

# IZRADA INTERNETSKE APLIKACIJE ZA NAJAM SMJEŠTAJA

---

**RAJIČ, ANTE**

**Undergraduate thesis / Završni rad**

**2018**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Split / Sveučilište u Splitu**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:228:538276>

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

*Download date / Datum preuzimanja:* **2024-07-10**



*Repository / Repozitorij:*

[Repository of University Department of Professional Studies](#)



**SVEUČILIŠTE U SPLITU**  
**SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE**

Preddiplomski stručni studij Informacijska tehnologija

**ANTE RAJIČ**

**ZAVRŠNI RAD**

**IZRADA INTERNETSKE APLIKACIJE ZA NAJAM  
SMJEŠTAJA**

Split, prosinac 2018.

**SVEUČILIŠTE U SPLITU**  
**SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE**

Preddiplomski stručni studij Informacijska tehnologija

**Predmet:** Oblikovanje web stranica

**Z A V R Š N I R A D**

**Kandidat:** Ante Rajič

**Naslov rada:** Izrada internetske aplikacije za najam smještaja

**Mentor:** Haidi Božiković, viši predavač

Split, prosinac 2018.

## Sadržaj

Sažetak.....	1
Summary.....	1
1 Uvod.....	2
2 Tehnologije.....	3
2.1 HyperText Markup Language.....	3
2.2 Cascading Style Sheets .....	4
2.3 JavaScript.....	6
2.4 jQuery .....	7
2.5 Bootstrap .....	8
2.6 Hypertext Preprocessor .....	10
2.7 XAMPP.....	11
2.8 My Structured Query Language .....	12
3 Opis praktičnog rada .....	13
3.1 Funkcionalnosti aplikacije .....	13
3.2 Korisničko sučelje.....	14
3.2.1 Upotreba aplikacije od strane korisnika .....	14
3.2.2 Upotreba aplikacije od strane administratora .....	20
3.3 Implementacija funkcionalnosti aplikacije .....	21
3.3.1 Implementacija baze podataka.....	21
3.3.2 Postupak rezervacije smještaja .....	22
3.3.3 Sustav za registraciju i prijavu korisnika.....	29
3.3.4 Administratorski i korisnički profil .....	33
3.3.5 Kontakt forma.....	40
3.3.6 Dizajn .....	41
4 Zaključak.....	44
5 Literatura .....	45

## Sažetak

Izrada internetske aplikacije za najam smještaja je napravljena da se pojednostavni proces rezervacije smještaja, olakša upravljanje i organizacija objekta koji se iznajmljuje. Aplikacija je prilagodljiva mobilnim uređajima.

Zadatak aplikacije je bio ubrzati proces rezerviranja smještaja, omogućiti korisniku prikaz rezervacija i upravljanje istim, dok je za vlasnika osim prikaza i upravljanja rezervacijama bio cilj omogućiti korigiranje cijene za buduće rezervacije.

Aplikacijom se želi riješiti problem ručnog upisa rezervacija gdje je veća mogućnost za stvaranje pogreške. Vođenje evidencije rezervacija nalazi se na jednom mjestu bez potrebe zapisivanja podataka na papir. Korisnik može u svega par minuta odabrati termin i željeni apartman. Na ovaj način, najam smještaja se ubrzao i uvelike pojednostavnio.

## Summary

### **Development of web application for accommodation renting**

The rental application is designed to simplify the process of booking accommodation, facilitating the management and organization of the rented facility. The application is made with a responsive design.

The task of the application was to speed up the accommodation reservation process, to allow user to view and delete the selected items, and for the owner to be able to track upcoming reservations and correcting the price for future bookings.

Also, the goal of application is to solve the problem of manual entry of reservations where there is a greater chance of making a mistake. Keeping the booking records is in one place without the need to write data on paper. In just a few minutes the user can choose the time and the desired apartment. In this way, renting accommodation accelerated and greatly simplified.

## 1 Uvod

Aplikacija prilagodljiva mobilnim uređajima olakšava rezervaciju smještaja za najam. Cilj aplikacije je stvoriti jednostavan i siguran pristup za rezervaciju smještaja. Korisnik ima mogućnost ispisa vlastitih rezervacija, te mogućnost brisanja istih u slučaju da ih korisnik želi otkazati. Isto tako napravljen je administratorski dio u kojem administrator ima mogućnost brisanja nadolazećih rezervacija uz obavijest samog korisnika pute e-maila o istom te mijenjanje trenutne cijene najma.

Pri izradi aplikacije je korišteno više različitih tehnologija zajedno, sa svrhom stvaranja boljeg korisničkog iskustva i funkcionalnosti aplikacije. Podaci u aplikaciji se ažuriraju prilikom bilo kakvih izmjena, stoga su promjene vidljive odmah.

Za funkcionalnost aplikacije u pozadini korištene su tehnologije *My Structured Query Language* (MySQL) i *Hypertext Preprocessor* (PHP) koje omogućuju strukturiranje baze i pristup njenim podacima. Tako se ostvaruje jednostavnija i bolja interakcija s korisnikom. Razlog za odabir ovih tehnologija je što su među najkorištenijim za izradu internetskih (engl. *web*) aplikacija. Obe tehnologije su otvorenog kôda (engl. *open-source*), što znači da je kôd dostupan javnosti na uvid, korištenje, izmjene i daljnje dijeljenje.

Tehnologije korištene za razvoj korisničkog sučelja su *HyperText Markup Language* (HTML), *Cascading Style Sheets* (CSS) i razvojni okvir CSS (engl. *framework*) Bootstrap. Bootstrap omogućuje pojednostavljeno razvijanje aplikativnog sučelja. Kreiran je sa svrhom da omogući prilagodljivost aplikacija svim veličinama zaslona, a sam sadržaj da se dinamički mijenja s obzirom na njegovu veličinu.

U trećem poglavlju su pobliže opisane korištene tehnologije, a u četvrtom poglavlju je opisana sama funkcionalnost aplikacije, metoda kojom je ostvaren odabir željenih datuma, odabir željenog smještaja i konačna rezervacija. Prikazana je implementacija korisničkog i administratorskog sučelja i njihove mogućnosti. Uz same opise priloženi su primjeri kôda i objašnjenja na koji način dijelovi aplikacije funkcioniraju.

## 2 Tehnologije

### 2.1 HyperText Markup Language

HTML je prezentacijski jezik uz pomoć kojeg se kreiraju hipertekstualne datoteke [1]. Preglednik (engl. *browser*) čita sadržaj i strukturu datoteke te od tih podataka oblikuje stranicu. HTML služi isključivo za prezentaciju stranice, njime se ne mogu izvršavati kao na primjer jednostavne računske operacije zbrajanja i oduzimanja.

Zadnja verzija HTML5 je donijela nove mogućnosti i promjene. Prethodna verzija HTML4 i *Extensible Hypertext Markup Language* (XHTML) sada su povezane u jedan prezentacijski jezik gdje se mogu koristiti obje sintakse. Unaprijeđena je tehnologija za reprodukciju video sadržaja na internetskim stranicama bez potrebe za dodatnim alatima kao što su *Adobe Flash Player* ili *Microsoft Silverlight*. *Document Object Model* (DOM) i *Application Programming Interface* (API) se smatraju ključnim dijelovima HTML5 specifikacije.

Uvedene su nove mogućnosti kao što je reprodukcija glazbe upotrebom oznake `<audio>` ili kreiranje crteža uz pomoć oznake `<canvas>`. Isto tako već postojeći elementi iz prijašnjih verzija su redefinirani i poboljšani, na primjer `<a>`, `<menu>` i `<select>`. Korištenjem HTML5 omogućio se brz i jednostavan razvoj mobilnih internetskih aplikacija.

Neki od novih elemenata koji su uvedeni u HTML5:

- `<header>` - predstavlja spremnik za uvodni sadržaj
- `<footer>` - definira zaglavlje dokumenta
- `<nav>` - koristi se za definiranje navigacije
- `<section>` - označava određena područja u dokumentu

HTML dokument sastoji se od dva dijela: zaglavlja (engl. *header*) i tijela (engl. *body*). Zaglavlje se definira oznakama (engl. *tag*) `<head></head>`, a tijelo oznakama `<body></body>`. Informacije napisane u zaglavlju služe za opis stranice, one se ne prikazuju u pregledniku, dok sadržaj koji se nalazi u tijelu dokumenta se vidi na stranici uključujući razne fontove, boje, slike itd.

Za kreiranje minimalne HTML5 stranice potrebno je definirati deklaraciju `!DOCTYPE` i znakovni prikaz. Kako to izgleda vidi se u ispisu 1.

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML5 Hello World</title>
    <meta charset="utf-8">
  </head>
  <body>
    <p>Hello World!</p>
  </body>
</html>
```

### Ispis 1: Predložak HTML stranice

## 2.2 Cascading Style Sheets

CSS je stilski jezik uz pomoć kojeg se oblikuju elementi HTML-a, odnosno kako će elementi izgledati, na kojoj poziciji će se nalaziti element i slično definiraju stilovi CSS-a [1]. S obzirom da se danas za pristup internetskim stranicama sve više koriste pametni telefoni, vrlo je važno prilagoditi sadržaj uređajima sa manjim zaslonima. Zato se javila potreba za uvođenjem nove CSS3 specifikacije.

Nova verzija donijela je niz novih mogućnosti kao što su medijski upiti (engl. *media queries*) koji omogućuju mijenjanje kompozicije stranice s obzirom na veličinu zaslona. Omogućena je izrada raznih animacija za elemente HTML-a.

CSS se može rabiti na više načina. Najčešći način je pozivanje dokumenta u zaglavlju stranice unutar elementa `<style>`, što se vidi na ispisu 2.

```
<link rel="stylesheet" type="text/css" href="style.css">
```

### Ispis 2: Poziv na CSS datoteku u zaglavlju dokumenta

Drugi način je pisanje unutar elementa HTML-a, što se vidi na ispisu 3.

```
<p style="font-weight: bold">Tekst je podebljan.</p>
```

### Ispis 3: Linijski CSS unutar HTML elementa



Sintaksa CSS-a je jednostavna i lako se pamti. Selektor (engl. *selector*) se koristi za dohvaćanje elemenata HTML-a i njegov naziv ne smije započinjati brojkom. Svojstva koja se primjenjuju na element pišu se unutar vitičastih zagrada: `selector{ property: value; }`. Može se selektirati i više različitih elemenata u jednom koraku tako da svaki element pojedinačno dobije istu vrijednost: `p, a, div{ color: red; }`.

CSS omogućava da se različitim elementima HTML-a mogu dodijeliti različiti definirani stilovi, pa se CSS zove kaskadnim stilskim jezikom. Pravila po kojima element poprima svojstva kreću se od općenitih pa do preciznijih.

```
<p>Tekst</p>
<p class="different">Tekst</p>
p { font-size: 12px; font-weight: bold; }
```

#### Ispis 4: Primjer općenitog stila

U ispisu 4 je definiran stil u kojem oba elementa poprimaju iste vrijednosti, veličinu slova i podebljan tekst. Ako se želi postaviti drugačija vrijednost određenom elementu tada se definira precizniji stil.

```
p.different { color: red; }
```

#### Ispis 5: Kaskadnost u CSS-u

U ispisu 5 vidljiva je kaskadnost. Tekst elementa sa klasom `.different` dobiva vrijednost crvene boje, ali i vrijednosti iz prethodno definiranog stila.

Elementi HTML-a se opisuju klasama i identifikatorima. Klasa služi za definiranje brojevnih i generičkih članova, a identifikator se koristi za definiranje jedinstvenih i pojedinačnih elemenata. Klasi se pristupa navođenjem `.` prije naziva, a identifikatoru znakom `#` prije naziva. Klase i identifikatori su osjetljivi na velika i mala slova te mogu sadržavati podvučene crte i alfa-numeričke znakove. U ispisu 6 se vidi primjer korištenja klase i identifikatora.

```
<p class="first">Tekst je nakošen.</p>
<p id="second">I ovaj tekst je nakošen.</p>
.first { font-style: italic; }
#second { font-style: italic; }
```

### Ispis 6: Primjer korištenja klase i identifikatora

## 2.3 JavaScript

JavaScript je skriptni jezik, a to znači da se kôd izvršava izravno bez prevođenja na binarni jezik [2]. Službeni naziv jezika je ECMA-262, a zadnja verzija objavljena je u 2018. godini pod nazivom ECMAScript 2018. Uz HTML i CSS, JavaScript je jezik koji svi mrežni razvojni programeri moraju znati. Lako se uči te je među najpopularnijim skriptnim jezicima današnjice kojeg podržavaju svi poznatiji preglednici. Omogućuje korištenje skripti koje se pokreću uz pomoć preglednika od strane klijenta (engl. *client-side*). S obzirom da podržava funkcionalno programiranje, pomoću JavaScript-a mogu se izrađivati video igre i desktop aplikacije. Omogućava i postavljanje kolačića (engl. *cookies*) u koje se spremaju informacije o korisniku. Unutar jezika je implementiran JavaScript DOM koji omogućava jednostavno dohvaćanje elemenata HTML-a i upravljanje njihovim izgledom, atributima i sadržajem. DOM objekte preglednik pruža jeziku JavaScript. Koristeći JavaScript korisničko sučelje može biti bogatije i ugodnije jer se mogu kreirati elementi koji reagiraju na korisnikove akcije pomoću miša ili tipkovnice. Točnost podataka se može provjeriti i prije slanja stranice na poslužitelj što olakšava promet i smanjuje dodatnu komunikaciju s poslužiteljem.

JavaScript s HTML dokumentom se može povezati na 3 načina: pomoću interne skripte, vanjskog povezivanja i unutar elementa. Načinom interne skripte kôd se piše unutar elementa `<script>` koji se može nalaziti unutar elementa `<head>` ili `<body>`.

Pri načinu vanjskog povezivanja potrebno je elementu `<script>` dodati atribut `src` koji sadrži informaciju na kojoj se lokaciji nalazi potrebna JavaScript datoteka. Primjer se nalazi u ispisu 8.

```
<script type="text/javascript" src="functions.js"></script>
```

### Ispis 8: Vanjsko povezivanje sa JavaScript datotekom

JavaScript kôd se može pisati i unutar oznaka HTML-a, na način da se elementu kao vrijednost doda funkcija JavaScript-a. Primjer se nalazi u ispisu 9.

```
<h1 onmouseover = "alert('Prelaskom miša preko elementa pokreće se kôd.')" = "">Hello world!</h1>
```

### Ispis 9: JavaScript unutar elementa HTML-a

## 2.4 jQuery

jQuery je najpopularnija biblioteka jezika JavaScript. Omogućava brže pisanje JavaScript kôda pa je s tim puno brža izrada internetskih stranica [2]. Jednom linijom jQuery kôda postigne se isti rezultat kao s deset linija običnog JavaScript kôda. Omogućava mijenjanje sadržaja, pristupanje raznim dijelovima stranice, dodavanje animacija, reagiranje na korisnikovu aktivnost kao što su klik ili prijelaz mišem preko nekog elementa. Još jedna velika prednost je to što je potpuno kompatibilan sa svim internetskim preglednicima.

jQuery omogućava jednostavan pristup DOM objektima i kreiranje dinamičnih *Asynchronous JavaScript And XML* (AJAX) elemenata. Kod klasičnih internetskih aplikacija gdje korisnici izvode različite radnje, potrebno je kompletno učitavanje cijele stranice sa svim podacima u njoj. Upotrebom AJAX tehnologije taj problem se rješava tako da se osvježava samo određeni dio stranice, dok se ostatak ne mijenja. Na taj način aplikacija se puno brže izvršava i poboljšava se sveukupno korisničko iskustvo.

jQuery nudi ulančavanje metoda u jednom nizu. Element se može pronaći putem CSS selektora, na jedan skup elemenata povezati više metoda i slično. Prednosti koje se nude korištenjem jQuery biblioteke su: jednostavna upotreba, široka dokumentacija raznih autora, podrška za AJAX, kompatibilnost sa svim preglednicima te velik broj dodatnih ekstenzija.

jQuery je potrebno uključiti u HTML kako bih ga mogli koristiti. Uključivanje se vrši preuzimanjem datoteke sa službene internetske stranice te pozivanjem u elementu `<body>` ili putem distribuirane mreže poslužitelja u svijetu (engl. *Content Distribution Network*). Kod pozivanja obje datoteke unutar elementa `<body>` potrebno je u `src` atributu navesti punu putanju. Primjer se nalazi u ispisu 10.

```
<script type="text/javascript" src="jquery.js"></script>
```

### Ispis 10: Poziv jQuery datoteke

Drugi način se češće koristi a prednost je veća brzina i spremanje (engl. *caching*) podataka. Primjer takvog poziva se nalazi u ispisu 11.

```
<script  
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">  
</script>
```

### Ispis 11: Poziv jQuery datoteke putem distribuirane mreže korisnika

Pristup elementima vrši se preko funkcije jQuery() ili skraćeno \$(). Kao parametar funkciji se proslijeđuje *string* te ona prolazi kroz cijeli DOM i locira element. Primjer dohvaćanja elementa kojemu se dodaje CSS svojstvo klase se nalazi u ispisu 12.

```
$('#div').addClass('klasa1');
```

### Ispis 12: Dohvaćanje elementa kojemu se dodaje CSS svojstvo klase

## 2.5 Bootstrap

Bootstrap je trenutno najpopularniji okvir za izradu internetskih stranica, kombinacija je elemenata HTML-a, stilskog jezika CSS i jezika JavaScript [4]. Zadnja stabilna verzija je 4.1.3. Razvijen je od kompanije Twitter, a osmišljen je da olakša razvoj internetskih aplikacija i njenih naprednih komponenti. Sadrži sve osnovne elemente koji su potrebni za izgled moderne internetske stranice: botuni, navigacije, tablice, forme, tekstualna polja. Izgled je prilagodljiv veličini zaslona (engl. *responsive design*), gdje su elementi potpuno dinamični i prilagodljivi. Bootstrap nudi dodatnu kolekciju ugrađenih komponenti kao što su modalni prozori, animacije, izbornici, različiti izmjenjivači sadržaja (engl. *slide show*). Jedan od najvažnijih elemenata okvira je mreža (engl. *grid*), koja se koristi za raspored elemenata na stranici. Kompletan sadržaj internetske stranice se smješta unutar elementa s klasom `.container` koji ima definiran izgled prilagodljiv veličini zaslona u pikselima ili u `.container-fluid` čiji je izgled prilagodljiv veličini zaslona definiran u

postotcima. Redovi služe kao omotači za stupce koji se nalaze unutar tih redova. Stupci (engl. *column*) imaju između sebe razmak koji se koristi za bolju preglednost te se u njih sprema sadržaj stranice.

Predefinirane klase uz pomoć kojih se stvara brz raspored mrežnih elemenata su `.col-sm-*` i `.row`. Stupci se kreiraju specificiranjem jednim od dvanaest mogućih stupaca. Ako se žele napraviti tri stupca koja će zauzeti cijelu širinu reda, to se radi kreiranjem tri elementa koji imaju naziv klase `.col-sm-4`. Primjer takvog rasporeda se nalazi u ispisu 13.

```
<div class="col-sm-4">...</div>
<div class="col-sm-4">...</div>
<div class="col-sm-4">...</div>
```

### Ispis 13: Primjer rasporeda koristeći Bootstrap klase

Okvir Bootstrap koristi predefinirane medijske upite za postizanje dizajna prilagodljivog veličini zaslona. Mogu se napraviti i prilagođeni (engl. *custom*) medijski upiti, jedan takav primjer se nalazi u ispisu 14.

```
@media only screen and (max-width: 600px)
{ body { background-color: lightblue; }
}
```

### Ispis 14: Prilagođeni medijski upit za postizanje prilagodljivog dizajna

Za korištenje okvira Bootstrap potrebno je preuzeti njegov izvorni kôd sa službene stranice u kojem se nalaze CSS, JavaScript te font datoteke. Elementu `<link>`, koji se nalazi unutar elementa `<head>`, dovoljno je postaviti putanju do `bootstrap.min.css` datoteke. Primjer se nalazi u ispisu 15.

```
<link rel = 'stylesheet' href = 'bootstrap.min.css'>
```

### Ispis 15: Poziv na Bootstrap CSS datoteku

Nakon toga je potrebno postaviti preostale dvije datoteke koje se nalaze unutar elementa za konačno korištenje okvira. Primjer se nalazi u ispisu 16.

```
<script src="jquery-3.3.1.js"></script>
<script src="bootstrap.min.js"></script>
```

### Ispis 16: Poziv na Bootstrap JavaScript datoteke

## 2.6 Hypertext Preprocessor

PHP jedan je od najraširenijih skriptnih jezika koji se izvodi na poslužiteljskoj strani, a koristi se za stvaranje interaktivnih internetskih stranica [3]. Kôd se vrlo lako umetne unutar HTML dokumenta i na taj način se kreira dinamička stranica.

Objektno je orijentiran i kompatibilan je s većinom operativnih sustava. Za povezivanje s drugim poslužiteljima koristi *Server Application Programming Interface* (SAPI). SAPI je mehanizam koji kontrolira komunikaciju između jezika PHP i ostalih poslužitelja.

PHP dokument započinje unutar elementa `<?php ?>` ili kraća verzija `<? ?>`, međutim preporuča se korištenje prethodne verzije radi jasnijeg prikaza. Primjer se nalazi u ispisu 17.

```
<body>
  <div>
    <?php
      echo "Hello World!";
    ?>
  </div>
</body>
```

### Ispis 17: PHP primjer ispisa „Hello World!“

PHP ima sličnu sintaksu kao jezici Perl, C i Java. Imena varijabli započinju znakom \$, a sljedeći znak je slovo ili podvučena crta. Kao i kod gore navedenih jezika koristi petlje za kontrolu toka programa. Varijable se u PHP mogu deklarirati bilo gdje u PHP bloku i ne

definiraju po tipu podatka. Ako je varijabla definirana i nije joj dodijeljena vrijednost, ona se smatra *variant* to jest bilo koji tip podatka.

Kôd se piše na početku dokumenta radi bolje preglednosti. PHP prepoznaje tip podatka spremljen u varijablu u trenutku kad joj se podatak pridjeljuje, što znači da je PHP je slobodno pisan jezik (engl. *loosely typed*). Drugi tipovi podataka koji postoje su *string*, *array*, *object*, *null*, *boolean*, *integer* i *float*.

U PHP-u se koriste i konstante koje se definiraju na početku kôda pomoću funkcije `define()`. Kad se jednom kreiraju, njihova vrijednost se u toku programa više ne može mijenjati. Pravila za imenovanje konstanti su jednostavna. Preporuča se početak naziva velikim slovima ili znakom `_` radi lakšeg razlikovanja od varijabli. Za razliku od varijabli, konstante su globalne i vidljive kroz cijeli program. Primjer definiranje konstante se vidi u ispisu 18.

```
define("CONSTANT", "Hello world.");  
echo CONSTANT;
```

### Ispis 18: Konstanta u PHP-u

## 2.7 XAMPP

XAMPP je besplatna platforma otvorenog kôda koja služi za izradu lokalnog poslužitelja za testiranje i razvoj aplikacija [5]. Kompatibilna je sa svim današnjim modernim operativnim sustavima Windows, Linux i Mac. Za spremanje podataka koristi se MariaDB SQL baza koja je razvijena od strane MySQL tima programera. XAMPP nudi mogućnost testiranja aplikacija napisanih u jezicima PHP i Perl. Kako bi se započela upotreba XAMPP platforme potrebno je preuzeti instalacijski paket sa službene internetske stranice te pokrenuti izvršnu datoteku. Osim odabira mape za instalaciju, potrebno je instalirati i MySQL servis za komunikaciju s bazom podataka. Nakon odabira završnih postavki pokreće se instalacija. Nakon pokretanja programa, u istom je potrebno pokrenuti module Apache i MySQL. Prilikom pokretanja prikazuju se informacije o tome koje je lokalne priključke (engl. *local port*) program zauzeo za svoj rad i nakon toga sve je spremno

za upotrebu. Svi projekti se spremaju u `htdocs` mapu i od tamo se pokreću. Za pristup projektu u preglednik se treba upisati sljedeća adresa <http://localhost/ime-projekta>.

## 2.8 My Structured Query Language

MySQL je najrašireniji sustav otvorenog kôda za upravljanje relacijskim bazama podataka na poslužitelju [3]. Koristi se SQL jezik za pisanje upita unutar baze podataka. MySQL koristi LAMP platformu koja uključuje Linux - Apache - MySQL i Perl/PHP/Python. Pristup MySQL bazi moguć je uz pomoć većine programskih jezika zbog toga što je napisan korištenjem jezika C i C++ koji su kompatibilni sa svim aplikacijama.

MySQL sustav podržava korištenje više tipova tablica u bazi, a najpoznatiji su InnoDB i MyISAM. Kod InnoDB tablica kada klijent pošalje upit poslužitelju za provjeru podataka u bazi, ne zaključava se cijela tablica kao kod MyISAM-a nego samo selektirani redovi. Upiti koji se međusobno ne podudaraju mogu se izvršavati paralelno za razliku od MyISAM tablice.

U izradi aplikacije korištene su funkcije koje omogućavaju spoj na bazu i dohvaćanje podataka kako bi aplikacija bila funkcionalna. Najviše su korištene:

- `mysqli_connect()` – služi za spajanje na bazu podataka
- `mysqli_query()` – pomoću ove funkcije izvršavaju se SQL upiti na poslužitelj i vraća rezultat
- `mysqli_fetch_array()` – koristi se za dohvaćanje reda u tablici i vraća ga u obliku niza
- `mysqli_num_rows()` – funkcija vraća broj redova koji zadovoljavaju upit
- `mysqli_fetch_assoc()` – funkcija koja vraća rezultat u obliku asocijativnog niza



## 3 Opis praktičnog rada

U ovom poglavlju poblize će biti opisana aplikacija. Prikazat će se korisničko sučelje aplikacije te kroz slike i dijelove izvornog kôda detaljnije objasniti funkcionalnosti aplikacije. Opisat će se korištenje aplikacije iz pogleda korisnika i administratora.

### 3.1 Funkcionalnosti aplikacije

Aplikacija je osmišljena da se olakša proces rezervacije smještaja. Korisnik ima mogućnost odabrati smještaj za željene datume, te vidjeti i upravljati odabirom. Administrator ima mogućnost pratiti i upravljati svim nadolazećim rezervacijama te pravo na mijenjanje cijene za buduće rezervacije.

Pri prvom pristupu aplikaciji korisnik dolazi na početnu stranicu. Tu se nalaze informacije o samom objektu, fotografije objekta, usluge uključene uz smještaj, informacije o lokaciji objekta, područje za odabir željenih datuma te kontakt informacije. U obzir je uzeta činjenica da će stranicu posjećivati korisnici iz raznih država svijeta, stoga je aplikacija na engleskom jeziku. Lokacija objekta je napravljena uz pomoć *Google Maps* funkcionalnosti. U navigacijskom dijelu nalaze se poveznice koje vode na druge stranice informativnog sadržaja. Dostupne su informacije o svakom apartmanu pojedinačno, više fotografija te što sve sadrži svaki apartman. Isto tako dostupno je područje *frequently asked questions* (FAQ) u kojoj korisnici mogu pronaći odgovore na često postavljana pitanja.

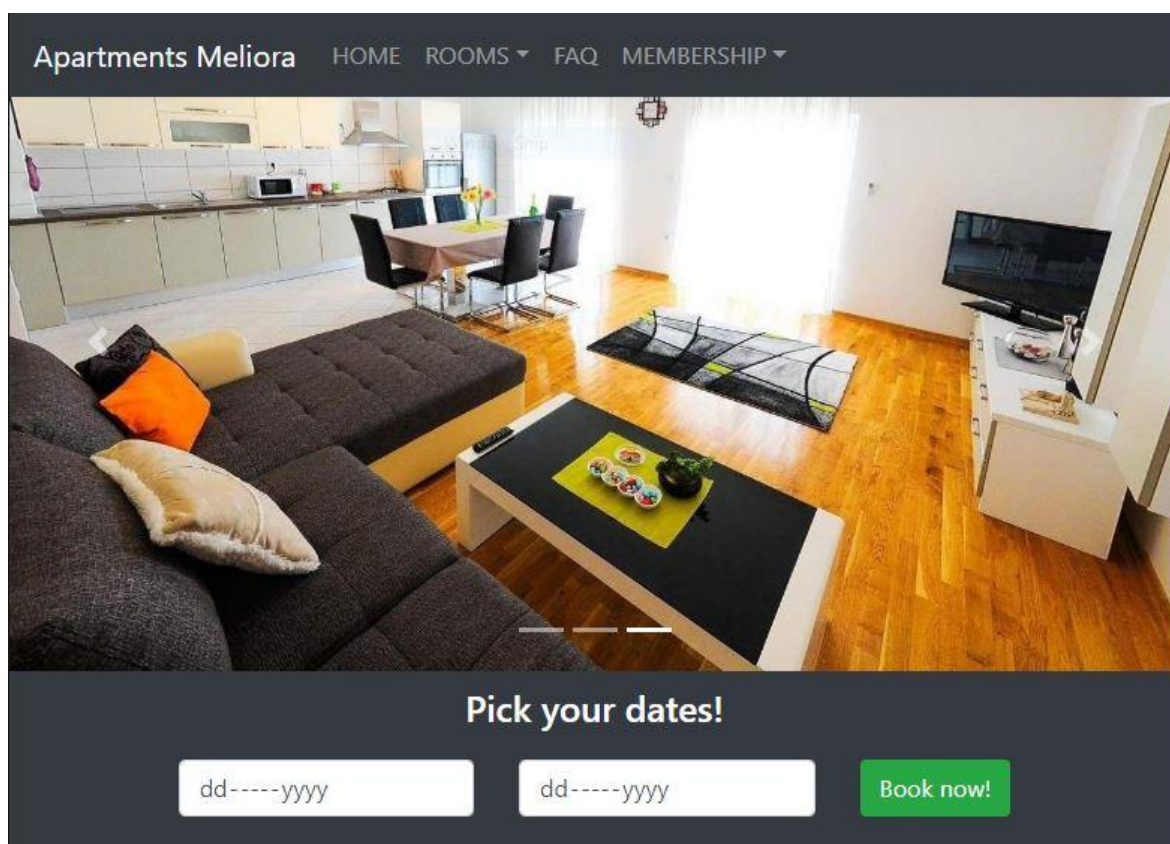
Dijelovi aplikacije koji su zanimljivi su njene funkcionalnosti, koje se odnose na korisnika i administratora. Ako korisnik želi rezervirati smještaj, prvo što može napraviti je provjeriti dostupnost apartmana za željene datume bez potrebe za registracijom ili prijavom. Ukoliko ima dostupnih apartmana za željene datume, korisnik se treba registrirati ili prijaviti te nakon toga nastaviti sa započetom rezervacijom. Nakon prijave ili registracije, u navigacijskoj traci je vidljivo ime trenutno prijavljenog korisnika. Korisnik ili administrator stranice pristupa svom profilu klikom na svoje ime. Korisnik ima pristup svim svojim nadolazećim rezervacijama koje može i otkazati, te mogućnost promjene lozinke računa. Administrator ima pristup nadolazećim rezervacijama svih korisnika koje može i otkazati, te mogućnost promjene lozinke računa i cijene za buduće rezervacije.

## 3.2 Korisničko sučelje

Aplikacija se sastoji od dva dijela, jedan dio se odnosi na korisnika, a drugi na administratora. Prvo će se prikazati iskustvo korisničke strane, a potom administratorske.

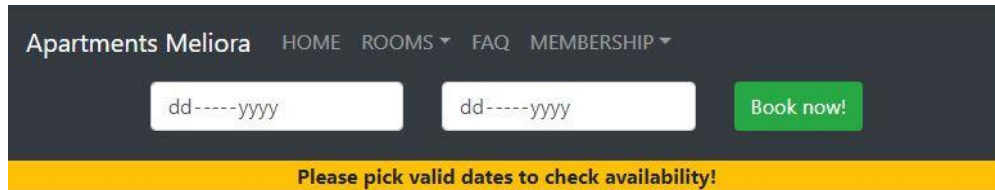
### 3.2.1 Upotreba aplikacije od strane korisnika

Prilikom prvog pristupa aplikaciji pojavljuje se početna stranica. Pri vrhu se nalazi navigacijska traka sa poveznicama na druge stranice informativnog sadržaja kao što su stranice sa detaljima apartmana i FAQ, te poveznica za članstvo (engl. *membership*) gdje se korisnik može registrirati ili prijaviti. Ispod navigacijske trake se nalazi fotogalerija smještaja te forma za odabir željenih datuma. Kako bi korisnik započeo sa korištenjem aplikacije potrebno je da unese željene datume kako bi provjerio dostupnost smještaja.



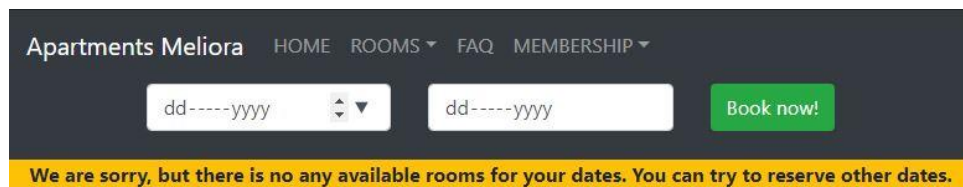
**Slika 1:** Izgled gornjeg dijela početne stranice

Potrebno je unijeti datume prijave i odjave. Datum prijave mora biti jednak ili veći od trenutnog datuma, a datum odjave mora biti veći od datuma prijave. U suprotnom će se prikazati poruka upozorenja za unos ispravnih datuma (Slika 2).



**Slika 2:** Poruka upozorenja pri odabiru pogrešnih datuma

Nakon što korisnik unese ispravne datume prijave i odjave, aplikacija mu vraća dostupne apartmane (engl. *availability*). U slučaju da niti jedan apartman nije dostupan, prikazat će se poruka upozorenja kako ni jedan apartman nije dostupan za tražene datume (Slika 3). Korisnik može unijeti nove datume ili napustiti stranicu.



**Slika 3:** Poruka upozorenja kada nema dostupnih apartmana

Ukoliko ima dostupnih apartmana, oni će biti prikazani zajedno sa ukupnom cijenom za tražene datume (Slika 4). Nakon toga korisnik je dužan se prijaviti ili registrirati ukoliko želi nastaviti rezervaciju smještaja.

Availability from 21/11/2018 to  
22/11/2018

Please Log In/Sign Up for next step	
Apartment Ground Floor	(Total price: 150 €)
Apartment 1st Floor	(Total price: 150 €)
Apartment 2nd Floor	(Total price: 150 €)

[Sign Up](#) [Log In](#)

**Slika 4:** Prikaz dostupnih apartmana

Prilikom registracije ili prijave korisnik je dužan popunit sva polja koja se od njega traže (Slika 5). U slučaju da korisnik pri registraciji unese adresu elektroničke pošte (engl. *email*) koja se već koristi, prikazati će se poruka upozorenja kako korisnik sa tom adresom već postoji. Isto tako ako korisnik pri prijavi unese adresu elektroničke pošte koja ne postoji u bazi podataka ili unese pogrešnu lozinku za postojeću adresu, aplikacija će vratiti poruke upozorenja.

### Registration

Email\*

Password\*

Name\*

Birthday\*

Country\*

Phone\*

[Submit](#)

### Login

Email

Password

[Submit](#)

**Slika 5:** Forme za registraciju i prijavu korisnika

Nakon podnošenja registracije, aplikacija korisnika vraća na dostupnost apartmana (s obzirom da je korisnik odabrao ranije željene datume) uz poruku potvrde registracije. U navigacijskoj traci je sada vidljivo ime prijavljenog korisnika. Korisnik sada može odabrati dostupne apartmane, jedan ili više njih, ovisno o dostupnosti (Slika 6). Ukoliko korisnik ne odabere ni jedan apartman, a klikne na *Submit* botun, aplikacija će vratiti poruku upozorenja kako je korisnik dužan odabrati najmanje jedan dostupni apartman ukoliko želi nastaviti sa rezervacijom. Ukoliko je korisnik odabrao apartman/e i potvrdio odabir, aplikacija ga prosljeđuje na stranicu potvrde (engl. *confirmation*).

Availability from 21/11/2018 to 22/11/2018

Please choose which apartment/s you want to reserve:

Apartment Ground Floor (Total price: 150 €)	<input checked="" type="checkbox"/>
Apartment 1st Floor (Total price: 150 €)	<input checked="" type="checkbox"/>
Apartment 2nd Floor (Total price: 150 €)	<input type="checkbox"/>

Submit

Your registration was successful!

**Slika 6:** Poruka uspješne registracije i odabir dostupnih apartmana

Na stranici potvrde aplikacija vraća korisniku odabrane apartmane, ukupnu cijenu za cijelu boravak i uvjete rezervacije (Slika 7). Ukoliko korisnik želi, može ostaviti i poruku za osoblje objekta. Kada korisnik potvrdi rezervaciju, aplikacija ga vraća na naslovnu stranicu uz poruku o uspješnosti rezervacije. Korisnik također dobiva potvrdu na svoju adresu elektroničke pošte.



## My Reservations

Upcoming reservations:

ID: 127  
Dates: 2018-11-21 to 2018-11-22  
Price: 300 €  
Apartment/s: - Ground Floor - 1st Floor  
Note:  
Date of reservation: 2018-11-19 20:01:50

Delete

## Password Change

Change your password:

Current password:

New password:

New password again:

Change

**Slika 8:** Prikaz korisničkog profila

Na dnu naslovne stranice te ostalih poveznica nalazi se kontakt forma u slučaju da korisnik ima upit (Slika 9). Korisnik je dužan unijeti ime, adresu elektroničke pošte na koju će primiti odgovor te sami upit, u protivnom aplikacija vraća poruke upozorenja. Nakon što korisnik unese podatke za sva polja te potvrdi slanje, aplikacija će vratiti poruku o uspješnom slanju upita.

## Contact

[f](#) [t](#) [G+](#) [v](#)

+385 98 111 22 33

myemail@blabla.com

Ulica 11 | 21000 Split | Croatia

Contact us and we'll get back to you within 24 hours.

Name

Email

Comment

Send

**Slika 9:** Kontakt područje

Korisnik se odjavljuje sa stranice putem poveznice u navigacijskoj traci pod svojim profilom.

### 3.2.2 Upotreba aplikacije od strane administratora

Administrator se treba prijaviti kao i svi korisnici sa svojom adresom elektroničke pošte te lozinkom. Nakon prijave na navigacijskoj traci se prikazuje profil administratora.

Ulaskom u profil, administratoru se prikazuju nadolazeće rezervacije svih korisnika, forma za promjenu lozinke te forma za promjenu cijene svih apartmana za buduće rezervacije (Slika 10).

Administrator ima mogućnost brisanja rezervacija na isti način kao i korisnik, te se prilikom brisanja šalje potvrda o otkazivanju na adresu elektroničke pošte određenog korisnika. Također na isti način kao i kod korisnika funkcionira promjena lozinke.

Forma za promjenu cijene je ograničena na vrijednosti od 50 do 200, s obzirom da se u tom rasponu inače kreću cijene. U slučaju da administrator unese vrijednosti van tog ograničenja ili ne unese vrijednost uopće, aplikacija vraća poruku upozorenja. Na ovaj način je smanjena mogućnost pogreške. Nakon ispravnog unosa aplikacija vraća potvrdu promjene cijene.

The image displays three distinct sections of the administrator interface:

- My Reservations:** A table titled "Upcoming reservations:" containing two entries. Each entry lists reservation ID, dates, price, apartment details, and reservation date. A "Delete" button is positioned below the table.
- Password Change:** A form titled "Change your password:" with three input fields: "Current password:", "New password:", and "New password again:". A "Change" button is located at the bottom right.
- Price Change:** A form titled "Change your price:" with a single input field for "New Price (€)". A "Change" button is located at the bottom right.

**Slika 10:** Prikaz administratorskog panela



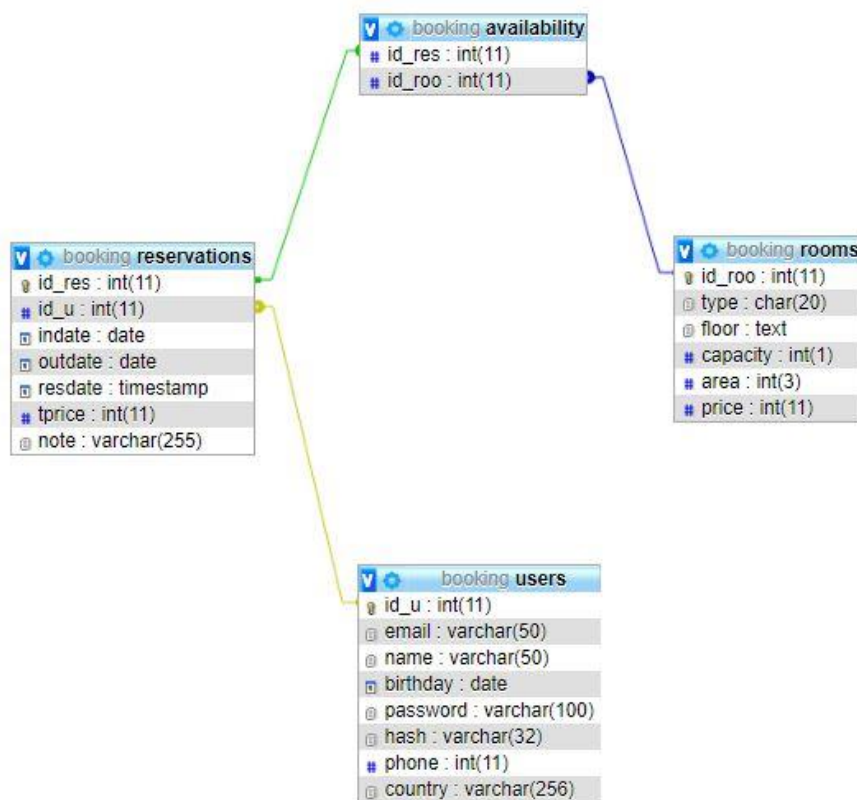
### 3.3 Implementacija funkcionalnosti aplikacije

Baza podataka je jedna od najbitnijih komponenti aplikacije, stoga će biti pojašnjena njena implementacija. Isto tako biti će pojašnjeno kako je kreirana forma za prijavu i registraciju, administratorsko i korisničko sučelje, proces rezervacije smještaja, kontakt forma te izgled prilagodljiv veličini zaslona.

#### 3.3.1 Implementacija baze podataka

Pristup bazi je omogućen preko *phpMyAdmin* sučelja. Kako bi korisnik pristupio sučelju, u preglednik upisuje *Uniform Resource Locator* (URL) adresu *localhost/phpmyadmin*.

Baza podataka „booking“ se sastoji od četiri tablice koje su relacijski povezane u jednu cjelinu. Tablice koje se nalaze u bazi su „users“, „reservations“, „availability“ i „rooms“ (Slika 11).



Slika 11: ER dijagram baze podataka „booking“

Dvije vrste relacija pomoću kojih su tablice međusobno povezane su 1:M i M:M. Veza između tablica „users“ i „reservations“ te „rooms“ i „availability“ je primjer veze jedan prema više ili 1:M. Kod veze više prema više ili M:M kojom su povezane tablice „reservations“ i „rooms“, problem se riješio uvođenjem treće tablice „availability“ koja u sebi sadrži dva strana ključa, po jedan primarni ključ iz svake tablice. Brisanje i ažuriranje podataka u vezi M:M vrši se kaskadno. To znači da ispravak ili brisanje vrijednosti nadređenog atributa podrazumijeva ispravak ili brisanje vrijednosti podređenog atributa.

### 3.3.2 Postupak rezervacije smještaja

#### 3.3.2.1 Provjera dostupnosti smještaja

Prilikom provjere dostupnosti smještaja, dohvaćaju se datumi prijave i odjave uneseni od strane korisnika. Aplikacija unesene datume korisnika postavlja kao `$_SESSION` varijable, te prosljeđuje korisnika na stranicu dostupnosti (Ispis 19).

```
if(filter_has_var(INPUT_POST, 'submit')){
    $_SESSION["chkin"] = $_POST['chkin'];
    $_SESSION["chkout"] = $_POST['chkout'];
    header('Location: availability.php');
}
```

#### **Ispis 19:** Deklariranje `$_SESSION` varijabli kao datum prijave i odjave gosta

Na stranici dostupnosti su deklarirane varijable `$chkin` i `$chkout` kao `$_SESSION` varijable iz prethodnog koraka kako bi ih bilo lakše koristiti. Kako bi se izračunao točan broj noći koje gost želi rezervirati, uz pomoć funkcije `date_diff()` izračunata je razlika u datumima koja je spremljena u varijablu `$nights`. Taj će se podatak kasnije koristiti za izračun cijene boravka (Ispis 20).

```
$chkin = $_SESSION['chkin'];
$chkout = $_SESSION['chkout'];

$date1=date_create($chkin);
$date2=date_create($chkout);
$diff=date_diff($date1,$date2);
$night = $diff->format('%a');
```

### Ispis 20: Deklaracija `$chkin` i `$chkout` varijabli te izračun broja noći

U sljedećem koraku aplikacija provjerava da li su ispravni datumi koje je unio korisnik. Ako je datum prijave `$chkin` manji od trenutnog datuma ili ako je datum odjave `$chkout` manji ili jednak datumu prijave `$chkin`, onda uneseni datumi nisu ispravni. Aplikacija vraća poruku upozorenja `$_['message']`, uneseni datumi se poništavaju te se prosljeđuje korisnika na naslovnu stranicu (Ispis 21).

```
if ($chkin < date("Y-m-d") OR $chkout <= $chkin)
{
    $_SESSION['message'] = 'Please pick valid dates to check
availability!';
    unset($_SESSION['chkin']);
    unset($_SESSION['chkout']);
    header("location: index.php#error");
    exit();
}
```

### Ispis 21: Provjera unesenih datuma od strane korisnika

U slučaju da su uneseni datumi ispravni, postavlja se upit u kojem se traže dostupni apartmani te ukupna cijena svakog apartmana pojedinačno za tražene datume. U ovom upitu se koristi varijabla `$night` koja je određena ranije. Koristi se za izračun ukupne cijene svakog dostupnog apartmana pojedinačno.

Ukoliko nema dostupnih apartmana za tražene datume, aplikacija vraća poruku upozorenja te savjetuje korisniku da odabere druge datume ako želi. Uneseni datumi od korisnika se poništavaju te se korisnika prosljeđuje na naslovnu stranicu (Ispis 22).

```

$result= $mysqli->query("SELECT rooms.id_roo, rooms.type, rooms.floor,
rooms.price * '$nights' AS tprice
    FROM rooms WHERE rooms.id_roo NOT IN
(SELECT availability.id_roo FROM availability
    WHERE availability.id_res IN
(SELECT reservations.id_res FROM reservations
    WHERE (reservations.indate < '$chkout' AND reservations.outdate -
INTERVAL 1 DAY >= '$chkin'))") or die($mysqli->error());

$postes = mysqli_fetch_all($result, MYSQLI_ASSOC);

if ($result->num_rows < 1)
{
    $_SESSION['message'] = 'We are sorry, but there is no any available
rooms for your dates. You can try to reserve other dates.';
    unset($_SESSION['chkin']);
    unset($_SESSION['chkout']);
    header("location: index.php#error");
    exit();
}

```

### Ispis 22: Upit dostupnosti apartmana

Dostupni apartmani se ispisuju `foreach()` petljom koja prolazi kroz rezultat prethodnog upita. Ukoliko je korisnik prijavljen, za svaki dostupni apartman biti će ispisana njegova vrsta, etaža, ukupna cijena za tražene datume te potvrdni okvir (engl. *check box*). Ukoliko korisnik nije prijavljen ispisati će se dostupni apartmani sa cijenom, međutim bez mogućnosti odabira. Kako bi korisnik nastavio sa rezervacijom dužan je se prijaviti.

Deklarirana je `$_SESSION['tprice']` varijabla koja će poslužiti kasnije za izračun ukupne cijene cijele rezervacije. Cijeli primjer se nalazi na ispisu 23.

```

<ul class="list-group list-group-flush">
  <?php foreach($posts as $post) : ?>
    <li class="list-group-item">
      <ul class="list-inline">
        <li class="list-inline-item">
          <?php
            echo $post['type'].'
'. $post['floor'].' (Total price: '.$post['tprice'].' €)';
            $_SESSION["tprice"] =
$post['tprice'];
          ?>
        </li>
        <li class="list-inline-item float-right">
          <input type="checkbox" name="Apartment[]"
value="<?php echo $post['floor'];?>">
        </li>
      </ul>
    </li>
  </ul>

```

**Ispis 23:** `foreach()` petlja koja ispisuje dostupne apartmane

### 3.3.2.2 Odabir dostupnog smještaja

Nakon ispisa dostupnih apartmana, korisnik je dužan odabrati putem potvrdnog okvira najmanje jedan apartman. U niz `$floors` se spremaju odabrani apartmani, te se onda taj niz deklarira kao `$_SESSION` jer će biti potreban kasnije. Isto tako sumu svih odabranih apartmana `count($floors)` se postavlja za `$_SESSION` varijablu jer će biti potrebna kasnije. U slučaju da korisnik ne odabere ni jedan apartman, aplikacija vraća poruku upozorenja. U suprotnom korisnika se prosljeđuje na stranicu potvrde (Ispis 24).

```

if(isset($_POST['submit']))
{
    $floors = $_POST['Apartment'];
    $_SESSION["floor"] = $floors;

    if(empty($floors))
    {
        $_SESSION['message'] = 'You must choose at least one
apartment to make reservation!';
        header("location: availability.php");
        exit();
    }
    else
    {
        $_SESSION["n"] = count($floors);
        header("location: total.php");
    }
}

```

#### **Ispis 24:** Prikaz deklaracije niza \$floors

Na stranici potvrde se prikazuje ispis odabranog smještaja, prostor za unos poruke osoblju smještaja od strane korisnika te ukupna cijena za cijelu rezervaciju. Prikaz odabranih apartmana se vrši `for()` petljom, `$N` je suma odabranih apartmana koja je određena korak ranije. `$floors` je niz odabranih apartmana koji su isto tako određeni korak ranije. `$price` je broj odabranih apartman `$N` pomnoženo sa ukupnom cijenom jednog apartmana za tražene datume `$_SESSION['tprice']`. Ovaj primjer se može vidjeti na ispisu 25.

```

<ul class="list-group list-group-flush">
  <li class="list-group-item">
    <strong>
      <?php
        for($i=0; $i < $N; $i++)
        {
          echo($floors[$i] . ", ");
        }
      ?>
    </strong>
  </li>
  <li class="list-group-item">
    <textarea class="form-control" id="comments" name="note"
placeholder="Note to Owner" rows="1"></textarea>
  </li>
  <li class="list-group-item text-right">
    h6>Total price:<?php echo ' '.$price.' €'; ?></h6>
  </li>

```

### Ispis 25: Prikaz for petlje za ispis odabranog smještaja

Kada korisnik potvrdi rezervaciju smještaja pritiskom na *submit*, u tablicu „reservations“ se unose vrijednosti za ID korisnika („id\_u“), datum prijave („indate“) i odjave („outdate“), ukupna cijena rezervacije („tprice“) te poruka korisnika („note“) ako ju je upisao. Prilikom unosa nove rezervacije, automatski se stvara ID rezervacije („id\_res“).

Za unos podatka u tablicu „availability“ potrebni su ID rezervacije te ID-ovi rezerviranih apartmana („id\_roo“). Postavlja se upit u kojem se traži ID nove rezervacije te `for()` petlja u kojoj se prolazi kroz upit za ID-ove odabranih apartmana. Rezultate tih dvaju upita se unose u tablicu „availability“ kroz `for()` petlju, tako da se unese svaki odabrani apartman za ID nove rezervacije. Cijeli primjer se nalazi na ispisu 26.

```

if(filter_has_var(INPUT_POST, 'submit'))
{
    $note = htmlspecialchars($_POST['note']);
}
if(isset($_POST['submit']))
{
    $query = "INSERT INTO reservations(id_u, indate, outdate, tprice,
note) VALUES ($user_id, '$skin', '$skout', $price, '$note)";

if(mysqli_query($mysqli, $query))
{
$result1 = $mysqli->query("SELECT reservations.id_res
FROM reservations
WHERE reservations.id_u = '$user_id'
ORDER BY reservations.id_res DESC");

    $posts1 = mysqli_fetch_row($result1);
    $res=$posts1[0];
}

for ($i=0; $i < $N; $i++)
{
    $result2 = $mysqli->query("SELECT rooms.id_roo
FROM rooms
WHERE floor = '$floors[$i]'");

    $posts2 = mysqli_fetch_row($result2);
    $roo = $posts2[0];

    $query2 = "INSERT INTO availability(id_res, id_roo) VALUES
($res, $roo)";
    mysqli_query($mysqli, $query2);
}
}

```

### Ispis 26: Prikaz unosa podataka nove rezervacije

Nakon unosa nove rezervacije u bazu, korisniku se šalje na adresu elektroničke pošte potvrda rezervacije, te ga se prosljeđuje na naslovnicu uz poruku potvrde rezervacije. Taj primjer se nalazi na ispisu 27.



```

$to= $email;
$subject = 'Successful reservation | Apartments Meliora';
$message_body = '
    Hello ' . $name . ',

    Thank you for choosing us,

    Your reservation details:

    ID: ' . $res . '
    Dates: ' . $chkin . ' to ' . $chkout . '
    Price: ' . $price . '
    Apartment/s: ' . $floors[0] . ' ' . $floors[1] . ' ' . $floors[2] . '
    Visit our site http://localhost/zavrzni/index.php for more
    informations.

    Kind regards,
    Apartments Meliora Team';

mail( $to, $subject, $message_body );

$_SESSION['message'] = 'Thank you for choosing us. Your reservation is
confirmed.';
header ('Location: index.php#error');
exit();
}

```

### Ispis 27: Poruka potvrde nove rezervacije

#### 3.3.3 Sustav za registraciju i prijavu korisnika

Prilikom registracije funkcija `filter_has_var()` provjerava da li postoji `submit`. Ako postoji, korisnički podaci kao što su `name`, `phone` i drugi se dohvaćaju preko `$mysqli->escape_string` funkcije koja izbjegava posebne znakove u *string*-u za upotrebu u SQL izvatku, uzimajući u obzir trenutačni skup znakova veze. Radi sigurnije zaštite privatnosti podataka lozinka korisnika je šifrirana `md5` algoritmom. Ovaj primjer se nalazi u ispisu 28.

```

if(filter_has_var(INPUT_POST, 'submit'))
{
    $name = $mysqli->escape_string($_POST['name']);
    $phone = $mysqli->escape_string($_POST['phone']);
    $country = $mysqli->escape_string($_POST['country']);
    $bday = $mysqli->escape_string($_POST['bday']);
    $email = $mysqli->escape_string($_POST['email']);
    $password = $mysqli-
>escape_string(password_hash($_POST['password'], PASSWORD_BCRYPT));
    $hash = $mysqli->escape_string( md5( rand(0,1000) ) );

```

### Ispis 28: Dohvaćanje unosa od korisnika

Upitom se provjerava da li u bazi već postoji korisnik s istom adresom elektroničke pošte. Ukoliko upit vraća ijedan redak, to znači da postoji isti korisnik. U slučaju da postoji, aplikacija vraća poruku upozorenja. Ovaj primjer se nalazi u ispisu 29.

```

$result = $mysqli->query("SELECT * FROM users WHERE email='$email'") or
die($mysqli->error());
if ( $result->num_rows > 0 )
{
    $_SESSION['message'] = 'User with this email already exists!';
    header("location: registration.php#sign");
    exit();
}

```

### Ispis 29: Provjera korisničke adrese elektroničke pošte

U slučaju da u bazi podataka ne postoji ista adresa elektroničke pošte, u tablicu „users“ se ubacuju podaci novog korisnika (Ispis 30).

```

else {
    $sql = "INSERT INTO booking.users (email, name, birthday,
country, phone, password, hash) "
        . "VALUES
('$email', '$name', '$bday', '$country', '$phone', '$password', '$hash')";

```

### Ispis 30: Unos podataka novog korisnika u bazu

S obzirom da će za budući rad trebati podaci novog korisnika, ponovnim upitom u tablicu „users“ se dohvaća i ID novog korisnika („id\_u“), koji se automatski stvara prilikom unosa svakog novog korisnika te se postavljaju \$\_SESSION varijable za podatke korisnika koje će se koristiti kasnije u radu. Ovaj primjer se nalazi u ispisu 31.

```
if ( $mysqli->query($sql) )
{
    $result1 = $mysqli->query("SELECT * FROM users WHERE
email='$email'");

    if ( $result1->num_rows == 0 )
    {
        $_SESSION['message'] = "Registration has failed!";
        header("location: registration.php");
        exit();
    }
    else
    {
        $user = $result1->fetch_assoc();
        $_SESSION['email'] = $user['email'];
        $_SESSION['name'] = $user['name'];
        $_SESSION['user_id']=$user['id_u'];
        $_SESSION['password']=$user['password'];
    }
}
```

**Ispis 31:** Upit podatka novog korisnika te određivanje \$\_SESSION varijabli za iste

Također se postavlja \$\_SESSION['logged\_in'] varijabla kojom se daje do znanja da je korisnik prijavljen. Ispisuje se poruka o uspješnosti registracije, te se šalje poruka potvrde na adresu elektroničke pošte novog korisnika (Ispis 32).

```

$_SESSION['logged_in'] = true;
    $_SESSION['message'] = "Your registration was successful!";
    $to      = $email;
    $subject = 'Successful registration | Apartments Meliora';
    $message_body = 'Hello '.$name.',
        Thank you for signing up,

        You are now ready to reserve our apartments!

        Visit our site http://localhost/zavrnsni/index.php for more
        informations.

        Kind regards,
        Apartments Meliora Team';

    mail( $to, $subject, $message_body );

```

### **Ispis 32: Potvrda prijave i registracije za novog korisnika**

Ukoliko je korisnik već ranije provjerio dostupnost apartmana, nakon registracije ga se prosljeđuje na stranicu dostupnosti, a ako nije, prosljeđuje ga se na naslovnu stranicu (Ispis 33).

```

if (isset($_SESSION['chkin'])){
    header("location: availability.php");
    exit();
}
else{
    header("location: index.php#error");
    exit();
}
}

```

### **Ispis 33: Prosljeđivanje korisnika nakon registracije**

### 3.3.4 Administratorski i korisnički profil

#### 3.3.4.1 *Brisanje rezervacija*

U ovom djelu će biti opisane mogućnosti administratora i korisnika u svojim profilima. Prvo će biti objašnjeno na koji način se upravlja rezervacijama.

Tokom registracije ili prijave su deklarirane `$_SESSION` varijable za ID korisnika te adresu elektroničke pošte korisnika. Na stranici profila se pozivaju njihove vrijednosti te se stavljaju u kraće izraze `$email` i `$user_id` kako bi kôd bio čitljiviji.

Ukoliko je administrator prijavljen, slijedi upit za ispis nadolazećih rezervacija svih korisnika, složene po datumu prijave i ID-u rezervacije. Nakon toga slijedi `foreach()` petlja koja prolazi kroz rezultat istog upita, tako da novim upitom vraća sve rezervirane apartmane za određenu rezervaciju. Primjer se nalazi u ispisu 34.

```
$email = $_SESSION['email'];
$user_id=$_SESSION['user_id'];
if(isset($_SESSION['logged_in'])&& $_SESSION['logged_in']==true &&
$user_id == 6){
    $rez = $mysqli->query("SELECT * FROM reservations
        WHERE reservations.indate >= CURRENT_DATE
        ORDER BY reservations.indate ASC,
reservations.id_res;") or die($mysqli->error());
    $rezz = mysqli_fetch_all($rez, MYSQLI_ASSOC);

    foreach ($rezz as $index => $key) {
        $key1 = $key['id_res'];
        $rez1 = $mysqli->query("SELECT rooms.floor FROM rooms
            INNER JOIN availability ON rooms.id_roo
=availability.id_roo
            WHERE availability.id_res = '$key1';") or
die($mysqli->error());
        $rezz1 = mysqli_fetch_all($rez1, MYSQLI_ASSOC);
        $rezz[$index]['rooms'] = $rezz1;
    }
}
```

**Ispis 34:** Upit za nadolazeće rezervacije svih korisnika

Ukoliko je korisnik prijavljen, razlika u odnosu na administratora je u tome što se postavlja upit za ispis nadolazećih rezervacija tog korisnika.

```
elseif(isset($_SESSION['logged_in']) && $_SESSION['logged_in']==true){
    $rez = $mysqli->query("SELECT *
        FROM reservations
        WHERE reservations.id_u = $user_id AND
reservations.indate >= CURRENT_DATE
        ORDER BY reservations.indate ASC,
reservations.id_res;") or die($mysqli->error());

    $rezz = mysqli_fetch_all($rez, MYSQLI_ASSOC);
```

### Ispis 35: Upit za nadolazeće rezervacije prijavljenog korisnika

Kako bi se rezervacije ispisale, korištene su dvije `foreach()` petlje, jedna koja je prolazila kroz rezultate upita nadolazećih rezervacija, te druga smještena unutar prve koja je prolazila kroz upit za sve odabrane apartmane određene rezervacije. Za svaku rezervaciju se odredio potvrdni okvir. Primjer se nalazi u ispisu 35.

```
<ul class="list-group list-group-flush">
    <?php foreach($rezz as $post) : ?>
        <li class="list-group-item">
            <?php echo 'ID: '.$post['id_res'].'<br> Dates:
'.$post['indate'].' to '.$post['outdate'].'<br> Price:
'.$post['tprice'].' € <br> Apartment/s: ';
                foreach ($post['rooms'] as $rez1){
                    echo ' - '.$rez1['floor'];
                }
            echo '<br> Note: '.$post['note'].'<br> Date of
reservation: '.$post['resdate'];?>
            <input class="float-right" type="checkbox"
name="id_res[]" value="<?php echo $post['id_res'];?>">
        </li>
    <?php endforeach; ?>
</ul>
```

### Ispis 35: Prikaz načina ispisa detalja rezervacija

Kako bi određena rezervacija bila izbrisana, korisnik ili administrator je dužan odabrati najmanje jednu putem potvrdnog okvira. Odabrane rezervacije se spremaju u `$reservations`. Ukoliko ne odabere ni jednu rezervaciju, tj. ako je `$reservations` prazan, aplikacija vraća poruku upozorenja. Primjer se nalazi u ispisu 36.

```
if(isset($_POST['submit'])) {  
    $reservations = $_POST['id_res'];  
  
    if(empty($reservations))  
    {  
        $_SESSION['message'] = 'Please choose reservation to  
delete!';  
  
        header("location: profile.php#error");  
        exit();  
    }  
}
```

### Ispis 36: Poruka upozorenja ako nije odabrana ni jedna rezervacija

Ukoliko administrator prosljedi odabrane rezervacije za brisanje tj. ako `$reservations` nije prazan, `foreach()` petljom se prolazi kroz niz odabranih rezervacija te se za svaku postavlja upit u kojem se traži korisnika koji je napravio pojedinu rezervaciju. Primjer se nalazi u ispisu 37.

```
else  
{  
    foreach ($reservations as $reservation)  
    {  
        $usr = $mysqli->query("SELECT * FROM users  
INNER JOIN reservations ON users.id_u=reservations.id_u  
WHERE reservations.id_res= $reservation") or  
die($mysqli->error());  
    }  
}
```

### Ispis 37: `foreach()` petlja koja prolazi kroz upit

Rezultati upita se koriste kako bi se došlo do adrese elektroničke pošte svakog od korisnika. Za svaku rezervaciju se šalje poruka na adresu elektroničke pošte korisnika o

potvrđi brisanja iste, te se briše sama rezervacija iz baze podatka. Nakon petlje, aplikacija vraća poruku o uspješnosti brisanja odabranih rezervacija. Primjer se nalazi u ispisu 38.

```
$usr1 = mysqli_fetch_row($usr);
$em = $usr1[1];

$to= $em;
$subject = 'Reservation Cancellation | Apartments Meliora';
$message_body = '
    Hello '.$usr1[2].',

    Your reservation is canceled.

    ID reservation: '.$reservation.'

    Visit our site http://localhost/zavrnsni/index.php for
more informations.

    Kind regards,
    Apartments Meliora Team';

mail( $to, $subject, $message_body );

$query = "DELETE FROM reservations WHERE reservations.id_res
= $reservation";
mysqli_query($mysqli, $query);
}
$_SESSION['message'] = 'Deletion was successful!';
header("Location: profile.php#error");
exit();
}
```

### Ispis 38: Brisanje rezervacija iz baze od strane administratora

Kod korisnika se razlikuje to što se zna koje vlasnik odabranih rezervacija, pa nema potrebe za upitom, već se za svaku odabranu rezervaciju šalje poruka o otkazivanju na adresu



elektroničke pošte administratora, te se rezervacija briše iz baze podataka. Primjer se nalazi u ispisu 39.

```
else {
    foreach ($reservations as $reservation) {
        $to= 'rajic1911@gmail.com';
        $subject = 'Reservation Cancellation | Apartments Meliora';
        $message_body = '
                Hello,

                Reservation is canceled.

                ID reservation: '.$reservation.'';

        mail( $to, $subject, $message_body );

        $query = "DELETE FROM reservations WHERE reservations.id_res
= $reservation";
        mysqli_query($mysqli, $query);}
```

### **Ispis 39:** Brisanje rezervacija iz baze od strane korisnika

#### *3.3.4.2 Uređivanje cijene*

Administrator ima mogućnost promjene cijene za buduće rezervacije. Unos nove cijene se sprema u \$quantity. Nakon toga slijedi upit koji ažurira u tablici „rooms“ cijene za sve apartmane novom cijenom koju je upisao korisnik. Aplikacija vraća poruku potvrde. Primjer je vidljiv u ispisu 40.

```
if(isset($_POST['submit3'])){
    $quantity = $_POST['quantity'];
    $querychange1 = ("UPDATE rooms SET price ='$quantity'");
    mysqli_query($mysqli, $querychange1);
    $_SESSION['message'] = 'Price change was successful!';
    header("Location: profile.php#error");
    exit();}
```

### **Ispis 40:** Izmjena cijene apartmana

### 3.3.4.3 Promjena lozinke

Promjenu lozinke mogu zatražiti i administrator i korisnik. Kako bi promijenili lozinku, potrebno je unijeti trenutnu lozinku koja se sprema u `$password`, novu lozinku koja se sprema u `$newpassword` te ponovljenu novu lozinku koja se sprema u `$repeatnewpassword`. Primjer se nalazi u ispisu 41.

```
if(filter_has_var(INPUT_POST, 'submit2'))
{
    $password = $_SESSION['password'];
    $newpassword = $_POST['newpassword'];
    $repeatnewpassword = $_POST['repeatnewpassword'];
```

#### **Ispis 41:** Dohvat unesenih lozinki

Prva provjera je da li je unesena trenutna lozinka stvarno jednaka trenutnoj lozinki, ukoliko nije, aplikacija vraća poruku upozorenja. Ako je prva provjera ispravna, slijedi druga provjera u kojoj se ispituje da li je nova unesena lozinka jednaka ponovo unesenoj novoj lozinki tj. `$newpassword == $repeatnewpassword`. Ukoliko je tvrdnja netočna, aplikacija vraća poruku upozorenja. Ako je tvrdnja ispravna, nova lozinka se kriptira te se unosi u bazu podatka za određenog korisnika. Aplikacija vraća poruku uspješnosti. Primjer se vidi u ispisu 42.

```

if ( password_verify($_POST['password'], $password) )
    {
        if ($newpassword == $repeatnewpassword)
            {
                $newpassword = $mysqli-
>escape_string(password_hash($_POST['newpassword'], PASSWORD_BCRYPT));

                $querychange = ("UPDATE users SET password
= '$newpassword'
WHERE users.id_u = '$user_id'");
                mysqli_query($mysqli, $querychange);

                $_SESSION['message'] = 'Password updated!';
                header("Location: profile.php#error");
                exit();
            }
        else
            {
                $_SESSION['message'] = 'New password does not match!';
                header("Location: profile.php#error");
                exit();
            }
    }
else
    {
        $_SESSION['message'] = 'Current password does not match!';
        header("Location: profile.php#error");
        exit();
    }
}

```

#### Ispis 42: Promjena postojeće lozinke

### 3.3.5 Kontakt forma

Kako bi korisnik poslao upit osoblju smještaja, dužan je popuniti sva polja kontakt forme. Ime se sprema u `$name`, adresa elektroničke pošte u `$email` te upit u `$message`. Ukoliko upit bude uspješno poslan, aplikacija vraća poruku uspješnosti. U suprotnom vraća poruku upozorenja. Primjer se nalazi u ispisu 43.

```
$msg = '';
$msgClass = '';
if(filter_has_var(INPUT_POST, 'submit1')){
    $name = htmlspecialchars($_POST['name']);
    $email = htmlspecialchars($_POST['email']);
    $message = htmlspecialchars($_POST['message']);

    $toEmail = 'rajic1911@gmail.com';
    $subject = 'Contact Request From '.$name;
    $body = '<h2>Contact Request</h2>
        <h4>Name</h4><p>'.$name.'</p>
        <h4>Email</h4><p>'.$email.'</p>
        <h4>Message</h4><p>'.$message.'</p>';

    $headers = "MIME-Version: 1.0" . "\r\n";
    $headers .= "Content-Type:text/html;charset=UTF-8" . "\r\n";

    $headers .= "From: " . $name . "<".$email.">". "\r\n";

    if(mail($toEmail, $subject, $body, $headers))
    {
        $msg = 'Your email has been sent!';
        $msgClass = 'alert-success';
    }
    else
    {
        $msg = 'Your email was not sent';
        msgClass = 'alert-danger';
    }
}
```

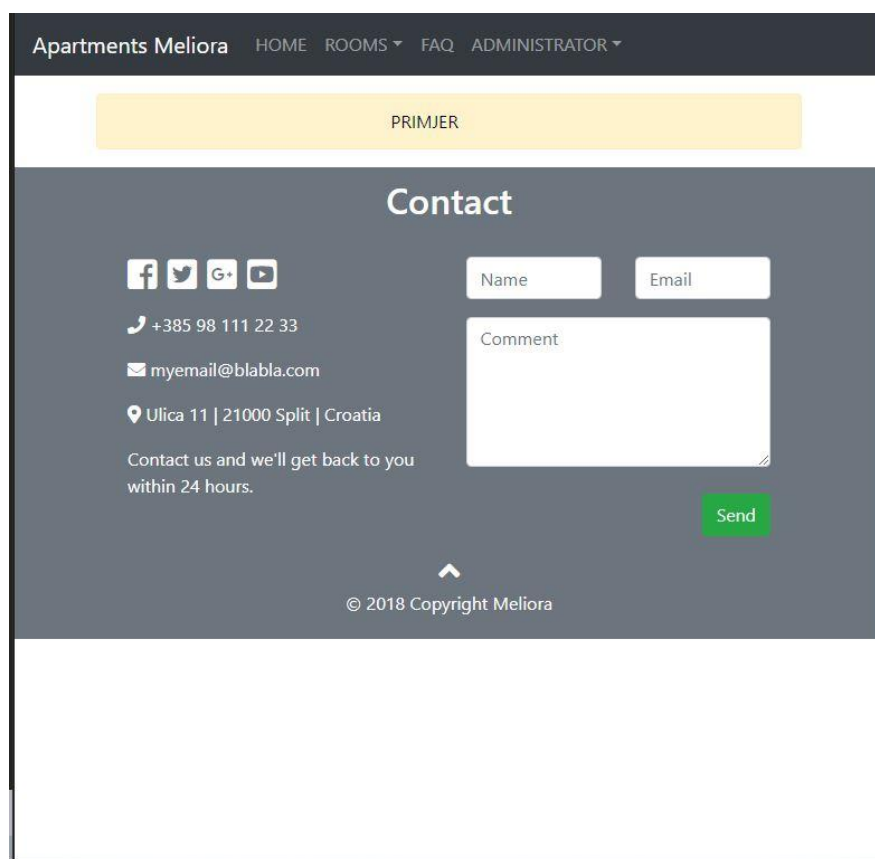
### Ispis 43: Kontakt forma

### 3.3.6 Dizajn

Izgled prilagodljiv veličini zaslona se temelji na tome da se sadržaj dinamički prilagodi uređajima manjih širina zaslona. Za uspješnu realizaciju korišten je okvir Bootstrap koji ima predefinirane klase prilagodljivih dimenzija. One imaju unaprijed napisana CSS svojstva te svojstvo *@media* koje oblikuje ponašanje elemenata pri određenoj širini zaslona.

Za definiranje ponašanja elemenata pojedinih dijelova stranice koristili su se prilagođeni medijski upiti. Te definicije se uglavnom odnose na *padding*, *margin* pojedinih elemenata te na fontove.

Prikazat će se način na koji se došlo do rješenja ukoliko na stranici u samom tijelu nema mnogo sadržaja, već samo neku poruku ili neki manji element pa se na većim zaslonima dno stranice „odljepljuje“ sa dna i ostaje praznina. Primjer se nalazi na slici 12.



**Slika 12:** Primjer lošeg dizajna

Taj problem je riješen postavljanjem cijelog sadržaja u HTML-u u jedan `<div>` element klase `wrap`, te se unutar tog `<div>` elementa odrede `<header>`, `<main>` i `<footer>` područja. Primjer se nalazi u ispisu 44.

```
<div class="wrap">
  <header>Header</header>
  <main>Main</main>
  <footer>Footer</footer>
</div>
```

#### Ispis 44: Postavljanje `<div>` elementa klase `wrap`

Sljedeći korak je definiranje ponašanja elemenata putem CSS-a. Za klasu `wrap` su definirana svojstva `display: flex`, `flex-flow: column nowrap` te `min-height: 100vh`, a za klasu `main` svojstvo `flex-grow: 1`. Primjer se nalazi u ispisu 45, a rezultat je vidljiv na slici 13.

```
.wrap {
  display: flex;
  flex-flow: column nowrap;
  min-height: 100vh;
}
main {
  flex-grow: 1;
}
```

#### Ispis 45: CSS svojstva klase `wrap` i područja `main`

PRIMJER

## Contact



+385 98 111 22 33

myemail@blabla.com

Ulica 11 | 21000 Split | Croatia

Contact us and we'll get back to you within 24 hours.

Name

Email

Comment

Send

© 2018 Copyright Meliora

**Slika 13:** Primjer dobrog dizajna

## 4 Zaključak

Aplikacija je ostvarila svoj cilj da olakša i pojednostavni rezerviranje smještaja. Implementiran je sustav koji omogućuje korisnicima odabir dostupnih apartmana za tražene datume te prikaz i upravljanje izabranim stavkama. Omogućeno je brisanje izbora te je implementirana opcija postavljanja upita prije nego se korisnik odluči na samu rezervaciju. Razvijeno je sučelje za administratora koji ima ovlasti upravljanja nad rezervacijama svih gostiju te cijenom smještaja za buduće rezervacije.

Jedna od bitnih funkcionalnosti je izgled prilagodljiv veličini zaslona jer danas se svakodnevno koriste pametni uređaji. Kako bi stranica bila prilagodljiva uređajima manjih dimenzija zaslona pobrinuo se okvir Bootstrap. U aplikaciju je implementiran sustav na engleskom jeziku što aplikaciju čini pristupačnom i razumljivom širem krugu ljudi.

Aplikaciji je ostavljeno prostora za implementaciju novih mogućnosti. Za korisnike se može uvesti plaćanje smještaja putem interneta. Mogu se uvesti različiti cjenici za određene periode, popusti za duže boravke ili rezervacije u zadnji čas (engl. *last minute*) i slično. Što se tiče statistike, mogu se uvesti razna praćenja primjerice broj zakazanih, ostvarenih i otkazanih rezervacija, praćenje iz kojih najviše krajeva svijeta sve gosti stižu kako bi se okrenuli tom tržištu te praćenje upita gostiju, što sve zahtijevaju i slično kako bi budućim gostima boravak učinili što ugodnijim.



## 5 Literatura

[1] Duckett J.: „Html & Css“, John Wiley & Sons, Inc., Kanada 2011

[2] McFarland Sawyer D.: „JavaScript & jQuery: The Missing Manual“, O'Reilly Media, Inc., Sjedinjene Američke Države 2012

[3] Beighley L. & Morrison M.: „Head First PHP & MySQL“, O'Reilly Media, Inc., Sjedinjene Američke Države 2008

[4] „Bootstrap“, Dostupno na <https://getbootstrap.com/> (18.11.2018).

[5] Ganesan P., „What is XAMPP?“, Dostupno na <https://www.wpblogx.com/what-is-xampp/> (18.11.2018).