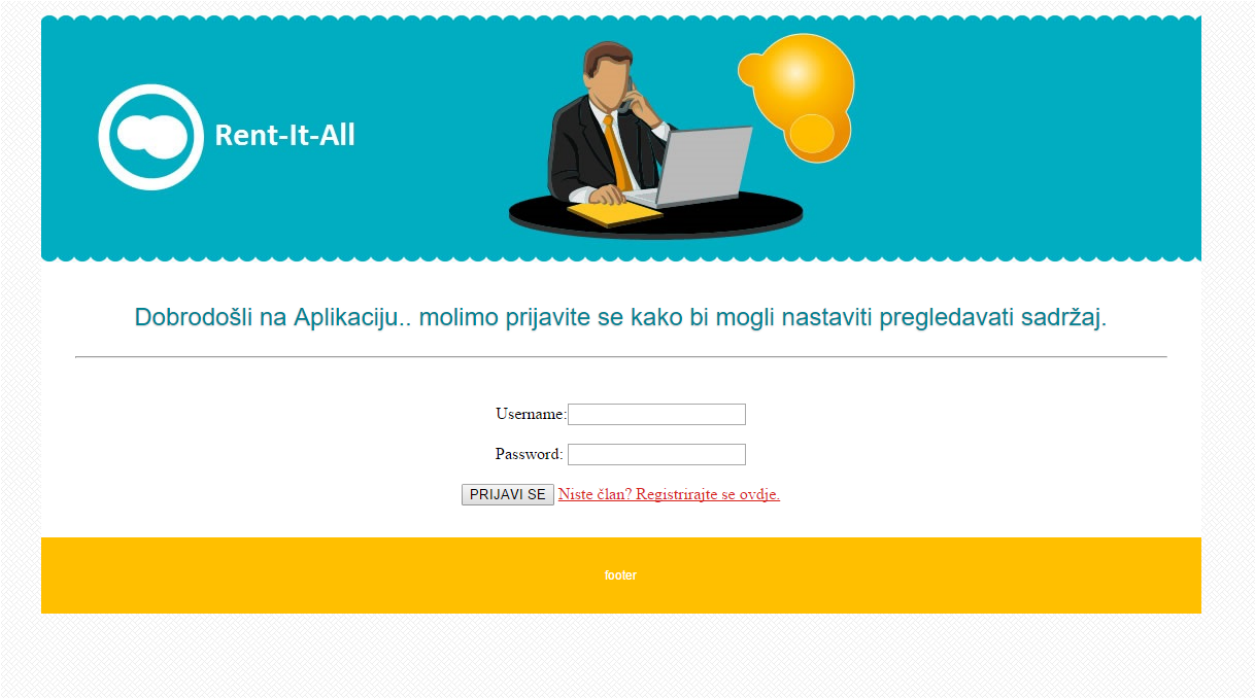
3.1. Sustav prijave

Izrada započinje kreiranjem sistema za prijavu/registraciju korisnika, nju će se najlakše razumjeti preko dijagrama.



Sl. 3.1. Dijagram toka za login sistem

Na slici je vidljivo da korisnik po učitavanju stranice nema pravo pristupa njezinim mogućnostima bez izrađenog računa. Skripta koja se nalazi u /scripts/status.phpprovjerava postoji li inicijalizirana sesija i ukoliko ne postoji korisnika prebacuje na 'login screen'.



Sl. 3.2. Neprijavljen korisnik je preusmjeren na stranicu za prijavu

Kako je vidljivo na slici, uz fomu za prijavu je i link na stranicu za registraciju novih korisnika. Ta stranica sadrži formu u kojoj se skupljaju sve informacije koje se INSERT naredbom ubacuju u tablicu „user“.

Forma za registraciju se sastoji isključivo od tekstualnih polja uz iznimku zadnjeg elementa koji određuje klasu korisnika, a taj je pak prikazan sa dva radio dugmeta. Radio je korišten zato što dopušta odabir samo jedne opcije. Kako je u tablici definirano da postoji dvije klase korisnika. ova je metoda odabrana za njezino popunjavanje.

Po ispunjenju forme i kreiranju profila, posjetitelj je ponovno preusmjeren na stranicu za prijavu gdje se novoregistrirani korisnik može prijaviti. Kod na slici 9. prikazuje skriptu za autentifikaciju posjetitelja da bi se utvrdilo imaju li pravo pristupa korištenju aplikacije.

U prvom dijelu bloka, ona zapravo koristi funkciju definiranu u /scripts/status.php da provjeri postojanje sesije te, ako ona postoji, preusmjeravaju korisnika na naslovnu stranicu. U praksi to znači da prijavljeni korisnici ne mogu vidjeti ovaj ekran, dok god se sesija ne obriše.

```php
<?php
include_once('scripts/status.php');
if (isset($_SESSION["id"]) && $user_ok == true) {
    header("location: index.php");
    exit();
}

if (isset($_POST["username"])) { //provjeravamo je li forma ispunjena

    //uzimamo variableiz podataka koje nam salje forma
    $username = mysqli_real_escape_string($db_conx, $_POST['username']);
    $password = mysqli_real_escape_string($db_conx, $_POST['password']);

    //saljemo upit bazi da bi provjerili postoji li takav korisnik
    $sql    = "SELECT * FROM users WHERE user='$username' AND pass='$password' LIMIT 1";
    $query  = mysqli_query($db_conx, $sql);
    $row    = mysqli_fetch_row($query);

    $id     = $row[0];
    $user   = $row[1];
    $pass   = $row[2];
    $lvl    = $row[5];

    if ($password != $pass || $username != $user) {
        echo 'Greska: korisnik ne postoji! ';
        echo '<a href="login.php">Vratite se natrag.</a>';
        exit();
    } else { //ako je sve u redu postavljamo session
        $_SESSION['id']   = $id;
        $_SESSION['user'] = $user;
        $_SESSION['pass'] = $pass;
        $_SESSION['lvl']  = $lvl;
    }

    header("location: index.php"); //i redirektamo korisnika na index.php
    exit();
}
?>
<!doctype html>
<html lang="hr">
<head>
```

Sl. 3.3. Login skripta

Drugi se dio bloka bavi samom prijavom korisnika nakon što korisnik pošalje ispunjenu formu za prijavu. Ona uzima poslane vrijednosti i uspoređuje ih sa podacima korisnika spremljenih u bazi te, ako nađe podudaranje, postavlja varijable sesije te preusmjerava korisnika na naslovnicu. Ako takav korisnik ne postoji u bazi, ona ispisuje grešku te vraća korisnika, ponovno, na ovu stranicu.

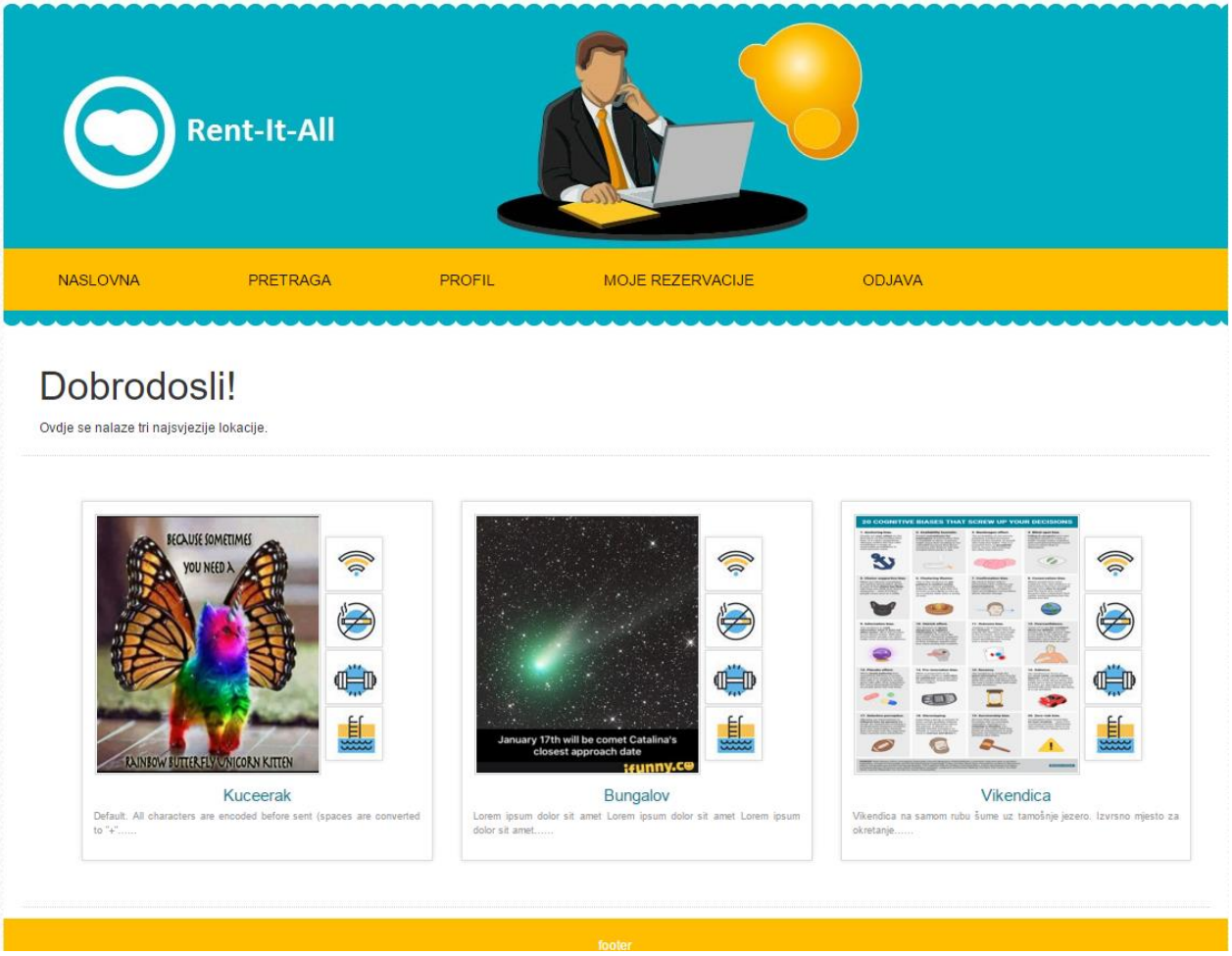
U ovom sistemu nalazi se još i skripta za odjavu korisnika. Ona sama je vrlo jednostavna jer je njezin jedini posao pokrenuti sesiju, obrisati sve iz nje te je uništiti. Jedino što treba napraviti nakon, jest preusmjeriti posjetitelja bilo gdje na stranici da bi ga skripta za autentifikaciju izbacila, ponovno, na ekran za prijavu.

3.2. Naslovnica

Naslovna stranica zajednička je svim registriranim korisnicima, bez obzira na njihovu klasu. Njezina se funkcija sastoji od dvije stvari:

1. Pokazati najrelevantnije informacije – zadnje 3 unesene nekretnine
2. Jednostavnost – radi lakšeg snalaženja

Kao što se može zaključiti prema slici 10. ona iz baze uzima posljednja tri unosa iz tablice –article'- te ih povezuje s njihovim pogodnostima iz tablice 'pogodnosti'- praveći tako pogled ćelije 1x3 gdje dominira slika zgrade,- i njezine pogodnosti u obliku ikonica u ovisnosti koje pogodnosti posjeduje. Također, svaka ćelija služi kao link na detalje te nekretnine. Još jedan, suptilniji, atribut naslovnice je i sposobnost rezanje teksta ispod slike na određen broj znakova, kako opis ne bi zatrpao naslovnicu, te kako bise osigurala ravnina ćelija.



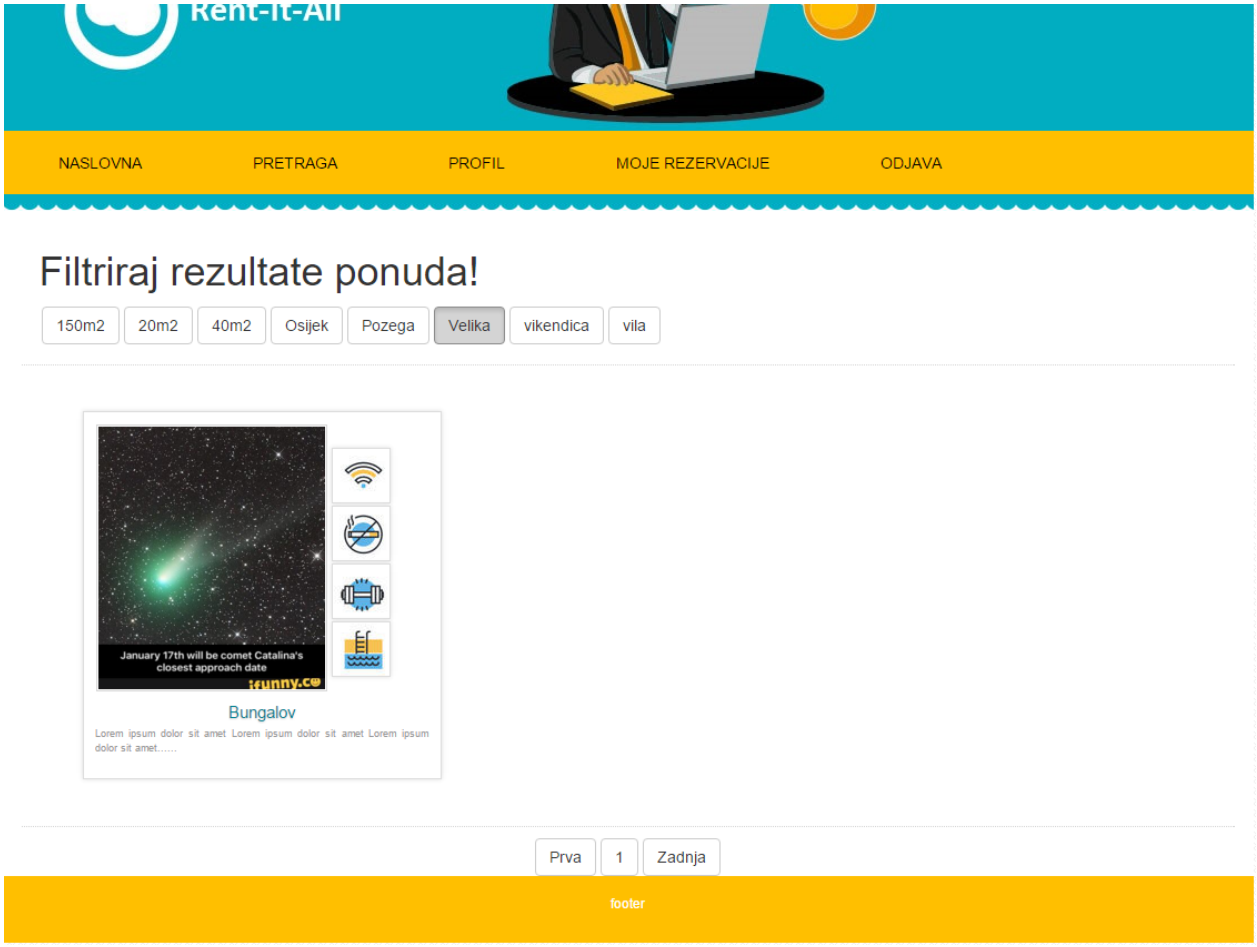
Sl. 3.4. Naslovnica

3.3. Pretraga

Iduća stranica služi za pretraživanje svih aktivnih oglasa. Zbog moguće veće količine ponuđenih nekretnina poduzete su dvije zaštitne mjere kako ne bi došlo do prenapučivanja. Mjere su sljedeće:

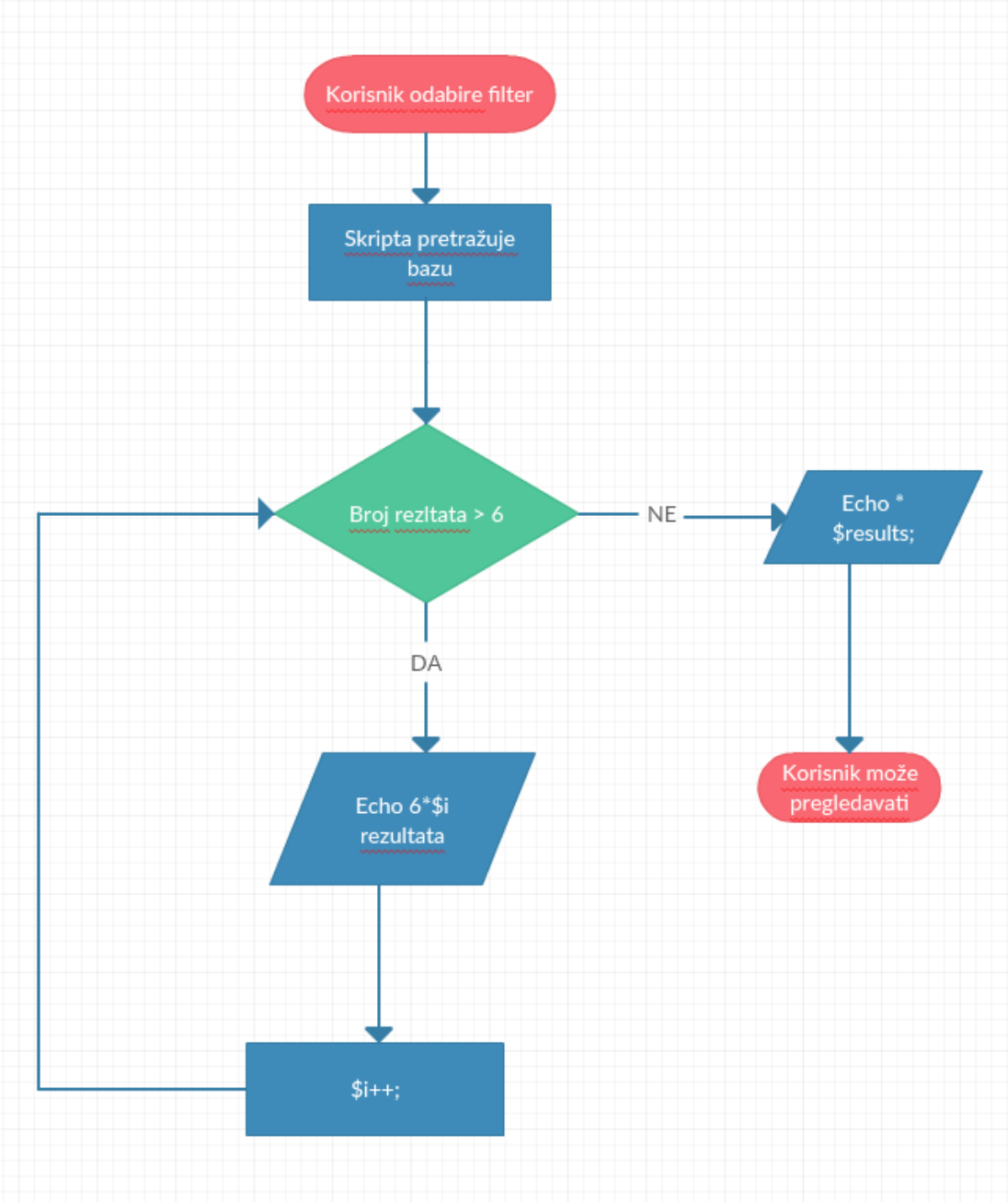
1. Sistem filtriranja –sistem koji filtrira rezultate dobivene iza baze preko ključnih riječi definiranih od strane ponuđača i spremljenih u polje tagovi.'Klikom na koji od njih korisnik može filtrirati rezultate samo na određeni tip ponuda te tako dobiti specijaliziranije rezultate pretrage
2. Sistem paginacije –sistem razdvajanja rezultata na stranice(detaljnije objašnjen u nastavku).

Dizajn stranice 'pretraga.php' sa aktivnim filterom „Velika“.



Sl. 3.5. Rezultati filtrirani na samo one koje se nalaze u blizini mjesta Velika

Iako je na slici prikazana samo jedna ponuda, postoji mogućnost prikaza navigacije na samom dnu stranice (Prva stranica, 1, Zadnja stranica) ukoliko broj rezultata premaši maksimalni zadani. Logika iza ovog sistema je, u suštini, malo globalnija petlja.



Sl. 3.6. Pojednostavljeni prikaz algoritma za paginaciju

Prvi dio algoritma nalazi se na samom vrhu dokumenta, u bloku koda koji priprema renderiranje ispisa na ekran. Tamo se umjesto selektiranja i dohvaćanja svih pogođenih unosa iz baze, dohvaća samo njihova suma. Zatim se provjerava jeli ona veća od 6. Ako nije onda se rezultati odabiru bez inicijalizacije paginacije. Ako suma rezultata jest veća od 6, tad se ona dijeli sa 6 i dobiva se broj stranica na koje će rezultati biti podijeljeni.

Tako za svaki SELECT iz baze stavlja LIMIT sa dva parametra – početak i interval.

```
if(isset($_GET['tag'])){
  $sql="SELECT * FROM articles WHERE concat(' ', tagovi, ',') LIKE '%$k%' LIMIT $start, $rez_p_page";
}elseif(!isset($_GET['tag'])){
   $sql="SELECT * FROM articles ORDER BY id DESC LIMIT $start, $rez_p_page";
}
```

Sl. 3.7. SELECT naredba sa LIMITOM

Drugi dio algoritma nalazi se na samom dnu stranice i on je zadužen za renderiranje prikaza navigacije stranicama te za uvećavanje iteratora algoritma. Svakom iteracijom ovog algoritma, prvi parametar LIMIT naredbe uvećava se za šest kako bi aplikacija znala od kojeg rezultata treba početi slagati prikaz.

```
169        </div>
170      </div>
171
172    <?php
173    if(isset($_GET['tag'])){
174      $page_sql = "SELECT * FROM articles WHERE concat(' ', tagovi, ',') LIKE '%$k%'";
175    }elseif(!isset($_GET['tag'])){
176      $page_sql = "SELECT * FROM articles ORDER BY id DESC";
177    }
178    $result = mysqli_query($db_conx, $page_sql);
179    $rows_total = mysqli_num_rows($result);
180    $pages_total = ceil($rows_total / $rez_p_page);
181
182    if(isset($_GET['tag'])){
183      $link_page="search.php?tag=" . $k . "&page";
184    }elseif(!isset($_GET['tag'])){
185      $link_page="search.php?page";
186    }
187    echo "<center><a class='btn btn-default' href='" . $link_page . "=1'>Prva</a> ";
188    for ($i=1; $i<=$pages_total; $i++) {
189      echo "<a class='btn btn-default' href='" . $link_page . "=" . $i . "'>" . $i . "</a> ";
190    };
191    echo "<a class='btn btn-default' href='" . $link_page . "=" . $pages_total . "'>Zadnja</a></center> ";
192    ?>
193
194    </section>
195
196    <footer>
```

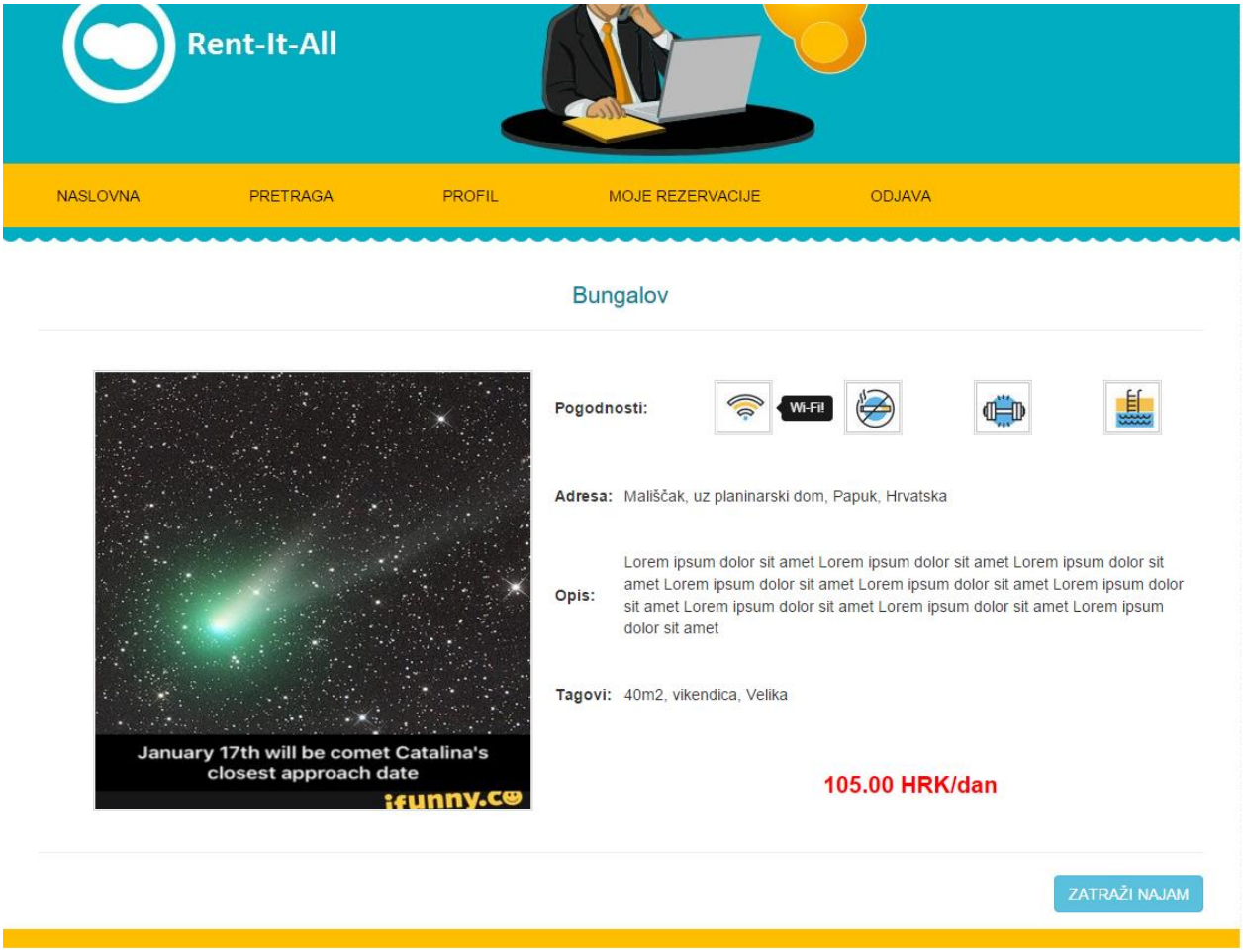
Sl. 3.8. Dio algoritma za navigaciju sistema paginacije

Klikom na bilo koju od renderiranih ponuda program preusmjerava korisnika na detalje te nekretnine i samu srž ove aplikacije.

3.4. Detalji proizvoda

Stranica s detaljima proizvod ima dvije glavne funkcije. Prva je da se detaljnije prouči oglas iznajmljivača te pregledaju sve pogodnosti i uvjeti. U suštini za to je zadužena

SELECT naredba koja vuče podatke o odabranom artiklu definiranom preko GET metode.Za povezivanje tablica koristi se obična JOIN funkcij da bi se dobili svi potrebni podaci vezani uz gledanu nekretninu.



Sl. 3.9. Prikaz detalja o ponudi

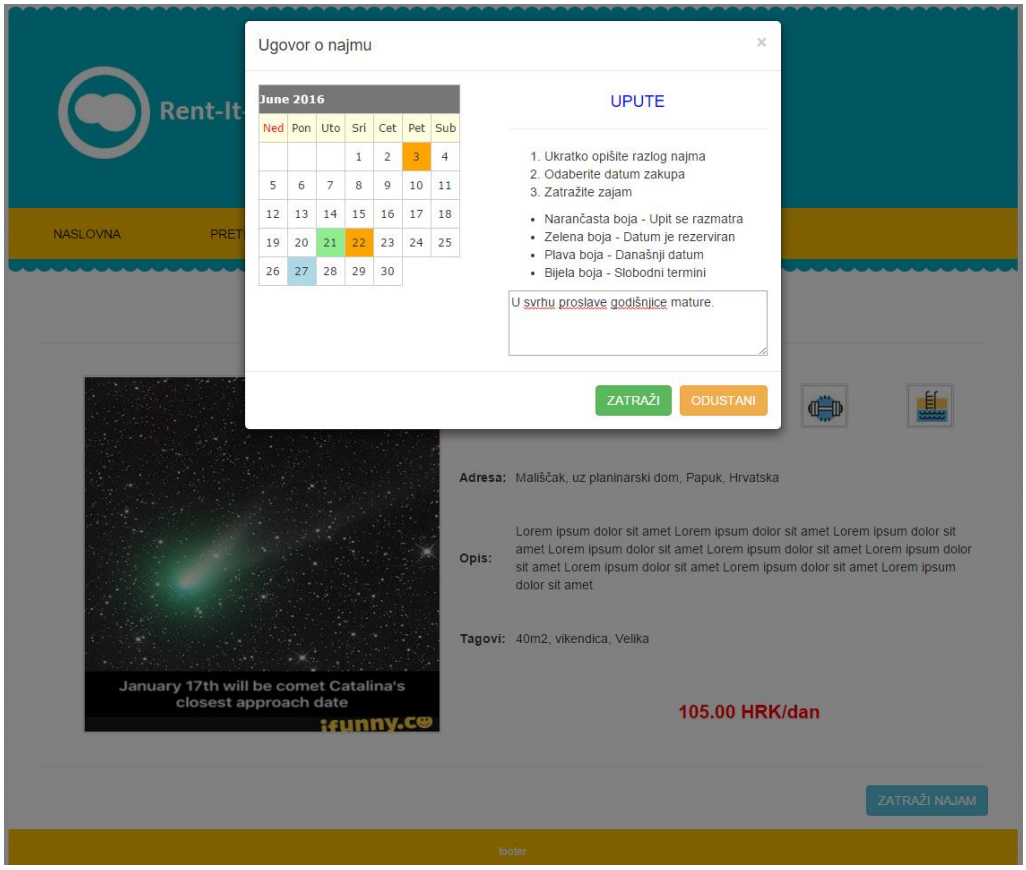
Ovdje također vrijedi napomenuti da ikonice za pogodnosti, iako nemaju fiksne opise, na :hover stanju izbacuju tooltipove s naslovom. Ova fukcionalnost je izvedena preko jQuery funkcije prikazanoj na sl. 3.10.

```
180   </footer>
181     </div>
182   <script>
183   $(document).ready(function(){
184       $('[data-toggle="tooltip"]').tooltip();
185   });
```

Sl. 3.10. jQuery funkcija za inicijalizaciju tooltipova

Druga glavna funkcija ove stranice je opcija zakupa. Klikom na dugme: Zatraži najam u donjem desnom kutu otvara se modalni dijalog sa organizacijskim kalendarom. Sam kalendar je kodiran bojama zbog lakšeg snalaženja, a legenda je dana u uputama. Skripta za generiranje kalendara nalazi se u /scripts/kalendar.php datoteci, a stilovi klase .kalendar u /style/style.css

Skripta za kalendar inicijalizira se u html dijelu /art.php datoteke unutar modalnog diva naredbom: *<**?php**include_once('scripts/kalendar.php'); **?**>* koja efektivno emulira kod iz referencirane datoteke na mjesto gdje ju se pozove. Modalni dijalog prikazan je na sl. 3.11.



Sl. 3.11. Modalni dijalog sa kalendarom

Kalendarska skripta radi tako da funkcijom *getdate()*dobije trenutno vrijeme u obliku TIMESTAMPA i onda ga razbije na komade koji joj trebaju. To se odnosi na trenutnu godinu, mjesec, dan u mjesecu te dan u godini. Preko toga locira se pozicija današnjega dana na kalendaru te broj dana u mjesecu.

```
$date = getdate();

$mday = $date['mday'];
$mon = $date['mon'];
$wday = $date['wday'];
$month = $date['month'];
$year = $date['year'];


$dayCount = $wday;
$day = $mday;

while($day > 0) {
    $days[$day--] = $dayCount--;
    if($dayCount < 0)
        $dayCount = 6;
}

$dayCount = $wday;
$day = $mday;

if(checkdate($mon,31,$year))
    $lastDay = 31;
elseif(checkdate($mon,30,$year))
    $lastDay = 30;
elseif(checkdate($mon,29,$year))
    $lastDay = 29;
elseif(checkdate($mon,28,$year))
    $lastDay = 28;

while($day <= $lastDay) {
    $days[$day++] = $dayCount++;
    if($dayCount > 6)
        $dayCount = 0;
}
```

Sl. 3.12. Seciranje TIMESTAMP objekta na elemente

Nakon toga iz tablice kalendar izvlače se svi datumi koji su rezervirani za trenutno pregledavanu nekretninu te se istom tehnikom iz tih datuma izvlače dani kojima se treba pozadina na kalendaru obojati u zeleno – vec rezervirano – i stavljaju se u niz.

Kad ima sve te informacije, skripta tada kroz petlju provlači sve datume i tako ih slaže u tablicu po danima / tjednima i efektivno crta interaktivni kalendar i svakom svojom

iteracijom uspoređuje ćeliju tablice s brojevima u nizu te tako za svaki datum posebno određuje boju pozadine.

Uz praktičan izgled, petlja također svaki datum pravi klikabilnim, koristeći <*a*>tag od kojih svaki ima *onclick*okidač koji poziva javascript funkciju koja pak taj datum sprema u skriveno polje forme za rezervaciju.

```
187  <script>
188  function get_datum(yyyy, mm, dd){
189      document.getElementById('yyyy').value = yyyy;
190      document.getElementById('mm').value = mm;
191      document.getElementById('dd').value = dd;
192      return false;
193  }
194  </script>
```

Sl. 3.13. Javascript funkcija za kopiranje odabranog datuma u skrivena polja forme

Forma se pak sastoji od 3 elementa. Prvi element skrivena su polja koja su gore navedena, a drugi je <*textarea*> element koji služi za upisivanje kratke molbe ili zahtjeva kako bi se vlasnik odlučio isplatili se dati kuću u najam podnositelju zahtjeva. Unošenje poruke je obavezno i osigurano je *required* parametrom unutar <*teaxtarea*>taga.

Klikom na dugme Zatraži pokreće se podnošenje molbe čiji je mehanizam zapravo php skripta koja je locirana na vrhu stranice. Ona provjerava je li forma ispunjena i ako je, onda kupi sve njezine elemente i, nakon što unificira datum, unosi u tablicu 'kalendar' zajedno sa porukom i identifikatorima prijavljenog korisnika i pregledavanog artikla.

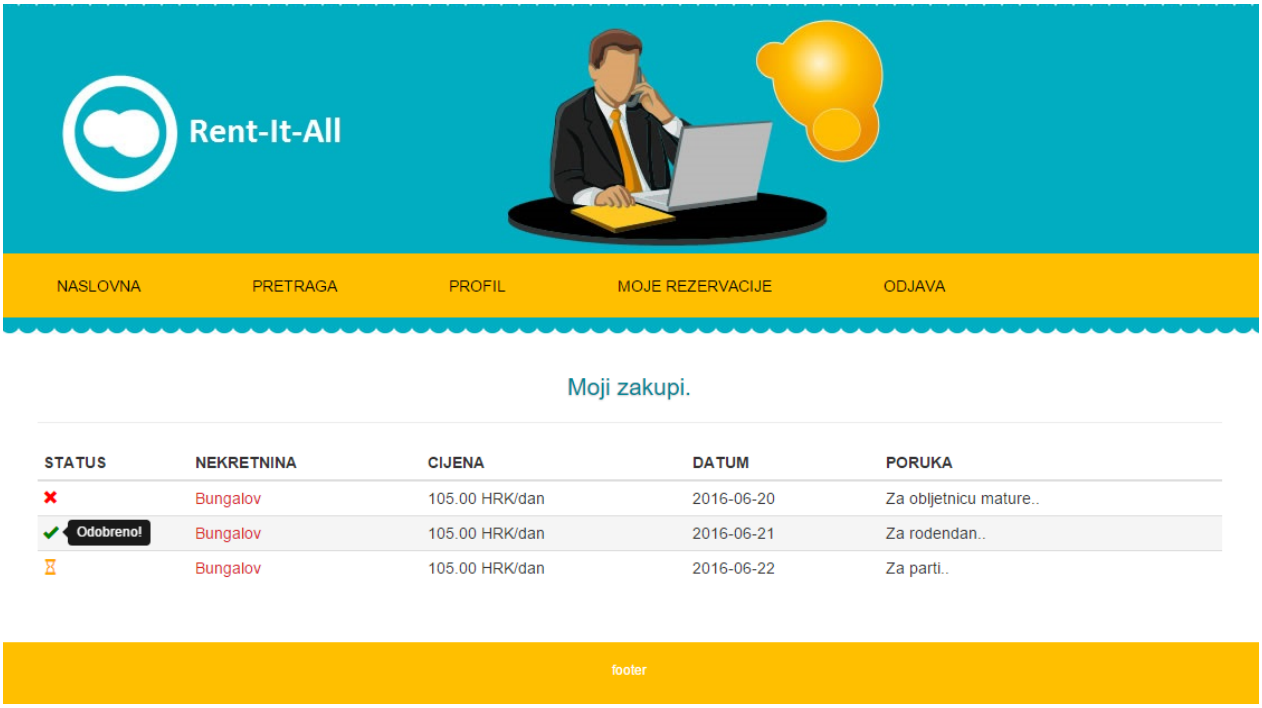
```
13  /****************************************************
14  ****** REZERVIRAJ
15  ****************************************************/
16
17  if(isset($_POST['okok'])){
18      $dd     = mysqli_real_escape_string($db_conx, $_POST['dd']);
19      $mm     = mysqli_real_escape_string($db_conx, $_POST['mm']);
20      $yy     = mysqli_real_escape_string($db_conx, $_POST['yyyy']);
21      $po     = mysqli_real_escape_string($db_conx, $_POST['poruka']);
22      $kor    = $_SESSION["id"];
23      $dat    = $yy . '-' . $mm . '-' . $dd;
24      $sql    ="INSERT INTO kalendar (art_id, kupac_id, datum, pending, poruka) VALUES ('$id', '$kor', '$dat', 0, '$po')";
25      mysqli_query($db_conx, $sql);
26      header('location: art.php?id=' . $id);
27      exit();
28  }
29
```

Sl. 3.14. Sistem rezervacije

Prije kraja vrijedi i napomenuti da zgrada upitom za rezervaciju nije automatski odobrena, već to treba učiniti vlasnik nekretnine.

3.5. Rezervacije

Stranica s rezervacijama mjesto je gdje potencijalni zakupci mogu vidjeti statuse svih svojih zamolbi. Na slici 3.15. nalazi se prikaz zamolbi trenutnog korisnika razvrstanih po statusu.



Sl. 3.15. Popis svih do sada poslanih molbi

Stanje svake pojedinačne zamolbe prikazano je ugođajno obojanim glif-ovima koji su dobiveni u paketu sa bootstrapframeworkom. Glif-ovi također, kao i sve druge ikonice, imaju jquerytooltiove na :hover stanju pa se razlikuju po boji, imenu i naslovu. To redom glasi :

1. Crveni X – zahtjev je odbijen
2. Zelena kvačica – zahtjev je prihvaćen
3. Narančasti pješčani sat – zahtjev je još uvijek na razmatranju

Naziv nekretnine ujedno služi i kao link na detalje o tom artiklu.

Skripta kojom se aplikacija služi za generiranje ispisa za ovu stranicu nalazi se na sl. 3.16., a pokazuje kako aplikacija koristi JOIN da bi spojila rezultate iz dviju tablica te dobivene rezultate koristila da bi preko *while* petlje generirala ispis u tablicu na stranici.

```
11  /****************************************************
12  ****** RENDER
13  ****************************************************/
14  $rend="";
15  $i = 1;
16  $kid=$_SESSION['id'];
17  $sm="
18      SELECT
19          a.id      AS ID,
20          a.naziv   AS IME,
21          a.cijena  AS HRK,
22          k.datum   AS DAT,
23          k.pending AS PEN,
24          k.poruka  AS POR
25      FROM
26          articles a,
27          kalendar k
28      WHERE
29          k.art_id=a.id
30          AND
31          k.kupac_id='$kid'
32      ";
33  $op=mysqli_query($db_conx, $sm);
34  while($row=mysqli_fetch_array($op, MYSQLI_ASSOC)){
35      $id     =$row['ID'];
36      $ime    =$row['IME'];
37      $cijena =$row['HRK'];
38      $datum  =$row['DAT'];
39      $pending=$row['PEN'];
40      $poruka =$row['POR'];
41
42      $rend .= '
43      <tr>
44          <td>' . $pen . '</td>
45          <td><a href="art.php?id=' . $id . '">' . $ime . '</a></td>
46          <td>' . $cijena . ' HRK/dan</td>
47          <td>' . $datum . '</td>
48          <td>' . $poruka . '.</td>
49      </tr>
50      ';
51      $i++;
52  }
53  ?>
```

Sl. 3.16. Mehanizam za renderiranje molbi

3.6. CRUD oglasa

Sve što je do sada obrađeno bilo je isključivo iz pogleda klijenata; iznajmljivači, s druge strane, imaju i neke dodatne mogućnosti koje će sada obraditi.

Najočitija razlika između iznajmljivača i klijenata jest ta što iznajmljivači imaju mogućnost iznajmljivati svoje nekretnine te je njima u izborniku dodana mogućnost pristupa formi za unošenje oglasa.

Ispod forme nalazi se lista objavljenih oglasa trenutno prijavljenog oglašivača (pod uvjetom da ima neku objavljenu), kao i CRUD mogućnosti za ažuriranje i brisanje.

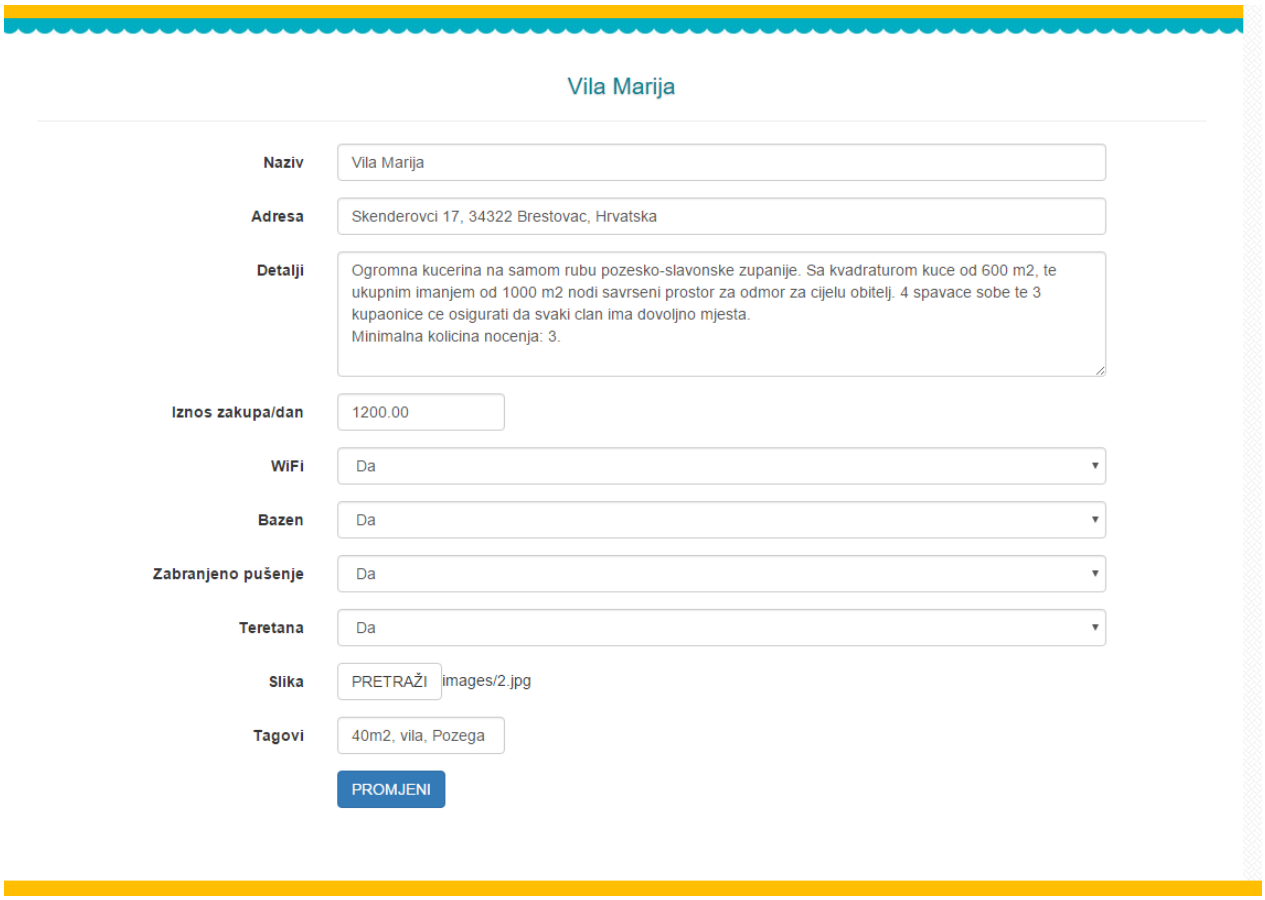
Ako postoji zanimanje ljudi za neku od oglašenih nekretnina, uz njeno će ime također biti mogućnost pregledavanja svih poslanih zamolbi za unajmljivanjem klikom na dugme Ponude. Skripta broji količinu zaprimljenih zamolbi te ih ispisuje iz naslov dugmeta.

Sl. 3.17. Lista objavljenih oglasa trenztno prijavljenog iznajmljivača

Klikom na dugme Uredi otvara se forma za uređivanje oglasa koja je dizajnerski identična onoj za unos, s tom razlikom da je ova već ispunjena. To funkcionira na način da se iz baze SELECT naredbom dohvate sve informacije vezane za odabrani proizvod i dodijele svojim varijablama.

Nakon toga se unutar html forme, u *value* parametrima tagova upiše ***<?phpecho*** *$ime_odredene_variable;* ***?>*** te se prilikom učitavanja stranice ti elementi napune podacima.

Sl. 3.18. prikazuje dizajn forme za uređivanje na konkretnom primjeru sa svim dohvaćenim podacima.



Sl. 3.18. Forma za uređivanje

Ovaj dio aplikacije isto tako koristi i jednu malu, ali izrazito moćnu javascript skriptu za redizajn default-nog dugmeta za upload datoteka, koji ova aplikacija koristi za dizanje slika oglasa. Da bi cijela forma bila u skladu sa dizajnom, defaultno dugme je 'prekriveno' **tagom klase *.btn*. Ovaj jQuery komad koda ima za prikazivanje imena učitane datoteke – slike, odnosno njezine adrese u ovom slučaju.

```
<script>
    $(document).on('change', '.btn-file :file', function() {
        var input = $(this),
        numFiles = input.get(0).files ? input.get(0).files.length : 1,
        label = input.val().replace(/\\/g, '/').replace(/.*\//, '');
        input.trigger('fileselect', [numFiles, label]);
    });
    $('.btn-file :file').on('fileselect', function(event, numFiles, label) {
        $( "#sl" ).text( label );
    });
</script>
```

Sl. 3.19. Javascriptlabel funkcija

4. ZAKLJUČAK

Uspješnim testiranjem ove aplikacije došlo se do kraja ovoga završnog rada te je kroz taj proces obrađen način nastanka kompletne aplikacije od početka do kraja. Prije svega baza podataka, ali dizajn aplikacije, njezina logika i arhitektura svi su elementi zajedno ono što je ovaj projekt učinilo izazovnim.

Web aplikacija za vođenje evidencije iznajmljivanja objekata jest na prvi pogled relativno linearan projekt, ali kroz njenu izradu došlo se do zaključka da je za funkcioniranje aplikacije zaduženo jako puno, ljudima van struke, nevidljivih komadića radne logike bez kojeg aplikacija vjerojatno ne bi funkcionirala.

Baza podataka se nameće kao prvak u tom pitanju. Zbog lakoće kojom se tablice u njoj izrađuju, programer se može uljuljati u lažni osjećaj sigurnosti te bi, nakon samo par sati pisanja aplikacije, uvidio da baza koju je napravio fali na nekom dijelu pa je potrebno vraćanje kako bi se ona prepravila. Logika bi, u tom slučaju, vjerojatno bila isto tako jako blizu.

No, tijekom pisanja došlo se do još jednog zaključka, a to je da je mogućnost pisanja cijelog projekta, kompletno od nule, vrlo važna programerima jer im daje uvid u sve kutove njihovih projekata, kao i one koji oni inače ne rade, a sada su prisiljeni.

5. LITERATURA

[1] http://php.net/ 25.05.2016

[2] http://www.w3schools.com/ 27.05.2016

[3] https://www.codecademy.com/ 30.05.2016

SAŽETAK

Zadatak ovog rada jest napraviti web aplikacija za vođenje evidencije iznajmljivanja objekata. Trebalo je isplanirati proces njene izrade, postaviti temelje u bazi podataka i arhitekturi datoteka. Također je trebalo predvidjeti mogućnost zadavanja vrste i svrhe objekta, različitih uloga i prava pristupa korisnika te vođenje detaljne evidencije iznajmljivanja objekta.

Nakon logičkog dijela trebalo se posvetiti i korisničkom iskustvu te dizajnirati interraktivno web sučelje za pristup svim mogučnostima aplikacije te razne manipulacije bazom podataka.

Ključne riječi: aplikacija, objekt, iznjmljivanje, web, korisnici

ABSTRACT

Web application for keeping records of renting facilities

The premise ofthispaperwas to create web application for keeping track of object renting traffic. It was required to preplan whole process of creating such an app, design its data base and folder arhitecture. Also, there was a question of different user levels with different privilegies and access rig hts and keeping detailed reports on past rentings of some object.

After the programs logic was completed, next on the agenda was to design an interactive web user interface with all necessery features of application and for data base manipulation.

Keywords:web, app, object, rent, user

ŽIVOTOPIS

Božo Knezović rođen je 20.08.1993.godine u Zagrebu, RH. Završio je osnovnu školu Grigor Vitez u Osijeku 2008.godine te iste upisao Graditeljsko-geodetsku školu Osijek koju je 2012.godine završio. Građevinski fakultet upisuje u akademskoj godini 2012/13, gdje je bio potpredsjednik studetskog zbora Građevinskog fakulteta, prebacuje se na Geodetski fakultet u Brčkom te Elektrotehnički fakultet Osijek akademske 2013/14 godine. U akademskoj godini 2015/16. izabran kao predstavnik ETF-a na Sveučilišnom zboruJ.J.Strossmayer u Osijeku. Radno iskustvo se sastojalo od geodetskog posla u Altitudid.o.o u trajanju od 4 mjeseca, studentskih poslova, a trenutno u Ezop antikvarijatu. Koristi se engleskim kao stranim jezikom te od znanja na računalu spadaju Word, Excel,PPT, AutoCAD, te programski jezici (HTML,CSS, C++,…).