Web prezentacija za turističku agenciju "Tourizm"

Sadržaj

[1. Uvod 2](#_Toc431388359)

[1.1 Cilj razvoja 2](#_Toc431388360)

[1.2 Obim sistema 2](#_Toc431388361)

[1.3 Tehnologije 2](#_Toc431388362)

[1.3.1 Front end 3](#_Toc431388363)

[1.3.2 Back end 3](#_Toc431388364)

[1.3.3 Druge popularne tehnologije za razvoj web-a 4](#_Toc431388365)

[1.4 Struktura aplikacije 4](#_Toc431388366)

[1.5 Alati 4](#_Toc431388367)

[1.5.1 Atom 4](#_Toc431388368)

[1.5.2 MySQL Workbench 5](#_Toc431388369)

[2. Funkcionalnost 5](#_Toc431388370)

[2.1 Frontend 5](#_Toc431388371)

[2.1.1 Korisnički deo 5](#_Toc431388372)

[2.1.2 Administracija 5](#_Toc431388373)

[2.2 Backend 6](#_Toc431388374)

[2.2.1 Rutiranje (Routing) 6](#_Toc431388375)

[3. Zaštita 6](#_Toc431388376)

[3.1 Zaštita izvornog koda i logike aplikacije 6](#_Toc431388377)

[3.2 Autorizacija i autentifikacija 7](#_Toc431388378)

[3.3 Obrada podataka 7](#_Toc431388379)

[4. Baza podataka 7](#_Toc431388380)

[4.1 UML dijagram 7](#_Toc431388381)

[4.2 Tabele 8](#_Toc431388382)

[5. Prilog 9](#_Toc431388383)

[5.1 Frontend 9](#_Toc431388384)

[5.2 Backend 11](#_Toc431388385)

[5.3 UML use case diagram 12](#_Toc431388386)

# 1. Uvod

## Cilj razvoja

Cilj razvoja ove web aplikacije je da olakša poslovanje turističke agencije na internetu. Zbog nedostatka odgovarajućih CMS (eng. Content Мanagement System –sistem za upravljane sadržajem) rešenja, odlučeno je da se razvije novi sistem u programskom jeziku PHP korišćenjem MVC dizajn paterna. Razlog za to je mogunost brzog razvoja i lakog održavanja aplikacije i koda.

## Obim sistema

Aplikacija se sastoji iz dela za administraciju destinacija i rezervacija i dela za pregled i rezervacije destinacija. Pored toga postoje stranice koje služe za detaljan pregled i pretragu svih destinacija, kao i kontakt stranica.

Administrator se prijavljuje na sistem upotrebom korisničkog imena i lozinke. On može da dodaje, briše i menja destinacije, takođe ima pristup pregledu svih rezervacija I svih destinacija.

Korisnik može da pregleda listu svih destinacija, pojedinačnu destinaciju, da pretražuje destinacije i može da napravi rezervaciju. Takođe može da pošalje email putem forme na stranici Kontakt.

## 1.3 Tehnologije

Za razvoj ove aplikacije korišćen je **MVC** (Model View Controller) dizajn patern. **MVC arhitektura** je softverski patern Model-View-Controller koji odvaja prikaz informacija od interakcije korisnika sa tim informacijama.

**Model** se sastoji od podataka aplikacije, poslovnih pravila, logike i funkcija.

**View** može da bude bilo koji izlazni prikaz podataka, kao što je dijagram ili grafik. Više prikaza istog podatka je moguće, kao što je grafik sa barovima za menadžment i tabelarni prikaz za računovođe.

**Controller** uzima ulazne podatke i konvertuje ih u komande za model ili view.

Za razvoj prezentacije korišćene su sledeće tehnologije: HTML, CSS i jQuery. Za logiku aplikacije (backend) korišćen je programski jezik PHP i MySQL sistem za rad sa bazom podataka.

### 1.3.1 Front end

**HyperText Markup Language**, poznatiji kao **HTML**, je standardni jezik za označavanje i koristi se za kreiranje veb stranica.Veb pretraživači mogu da čitaju HTML datoteke i da ih prevode u vidljive ili zvučne veb stranice. HTML opisuje strukturu sajta semantički zajedno sa signalima za prezentaciju , što ga čini markup jezikom, a ne programskim jezikom.

**Cascading Stile Sheets (CSS)** je stilski jezik koji se koristi za opisivanje prezentacije dokumenta pisanog u markup jeziku (HTML-u). CSS je dizajniran pre svega da omogući odvajanje sadržaja dokumenta od prezentacije dokumenta, uključujući aspekte kao što su raspored, boje i fontovi .

**jQuery** je kros platformska **JavaScript** biblioteka koja je dizajnirana da olakša izvršavanje skripti na klijentskoj strain. jQuery je najpopularnija JavaScript biblioteka koja se danas koristi, koriste je 65% od 10 miliona najposećenijih sajtova na internetu.

### 1.3.2 Back end

**PHP** je jezik za izvršavanje skripti na serverskoj strani dizajniran za veb razvoj, ali se koristi i kao programski jezik opšte namene. Od januara 2013. godine, PHP je instaliran na više od 240 miliona sajtova (39% od svih ispitanih) i na 2,1 miliona veb servera. Prvobitno razvijena od strane Rasmus Lerdorf-a 1994. godine, referentna implementacija PHP-a (koju pokreće Zend Engine) sada održava PHP Group. Iako je prvobitno bio skraćenica za Personal Home Page, sada se zove PHP: Hypertext Preprocessor, koji je rekurzivni akronim.

**MySQL** je sistem otvorenog koda za upravljanje relacionim bazama podataka (RDBMS); u julu 2013., bio je drugi najkorišćeniji RDBMS u svetu, i najrasprostranjeniji klijent–server RDBMS otvorenog koda. Dobio je naziv po ćerki suosnivača Michael Widenius-a, My. SQL je akronim za Structured Query Language.

### 1.3.3 Druge popularne tehnologije za razvoj web-a

**ASP.NET** je razvojno okruženje na serverskoj strani otvorenog koda za razvoj dinamičkih veb aplikacija. Razvijen je od strane Microsoft-a s namerom da olakša programiranje dinamičkih veb sadržaja i veb servisa.

## 1.4 Struktura aplikacije

Aplikacija je razdvojena u dva direktorijuma, ’**system**’ i ’**public**’. U public folderu aplikacije se nalaze fajlovi kojima je moguće direktno pristupiti.

Izgled **/public** direktorijuma:

* */css – direktorijum za smeštanje CSS fajlova tj. stilova.*
* */js – direktorijum u kome se čuvaju JavaScript fajlovi.*
* */img – direktorijum za skladištenje slika*
* *.htaccess (hypertext access) – konfiguracioni fajl*
* *Index.php – glavna skripta koja pokreće aplikaciju*

U folderu **/system** se nalaze svi fajlovi vezani za izvršavanje aplikacije

* /models – fajlovi koji služe za rad sa podacima
* /views – fajlovi za prezentaciju podataka
  + /admin – fajlovi za prezentaciju podataka iz admin dela
* /controllers – fjlovi koji prihvataju zahteve I podatke od korisnika
* /includes – fajlovi sa pomoćnim metodama i klasama

## 1.5 Alati

Spisak softverskih alata korišćenih za razvoj ove veb aplikacije.

### 1.5.1 Atom

Za pisanje izvornog koda korišćen je tekstualni editor **Atom**. Atom je besplatan editor otvorenog koda za OS X, Linux i Windows sa podrškom za plug-inove pisane u Node.js-u i ugrađenom Git kontrolom, razvijen od strane GitHub-a. Atom je desktop aplikacija izgrađena na veb tehnologijama; Drugim rečima, nije online veb aplikacija.

### 1.5.2 MySQL Workbench

Zadizaj baze podataka i tabela, korišćen je MySQL Workbench. **MySQL Workbench** je dizajn alat za vizuelni prikaz baza podataka koji integriše SQL razvoj, administraciju, dizajn baza podataka, kreiranje i održavanje u jedano integrisano razvojno okruženje za MySQL sistem baza podataka. On je naslednik DBDesigner-a 4 kreiran od strane fabFORCE.net i zamenjuje prethodni softverski paket, MySQL GUI Tools Bundle.

# 2. Funkcionalnost

## 2.1 Frontend

Tipovi korisnika veb aplikacije, ograničenja i funkcionalnosti koje mogu da koriste.

### 2.1.1 Korisnički deo

Na početnoj stranici korisniku se prikazuje deo sa 4 poslednje destinacije koje su ubačene u ponudu. Ispod se nalazi deo koji služi za predstavljanje turističke agencije i kratku istoriju.

Na stranici *Sve ponude* korisniku se prikazuju sve destinacije iz ponude. Ukoliko ima više ponuda, postoji straničenje (paginacija) koja služi za navigaciju između stranica. Korisnik takođe ima mogućnost da koristi pretragu ponuda. Pretraga se vrši po nazivima destinacija i njihovom opisu.

*Kontakt* stranica prikazuje kontakt informacije, mapu i formu za slanje email poruke turističkoj agenciji. Mapa se prikazuje pomoću Google Maps servisa.

Na stranici pojedinačne destinacije, korisnik može da pročita ceo opis destiancije i klikom na dugme ‘Rezerviši’ otvara novu stranicu sa formom za rezervaciju. Osim prisutnosti obaveznih podataka (ime, telefon i email) prilikom slanja forme za rezervaciju proverava se i preostali broj slobodnih aranžmana. Ukoliko je sve u redu, rezervacija se zapisuje u tabelu ‘reservations’.

### 2.1.2 Administracija

Administrator se prijavljuje na sistem pomoću korisničkog imena i lozinke. Podaci koje je korisnik poslao se porede sa podacima iz tabele ‘users’, gde se lozinka pre provere hešuje pomoću php funkcije md5().

Ukoliko je logovanje na sistem uspešno, korisničko ime se zapisuje u $\_SESSION[‘username’] promenljivu i korisniku se preikazuje deo za pregled destinacija, opcija za dodavanje novih destinacja, za izmenu postojećih i za brisanje destinacija.

U delu za dodavanje novih destinacija korisnik može da navede ime destinacije, opis, sliku, ukupan broj aranžmana za tu destinaciju i datum od kad do kad može da se poseti destinacija.

Administrator takođe ima uvid u sve rezervacije, kao i mogućnost da ih obriše. Da bi se sprečilo neželjeno brisanje podataka iz baze, uveden je dodatni korak za potvrdu u vidu JavaScript popup prozora.

## 2.2 Backend

Tehnologije i funkcionalnosti korišćene za razvoj pozadinskog dela aplikacije.

### 2.2.1 Rutiranje (Routing)

Pored implementacije MVC arhitekture, za olakšani razvoj aplikacije uveden i je sistem rutiranja. U .htaccess fajlu je podešeno da svi zahtevi idu preko index.php fajla, u njemu se instancira Route klasa i na osnovu vrednosti niza $\_SERVER[’ REQUEST\_URI’] ona određuje koji deo aplikacije će biti izvršen.

U direktorijimu ‘*system/includes/routes.php*’ su definisane sve rute koje se koriste u aplikaciji kao i klase i metode koje je potrebno pozvati ukoliko korisnik zatražu tu adresu. Ako korisnik zatraži adresu koja nike definisana i ovom fajlu, korisniku se prikazuje stranica sa greškom 404 (page not found).

Cilj ovoga je da se naprave lepi (eng. pretty) URLs, tj. da se umesto klasicnih ruta kao što je npr. *tourizm.app/rezervisi.php?id=8* napravi *tourizm.app/rezervisi/8.*

Nove rute se definišu tako što se metodi $route->add() proslede dva argumenta, naziv rute npr. „/nova/ruta“ i ime kontrolera i metode „Kontroler@metoda“. Kao alternativa, ukoliko ne želimo da pozivamo kontroler, metodi add() možemo proslediti i anonimnu funkciju (anonymous function).

Ruta koja prosleđuje argumente za metodu kontrolera se definiše tako što se na kraj putanje doda ključna reč /(:num) ili /(:any) ukoliko prosleđujemo integer vrednost ili string, respektivno.

### 2.2.2 Template sistem (View fajlovi)

Cilj ove funkcionalnosti je da dodatno olakša razvoj frontend-a od logike aplikacije. Na ovaj način će u budućnosti biti lakše izmeniti neki vizuelni deo aplikacije, a da se pri tome ne ugrozi postojeća dunkcionalnost.

Template sistem radi tako što sve podatke koje je potrebno ispisati na stranici čuva u promenljivoj koja se pomoću glavnog kontrolera (Core controller) prosleđuje View fajlovima.

Svaka stranica ima svoj view fajl sa odgovarajućim imenom koji se učitava u glavni (main) view fajl. U glavnom view fajlu su definisane putanje ka resursima kao što su css i javascript fajlovi, glavna struktura HTML-a, header i footer.

# 3. Zaštita

## 3.1 Zaštita izvornog koda i logike aplikacije

Zbog bolje zaštite od nedozvoljenog pristupa, sva logika aplikacije je smeštena u folder ‘system’ koji je izmešten van root direktorijuma.

Kao dodatna zaštita, na početku svake php skripte se proverava odakle je skripta pozvana i zabranjuje se direktan pristup fajlu, tako što se proverava da li je definisana konstanta ‘BASEPATH’.

## 3.2 Autorizacija i autentifikacija

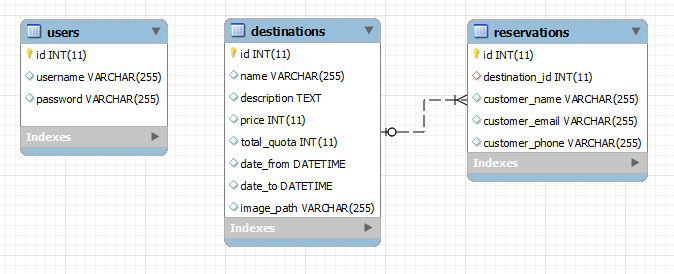
Prilikom prikazivanja stranica iz admin dela, vrši se provera da li je korisnik prijavljen na sistem. Provera se vrši tako što se proverava prisutnost vrednosti $\_SESSION[’username’]. Ukoliko provera vrati *false,* korisnik se preusmerava na stranicu za logovanje na sistem. U suprotnom korisnik nesmetano nastavlja da koristi sistem.

## 3.3 Obrada podataka

Pri slanju svake form radi se validacija i normalizacija ulaznih podataka. Proverava se da li su popunjena sva obavezna polja i ako jesu radi se prevencija SQL Injection napada pomoću mysqli funkcije real\_escape\_string().

# 4. Baza podataka

## 4.1 UML dijagram

***Slika 1.*** *Dijagram UML profila za reprezentaciju šeme relacione baze podataka.*

## 4.2 Tabele

Baza je sačinjena od tri tabele.

**Destinations** (id, name, description, price, total\_quota, date\_from, date\_to)

CREATE TABLE `tourizm`.`**destinations**` (

`id` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(255) NULL,

`description` TEXT NULL,

`price` INT NULL,

`image\_path` VARCHAR(255) NULL,

`date\_from` DATETIME NULL,

`date\_to` DATETIME NULL,

`total\_quota` INT NULL,

PRIMARY KEY (`id`));

**Users** (id, username, password)

CREATE TABLE `tourizm`.`**users**` (

`id` INT NOT NULL AUTO\_INCREMENT,

`username` VARCHAR(255) NULL,

`password` VARCHAR(255) NULL,

PRIMARY KEY (`id`));

**Reservations** (id, destination\_id, customer\_name, customer\_email, customer\_phone)

CREATE TABLE `tourizm`.`reservations` (

`id` INT NOT NULL AUTO\_INCREMENT,

`destination\_id` INT NULL,

`customer\_name` VARCHAR(255) NULL,

`customer\_email` VARCHAR(255) NULL,

`customer\_phone` VARCHAR(255) NULL,

PRIMARY KEY (`id`));

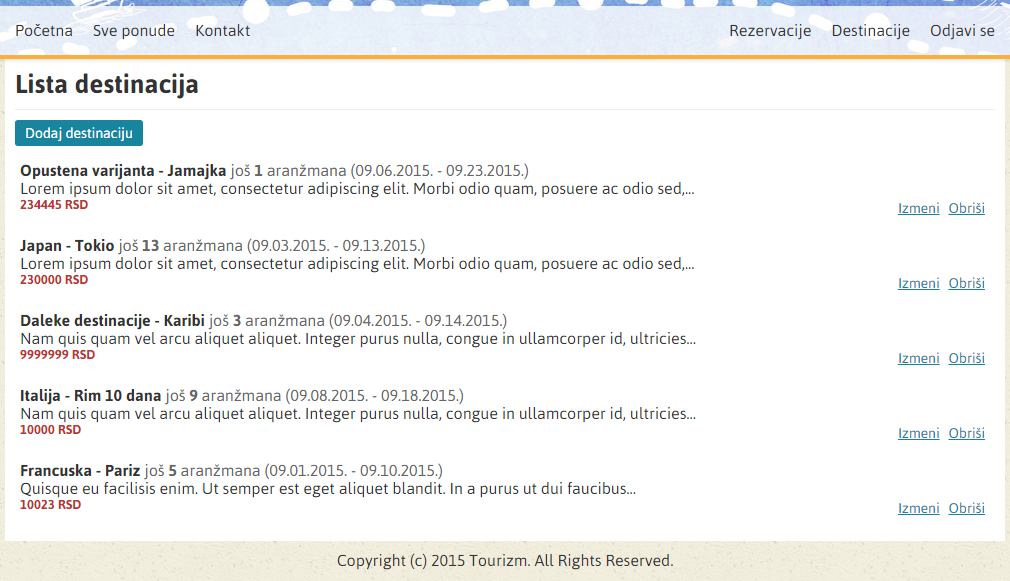
# 5. Prilog

Prikaz dela aplikacije i korisničkog interfejsa.

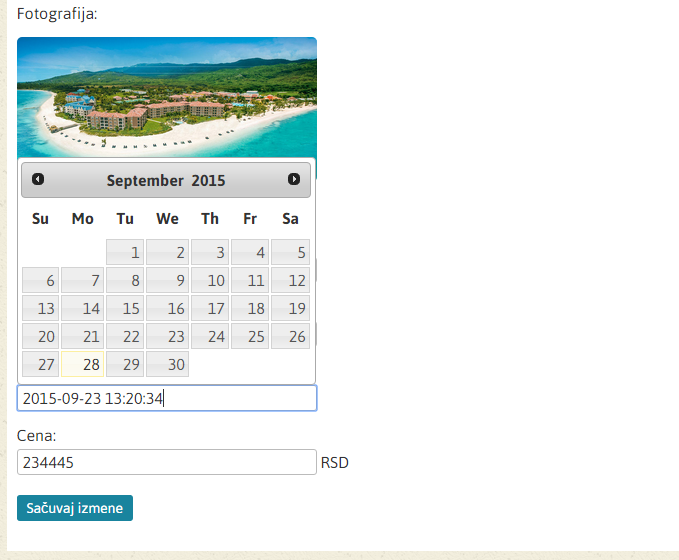
## 5.1 Frontend



***Slika 2.*** *Prikaz stranice za pristup sistemu i poruke za korisnika ukoliko nije popunio sva polja prilikom prijave.*



***Slika 3.*** *Prikaz administratorskog dela za upravljanje sadržajem.*

**

***Slika 4.*** *Prikaz upotrebe jQuery tehnologije za stilizovanje elemenata za odabir datuma.*

## 5.2 Backend

Metoda za **čuvanje** podataka. Primer je sveden ma minimum. *(/system/models/destination.php)*

public function add\_destination()

{

// Check input

if ( $\_POST['name'] == '' )

{

// Custom method for database connection

$conn = $this->connect\_to\_db();

// Prevent SQL Injection

$name = $conn->real\_escape\_string( $\_POST['name'] );

$sql = "INSERT INTO destinations ( name ) VALUES ( '$name' )";

$result = $conn->query( $sql );

}

}

Metoda za **konektovanje** na MySQL bazu podataka. *(/includes/app.class.php)*

public function connect\_to\_db()

{

// Database connection

$db\_user = "root";

$db\_pass = "";

$hostname = "localhost";

$db\_name = "tourizm";

$conn = new mysqli( $hostname, $db\_user, $db\_pass, $db\_name );

if ( $conn->connect\_error ) {

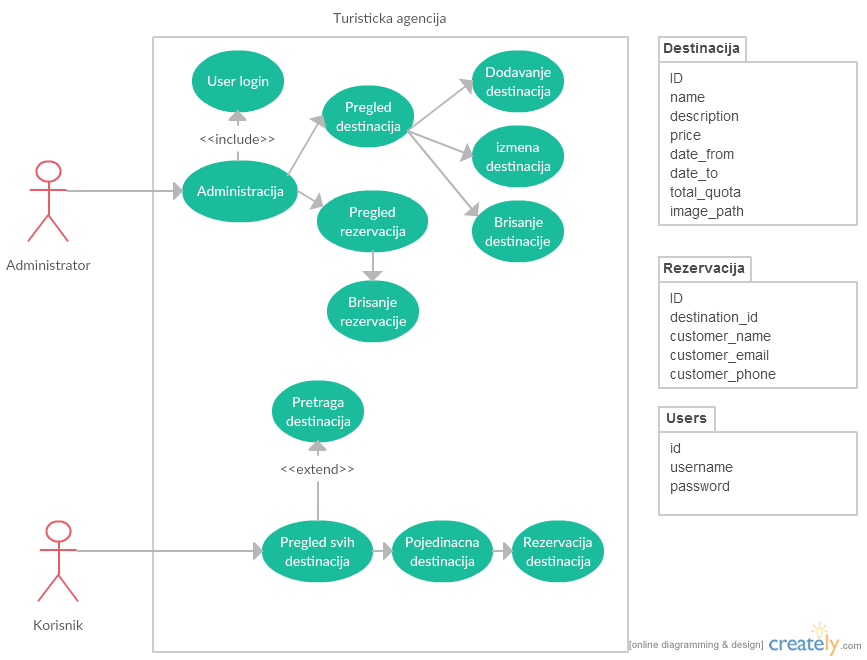
die( "Connection failed: " . $conn->connect\_error ) ;

}

return $conn;

}

## 5.3 UML use case diagram



***Slika 5.*** *UML dijagram slučajeva korišćenja.*