# Funkcionalnost

## Korisnik

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 stranoici *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’.

## Administracija

Korisnik 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 system 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. Takođe mu je ponuđena i opcija za pregled svih rezervacija.

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.

# Zaštita

## Provera autorizacije

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 *fale,* korisnik se preusmerava na stranicu za logovanje na sistem. U suprotnom korisnik nesmetano nastavlja da koristi sistem.

## 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().

## Baza podataka

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`));

# Prilog

## Backend

Primer php koda za **čuvanje** podataka . Primer je sveden ma minimum. *(/admin/add-destination.php)*

if ( !empty( $\_POST ) )

{

// Check input

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

{

// Custom method for database connection

$conn = 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/helpers.php)*

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;

}

## UML use case diagram

