**VISOKA TEHNIČKA ŠKOLA**

**STRUKOVNIH STUDIJA**

**S** **U B O T I C A**

**PROJEKAT**

**iz predmeta Napredno web programiranje**

**STUDENT(I) MENTOR**

**Veljko Bogdan dr Zlatko Čović**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**SUBOTICA 2024. god.**

**SADRŽAJ**

[OPIS ZADATKA 3](#_Toc4765)

[ISTRAŽIVANJE TRŽIŠTA 4](#_Toc13818)

[REALIZACIJA ZADATKA 5](#_Toc18192)

[STRUKTURA BAZE PODATAKA 12](#_Toc32441)

[OPIS FUNKCIONALNOSTI 13](#_Toc16814)

[KORIŠĆENA LITERATURA 22](#_Toc31474)

### OPIS ZADATKA

Cilj projekta je da se kreira web aplikacija-sajt za upravljanje turističkim turama. Aplikacija  razlikuje četiri nivoa pristupa: gost, registrovan/prijavljen korisnik, turistička organizacija  grada i administrator.

Gost

• može da pogleda ponuđene informacije na sajtu

• može da pogleda turističku ponudu gradova

• može da putem detaljne pretrage i upotrebom različitih filtera pretrage pronađe  turističke znamenitosti

• može da izvrši registraciju na sajtu

Registrovan/prijavljen korisnik

• može da pogleda ponudu turističkih ponuda po gradovima

• može da putem detaljne pretrage i upotrebom različitih filtera pretrage pronađe  turističke znamenitosti

• može da kreira, menja i briše podatke u listi omiljenih turističkih lokacija • može da detaljno pogleda informacije o svakoj lokaciji sa opisom, slikama  • može da kreira ture koje sadrže lokacije koje želi da poseti

• može da zahteva promenu lozinke (zaboravljena lozinka)

Turistička organizacija

• može da kreira/menja/briše turističke znamenitosti za određeni grad (mora da poseduje  naziv, opis, fotografije i adresu)

• može da pregleda statistiku popularnosti određenih znamenitosti (upotrebiti filtere za  pretragu)

Administrator

• može da kreira naloge za turističke organizacije

• može da pregleda i briše sve turističke organizacije i njihove znamenitosti • može da kreira, menja i briše gradove

• može da aktivira/deaktivira prikaz podataka za jednu turističku organizaciju • može da zabrani/odobri korisniku pravo prijavljivanja na sistem

Registracija korisnika mora biti urađena na bezbedan način i obaveznim slanjem aktivacionog  linka putem e-maila. Slanje linka koristiti i kod zahteva za promenu lozinke. Ne sme se dozvoliti  registracija korisničkih naloga sa istom e-mail adresom. E-mail adresa mora biti jedinstvena i  ona predstavlja korisničko ime.

### **ISTRAŽIVANJE TRŽIŠTA**

GetYourGuide (https://www.getyourguide.com/):

* Automatsko kreiranje rute grada
* Lični Wishlist
* Sortiranje po kategorijama znamenitosti

Prednosti: Personalizovani predlozi, rute

Mane: Veoma mnogo informacija tokom pregleda, loš layout, nema filtera za pretragu

TourScanner (https://tourscanner.com/):

* Lični Wishlist
* Ponuda u blizini
* Mobilna aplikacija

Prednosti: Uredan sajt, personalizovani predlozi

Mane: Nema filtera za pretragu, neintuitivan dizajn

Viator (https://www.viator.com/):

* Biranje jezika i valute
* Biranje datuma polaska

Prednosti: Uredan sajt, personalizovani predlozi, velika ponuda kategorija

Mane: Nema filtera za pretragu

Tri dodatne funkcionalnosti:

* Biranje rute: Korisnik može pomoću mape da izabere rutu kojom želi da se kreće
* Omiljeni gradovi i znamenitosti: Korisnik će moći da sačuva svoje omiljene lokacije
* Mapa: Korisnik će moći da bira gradove na mapi i vidi njihove ponude

### REALIZACIJA ZADATKA

**Config.php kod:**

<?php  
  
$options = [  
 'cost' => 10  
];  
  
const PARAMS = [  
 "HOST" => 'localhost',  
 "USER" => 'root',  
 "PASSWORD" => '',  
 "DB" => 'bogdan',  
 "CHARSET" => 'utf8mb4'  
];  
  
$dsn = "mysql:host=" . PARAMS['HOST'] . ";dbname=" . PARAMS['DB'] . ";charset=" . PARAMS['CHARSET'];  
  
$pdoOptions = [  
 PDO::ATTR\_ERRMODE => PDO::ERRMODE\_EXCEPTION,  
 PDO::ATTR\_DEFAULT\_FETCH\_MODE => PDO::FETCH\_ASSOC,  
 PDO::ATTR\_EMULATE\_PREPARES => false  
];  
  
$messages = [  
  
];

**Povezivanje sa bazom:**

$GLOBALS['pdo'] = connectDatabase($dsn, $pdoOptions);  
  
/\*\* Function connects to database using PDO  
 \* @param string $dsn  
 \* @param array $pdoOptions  
 \* @return PDO  
 \*/  
function connectDatabase(string $dsn, array $pdoOptions): PDO  
{  
  
 try {  
 $pdo = new PDO($dsn, PARAMS['USER'], PARAMS['PASSWORD'], $pdoOptions);  
 } catch (\PDOException $e) {  
 var\_dump($e->getCode());  
 throw new \PDOException($e->getMessage());  
 }  
  
 return $pdo;  
}

**Leaflet.js kod za kreiranje mape:**

var map = L.map('map').setView([0, 0], 2);  
  
L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {  
 attribution: '&copy; OpenStreetMap contributors'  
}).addTo(map);

**Kreiranje markera**

function onMapClick(e) {  
 var clickLatLng = e.latlng;  
  
 var marker = L.marker(clickLatLng).addTo(map);  
 markers.push(marker);  
  
 updateRoutingControl();  
  
 fetch('get\_cities.php')  
 .then(response => response.json())  
 .then(data => {  
 // Find the nearest city  
 var nearestCity = findNearestCity(clickLatLng, data.cities);  
  
 // Display city information in a card or alert  
 if (nearestCity) {  
 displayCityInfo(nearestCity);  
 }  
 })  
 .catch(error => {  
 console.error('Error fetching city data:', error);  
 });  
}

**Kreiranje ruta**  
function updateRoutingControl() {  
 // Remove old routing control  
 if (routingControl) {  
 map.removeControl(routingControl);  
 }  
  
 // Create a route between all markers  
 if (markers.length >= 2) {  
 document.getElementById("saveRoute").style.display = "none";  
 var waypoints = markers.map(function (marker) {  
 return L.latLng(marker.getLatLng().lat, marker.getLatLng().lng);  
 });  
  
 routingControl = L.Routing.control({  
 waypoints: waypoints  
 }).addTo(map);  
 }  
}

**Brisanje mape**  
function clearMap(){  
 if (routingControl) {  
 map.removeControl(routingControl);  
 }  
 markers.forEach(function (marker) {  
 map.removeLayer(marker);  
 });  
  
 markers = [];  
  
 document.getElementById('cityCardsContainer').innerHTML = '';  
 document.getElementById("saveRoute").style.display = "none";  
}

**Logika za najbliži grad**  
function findNearestCity(clickedLatLng, cities) {  
 var nearestCity = null;  
 var minDistance = Infinity;  
  
 for (var i = 0; i < cities.length; i++) {  
 var cityLatLng = L.latLng(cities[i].latitude, cities[i].longitude);  
 var distance = clickedLatLng.distanceTo(cityLatLng);  
  
 if (distance < minDistance) {  
 minDistance = distance;  
 nearestCity = cities[i];  
 }  
 }  
  
 return nearestCity;  
}

**Logika za prikazivanje grada**  
function displayCityInfo(city) {  
 // Create a new card element  
 var cityCard = document.createElement('a');  
 cityCard.className = 'recommended-cities-a';  
 cityCard.href = `city.php?id=${city.id\_city}`;  
  
 // Update the content of the card  
 cityCard.innerHTML = `  
 <div class="recommended-cities city-card">  
 <h2>${city.name}, ${city.country}</h2>  
 <p>Population: ${city.population}</p>  
 </div>  
 `;  
  
 var container = document.getElementById('cityCardsContainer');  
 container.insertBefore(cityCard, container.firstChild);  
  
  
}  
  
map.on('click', onMapClick);  
map.on('contextmenu', clearMap);

**Kod za flitriranje pretrage:**

<?php  
if (isset($\_GET['search'])) {  
 $search = trim(strip\_tags($\_GET['search-input']));  
 $category = null;  
 $location = null;  
 $fee = null;  
 $any = false;  
 if($search == "" || $search == "Search for a place"){  
 $search = null;  
 }  
 if(isset($\_GET['categories']) && $\_GET['categories'] != "") {  
 $category = $\_GET['categories'];  
 $any = true;  
 }  
 if(isset($\_GET['locations']) && $\_GET['locations'] != "") {  
 $location = $\_GET['locations'];  
 $any = true;  
 }  
 if(isset($\_GET['fee']) && $\_GET['fee'] != "") {  
 switch ($\_GET['fee']){  
 case -1: $fee = "> -1"; break;  
 case 0: $fee = "< 10"; break;  
 case 1: $fee = "< 50"; break;  
 case 2: $fee = "> 50"; break;  
 }  
 $any = true;  
 }  
 $sql = "SELECT \* FROM sights" .  
 ($category == null ? "" : " WHERE id\_category = " . $category) .  
 ($location == null ? "" : ($category == null ? " WHERE" : " AND") . " id\_city = " . $location) .  
 ($fee == null ? "" : ($category == null && $location == null ? " WHERE" : " AND") . " fee " . $fee) .  
 ($search == null ? "" : ($category == null && $location == null && $fee == null ? " WHERE" : " AND") . " name LIKE '%" . $search . "%'");  
  
 $query = $pdo->query($sql);  
  
 while ($row = $query->fetch(PDO::FETCH\_ASSOC)) {  
 echo '<a class="recommended-cities-a" href="sight.php?id='.$row['id\_sight'].'">  
 <div class="recommended-cities city-card">  
 <h2>'.$row['name'].', <br></h2><h4> '.$row['address'].'</h4>  
 <p>Description: '.$row['description'].'</p>  
 </div>  
 </a>';  
 }  
}  
?>

**Kod za prikazivanje 5 nasumičnih gradova ispod mape**

$query = $pdo->query('SELECT \* FROM cities ORDER BY RAND() LIMIT 5');  
  
if (isset($row['image']) && $row['image'] != 0){  
 $placeholder = $row['image'];  
}  
else{  
 $placeholder = "../uploads/placeholder.jpg";  
}  
  
  
while ($row = $query->fetch(PDO::FETCH\_ASSOC)) {  
 echo  
 '<a class="recommended-cities-a" href="city.php?id='.$row['id\_city'].'"><div class="recommended-cities city-card" style="background-size: 100% 150%; background-image: url('.$placeholder.')">   
 <h2>'.$row['name'].'</h2>  
 <p>'.$row['country'].'</p>  
 <h3>'.$row['region'].'</h3>  
 </div></a><br>'  
 ;  
}

**Ban\_check.php kod**

<?php  
require\_once 'functions.php';  
  
if (isset($\_SESSION['email'])) {  
 $email = $\_SESSION['email'];  
  
 // Prepare and execute the query  
 $stmt = $pdo->prepare("SELECT \* FROM users WHERE email = :email");  
 $stmt->bindParam(":email", $email);  
 $stmt->execute();  
  
 $result = $stmt->fetch(PDO::FETCH\_ASSOC);  
  
 if ($result) {  
 $isBanned = $result['is\_banned'];  
  
 if ($isBanned) {  
 // Destroy the session to log the user out  
 session\_unset();  
 session\_destroy();  
  
 // Redirect the user to "login.php"  
 echo '<script> alert("You are banned. Contact the admin for further information"); </script>';  
 header("Location: login.php");  
 exit; // Terminate the current script to prevent further execution  
 }  
 }  
  
 $stmt = $pdo->prepare("SELECT \* FROM agencies WHERE email = :email");  
 $stmt->bindParam(":email", $email);  
 $stmt->execute();  
  
 $result = $stmt->fetch(PDO::FETCH\_ASSOC);  
  
 if ($result) {  
 $isBanned = $result['is\_banned'];  
  
 if ($isBanned) {  
 // Destroy the session to log the user out  
 session\_unset();  
 session\_destroy();  
  
 // Redirect the user to "login.php"  
 echo '<script> alert("You are banned. Contact the admin for further information"); </script>';  
 header("Location: login.php");  
 exit; // Terminate the current script to prevent further execution  
 }  
 }  
}

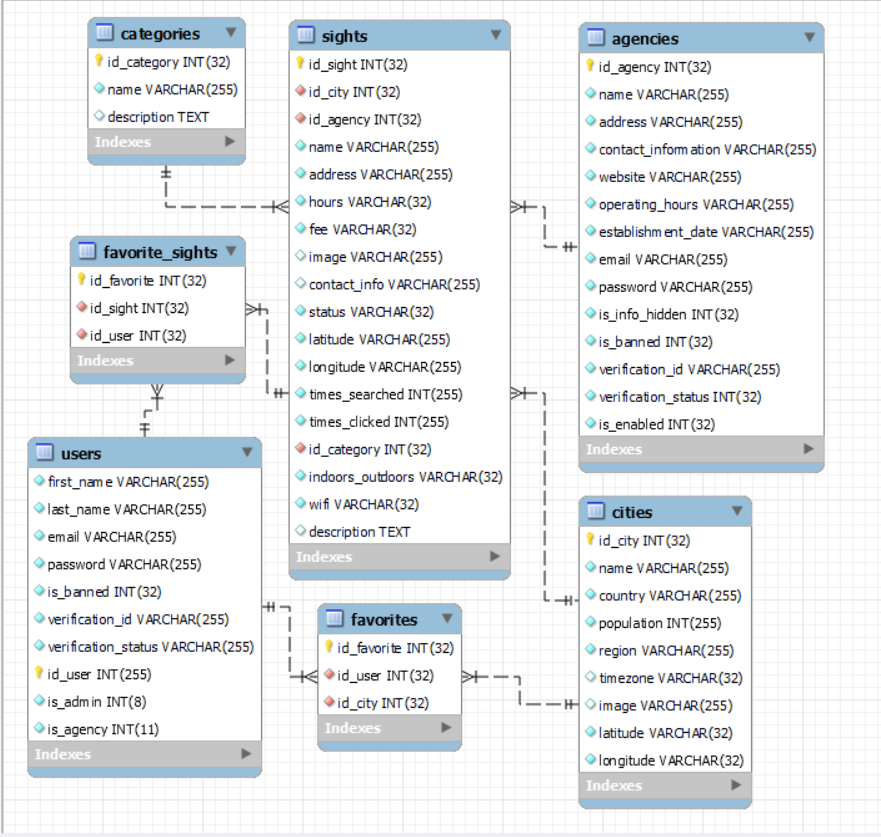
**Funkcija za ubacivanje registrovanog korisnika u bazu**

function insertRegistrationIntoTable(string $name, string $surname, string $email, string $pass, string $v\_id): bool  
{  
 $sql = "INSERT INTO users(first\_name, last\_name, email, password, verification\_id)   
 VALUES(:first\_name, :last\_name, :email, :password, :verification\_id)";  
  
 $stmt = $GLOBALS['pdo']->prepare($sql);  
 $stmt->bindValue(':first\_name', $name, PDO::PARAM\_STR);  
 $stmt->bindValue(':last\_name', $surname, PDO::PARAM\_STR);  
 $stmt->bindValue(':email', $email, PDO::PARAM\_STR);  
 $stmt->bindValue(':password', $pass, PDO::PARAM\_STR);  
 $stmt->bindValue(':verification\_id', $v\_id, PDO::PARAM\_STR);  
  
 return $stmt->execute();  
}

**Funkcija za slanje mail-a**

function sendMail($email, $v\_cod){  
 require ('vendor/PHPMailer/PHPMailer/src/PHPMailer.php');  
 require ('vendor/PHPMailer/PHPMailer/src/Exception.php');  
 require ('vendor/PHPMailer/PHPMailer/src/SMTP.php');  
  
 $mail = new PHPMailer(true);  
  
 try {  
 $mail->isSMTP();  
 $mail->Host = 'sandbox.smtp.mailtrap.io';  
 $mail->SMTPAuth = true;  
 $mail->Port = 2525;  
 $mail->Username = '01484a2ec4eabe';  
 $mail->Password = '2c38b334d9cd44';  
  
 $mail->setFrom('tourismroutestest@gmail.com', 'EmailTest');  
 $mail->addAddress($email);  
  
 $mail->isHTML(true);  
 $mail->Subject = 'Email Verification for Your Account';  
 $mail->Body = "<p class=\"text-center\">Thank You for registering!<br>Click the link below to verify your email address:<br>  
 <a href='http://localhost:63342/TourismRoutes/verify.php?email=$email&v\_cod=$v\_cod'>Verify</a></p><br><hr><br>If this is not you, ignore this email.";  
 $mail->send();  
 return true;  
 } catch (Exception $e) {  
 echo "<script>  
 alert('Exception thrown');  
 </script>";  
 return false;  
  
 }  
}

### STRUKTURA BAZE PODATAKA



### OPIS FUNKCIONALNOSTI

**Početna stranica**

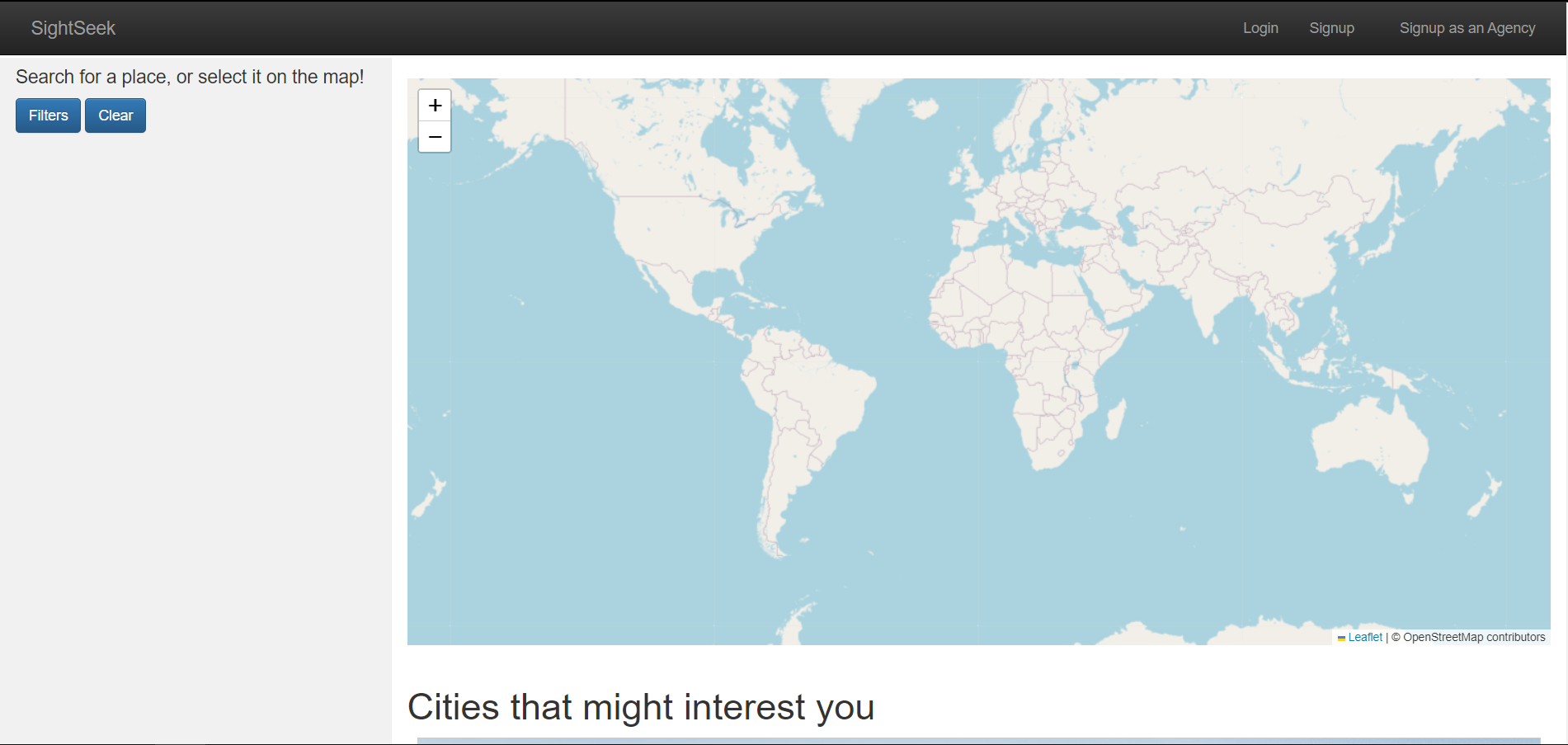


Image 1 - Index

Kada kliknemo na mapu izbaci nam najbliži grad našem markeru.

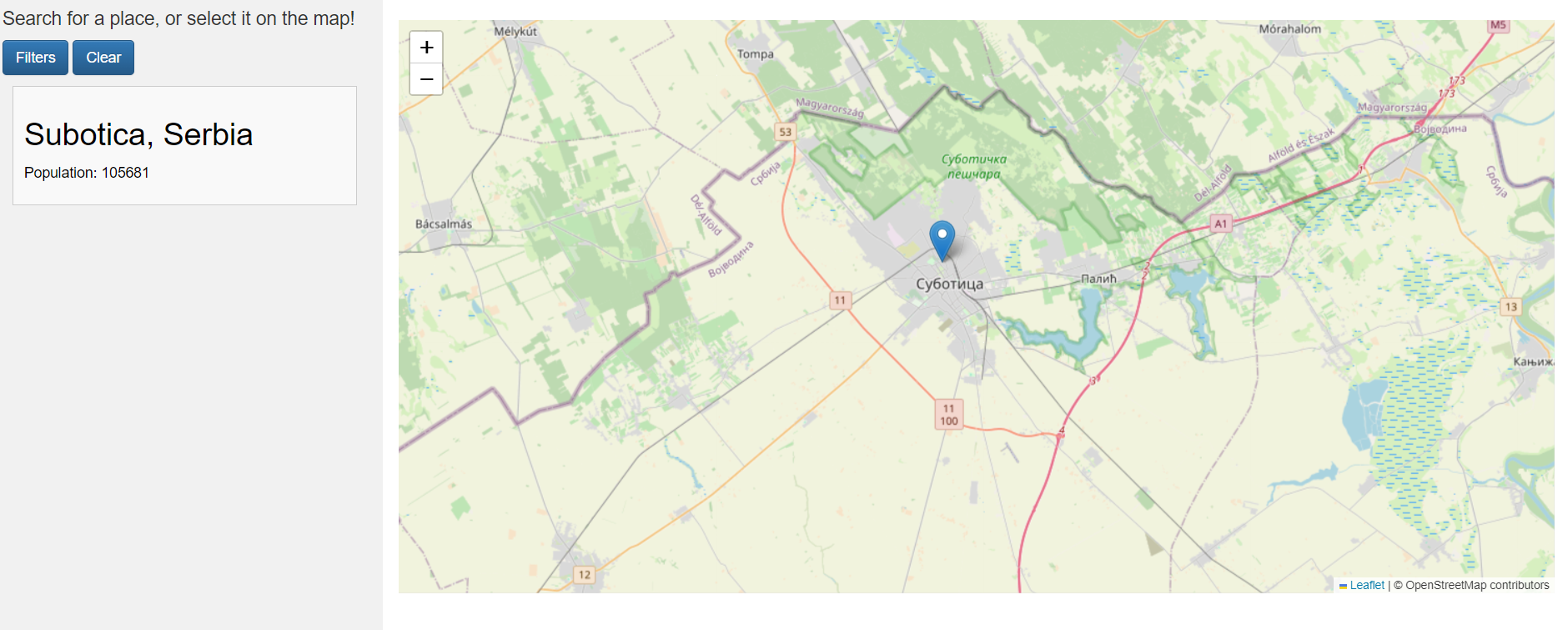


Image 2 - Marker

Ako izaberemo dva markera, stvori se putanja od jednog do drugog.

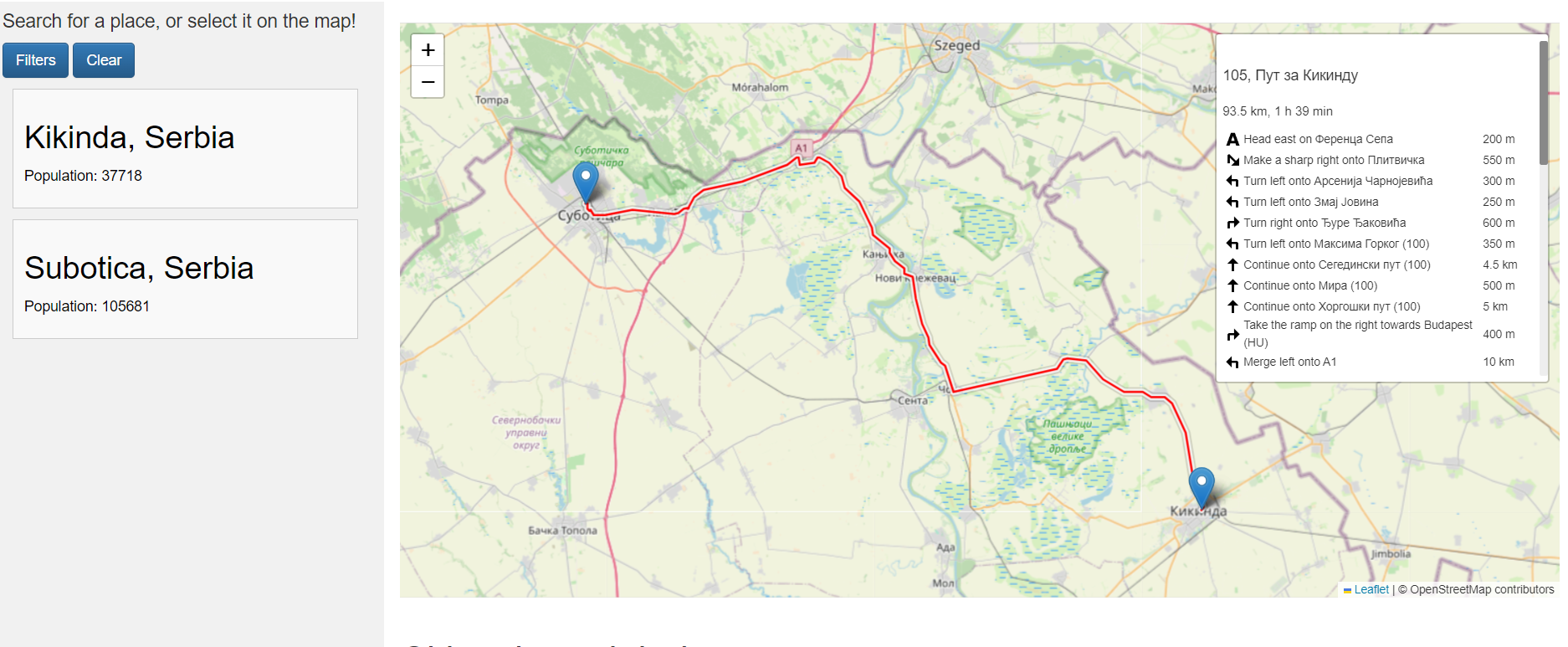


Image 3 - Putanja

Ako stisnemo na Filters dugme, izađe nam forma.

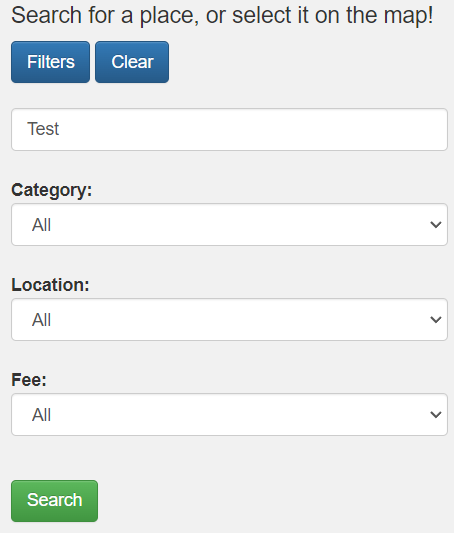


Image 4 - Filter

Pretragom nam izađu znamenosti koje se podudaraju sa filterima.

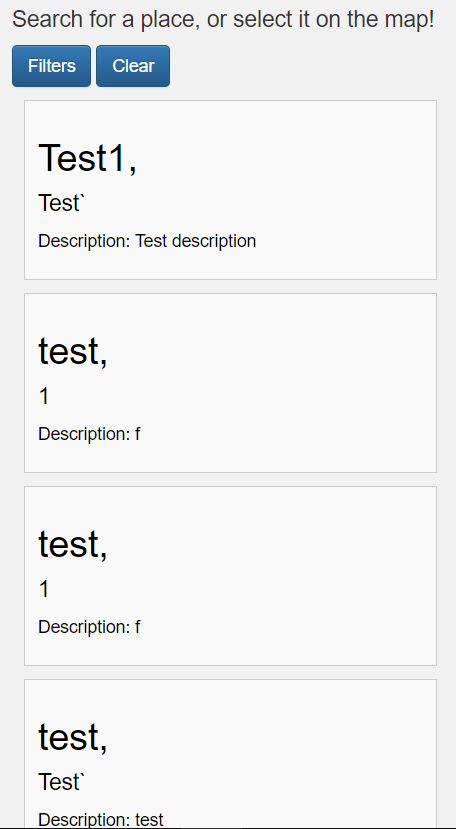


Image 5 - Pretraga

Ispod mape postoje nasumični gradovi koje možemo da pogledamo.

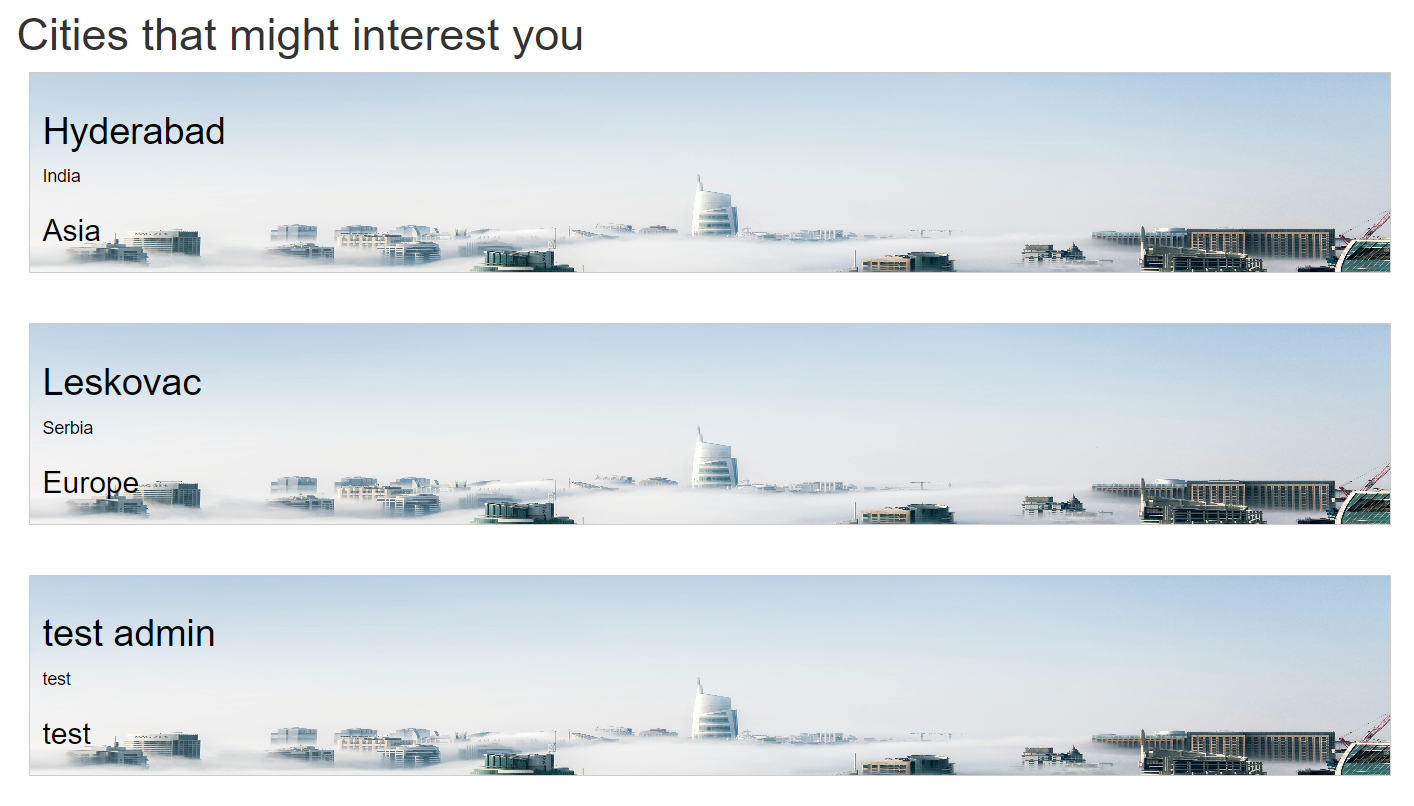


Image 6 - Random gradovi

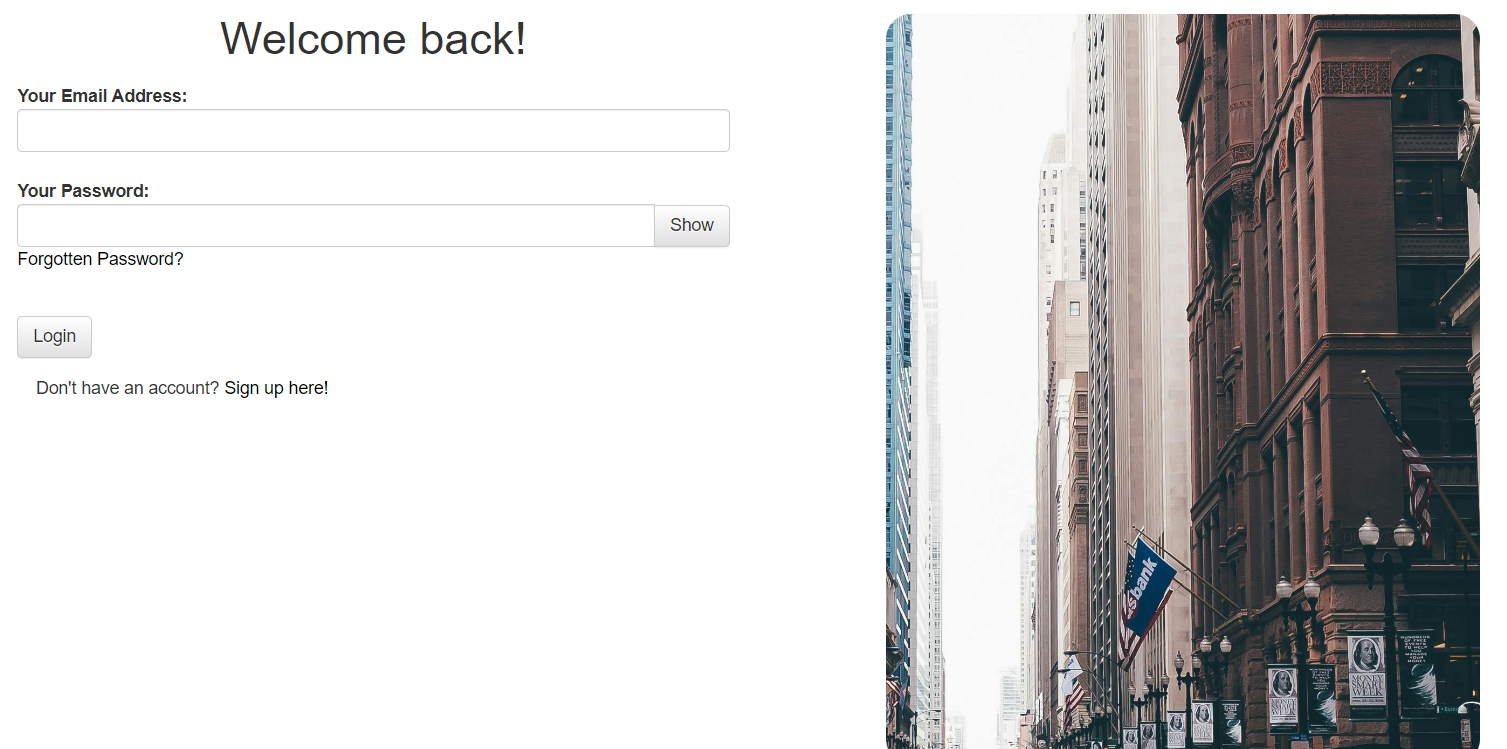


Image 7 - Login

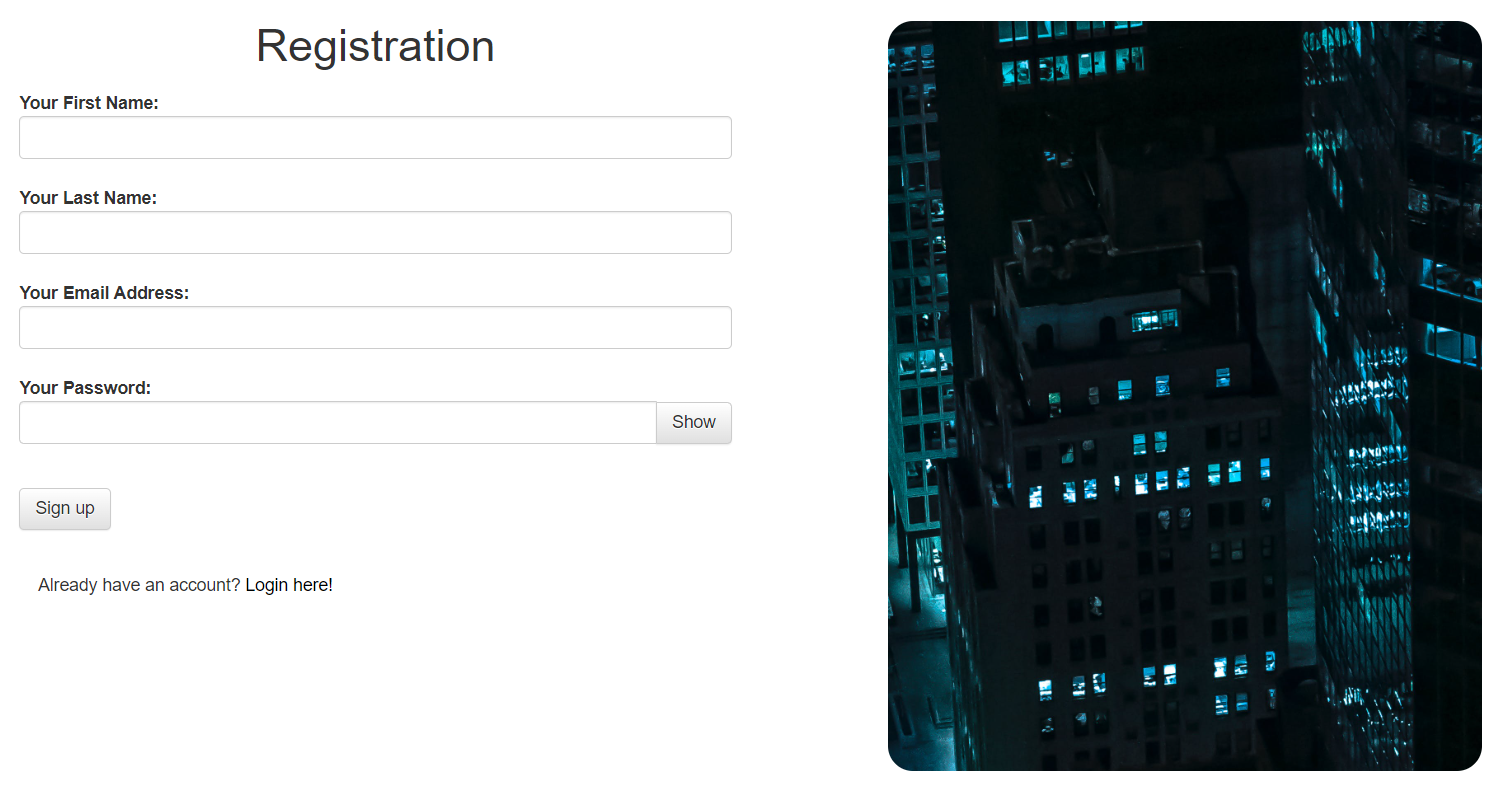


Image 8 - Registracija

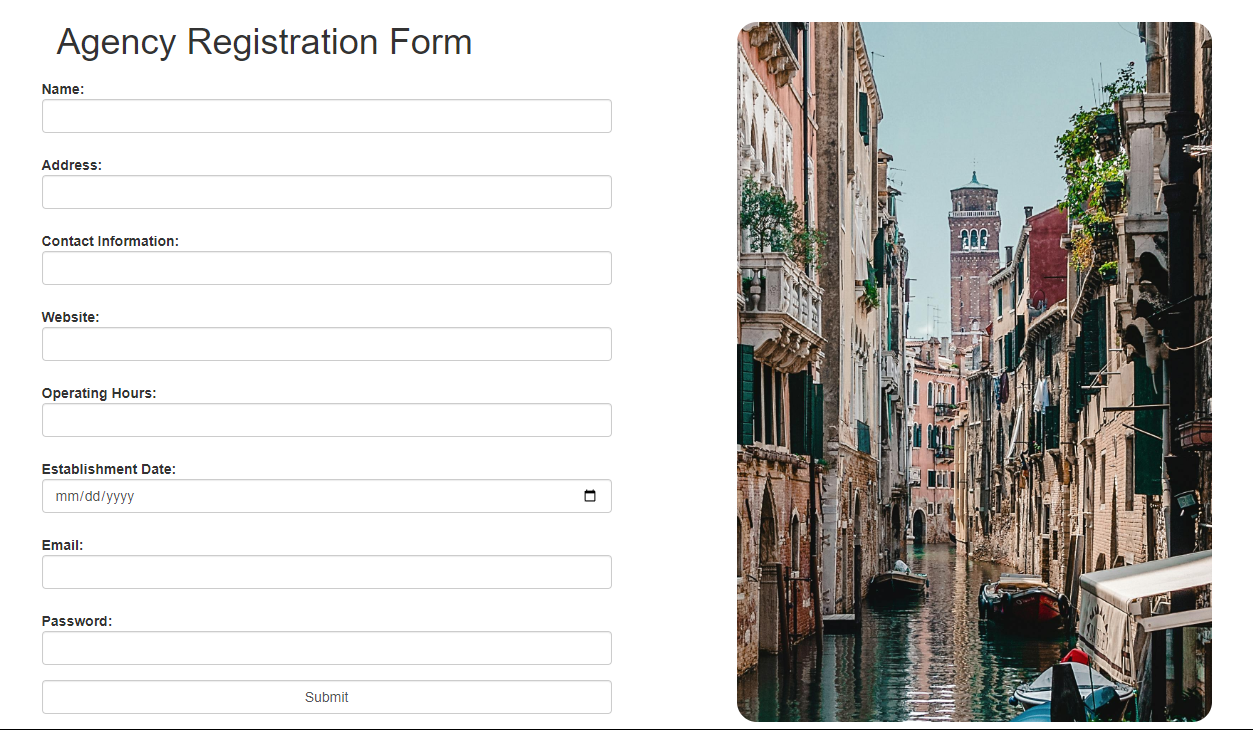


Image 9 - Registracija za agenciju

Kada smo se ulogovali, možemo da pogledamo i stavljamo gradove i znamenitosti u omiljene.

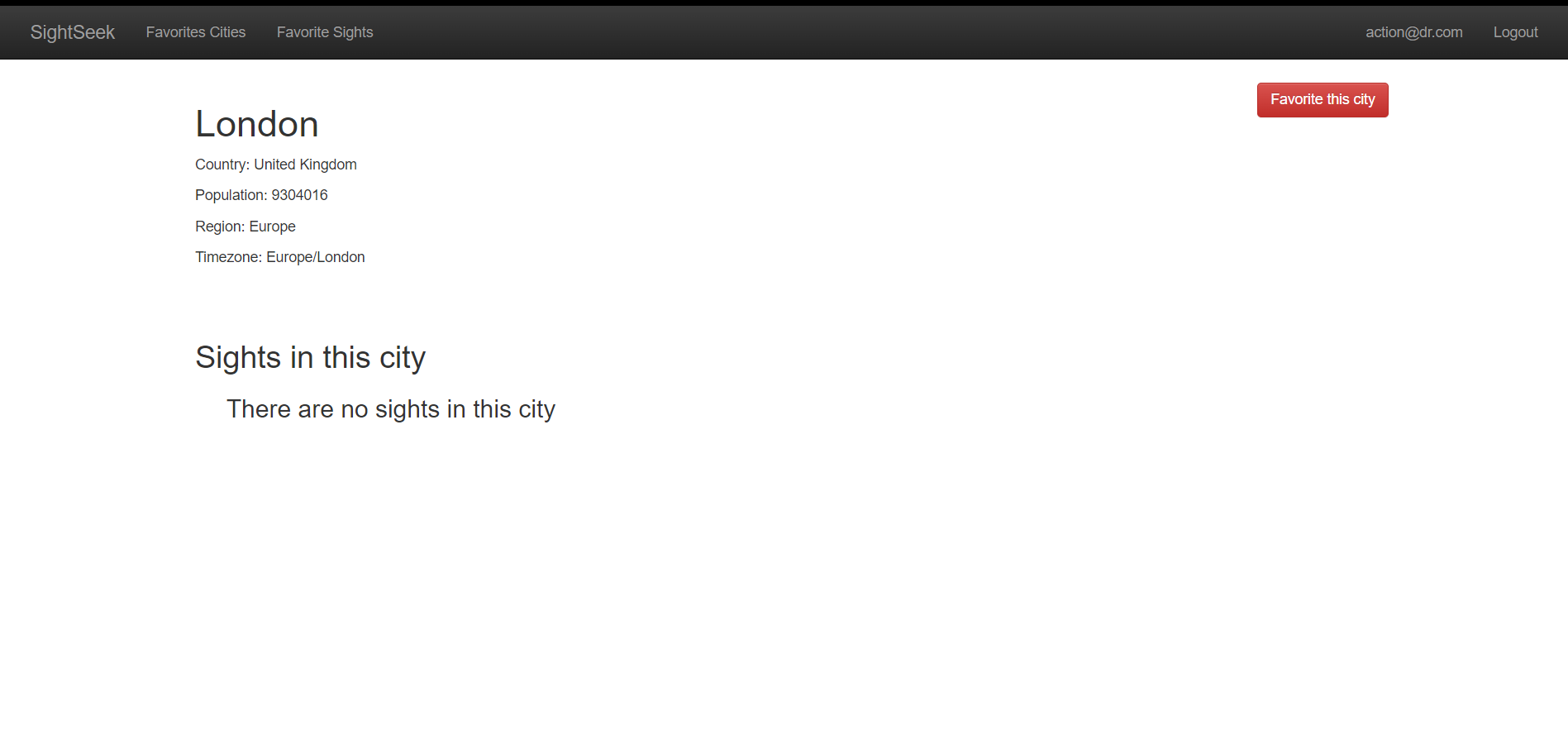


Image 10 - Grad

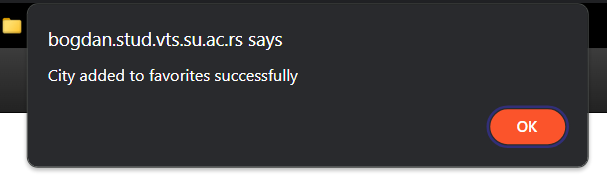


Image 11 - Favorite

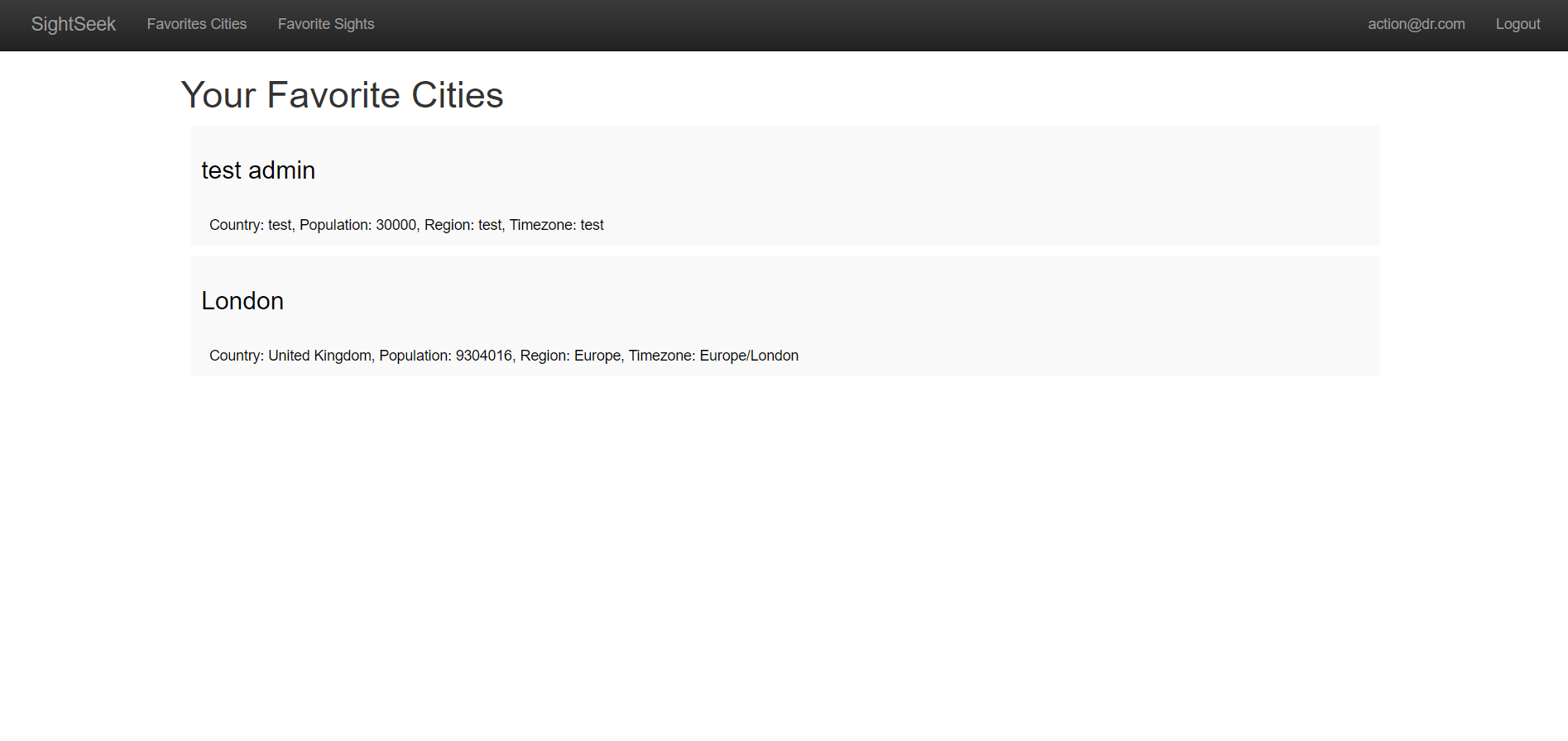


Image 12 - Omiljeni gradovi

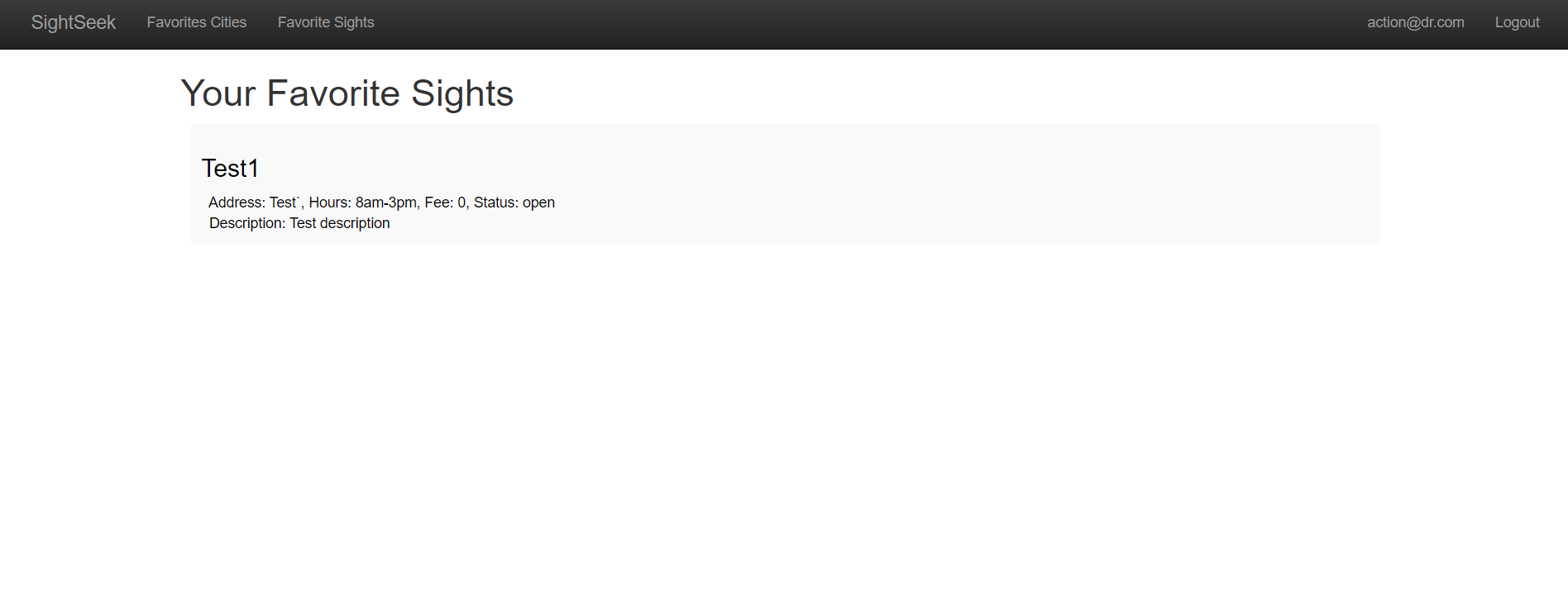
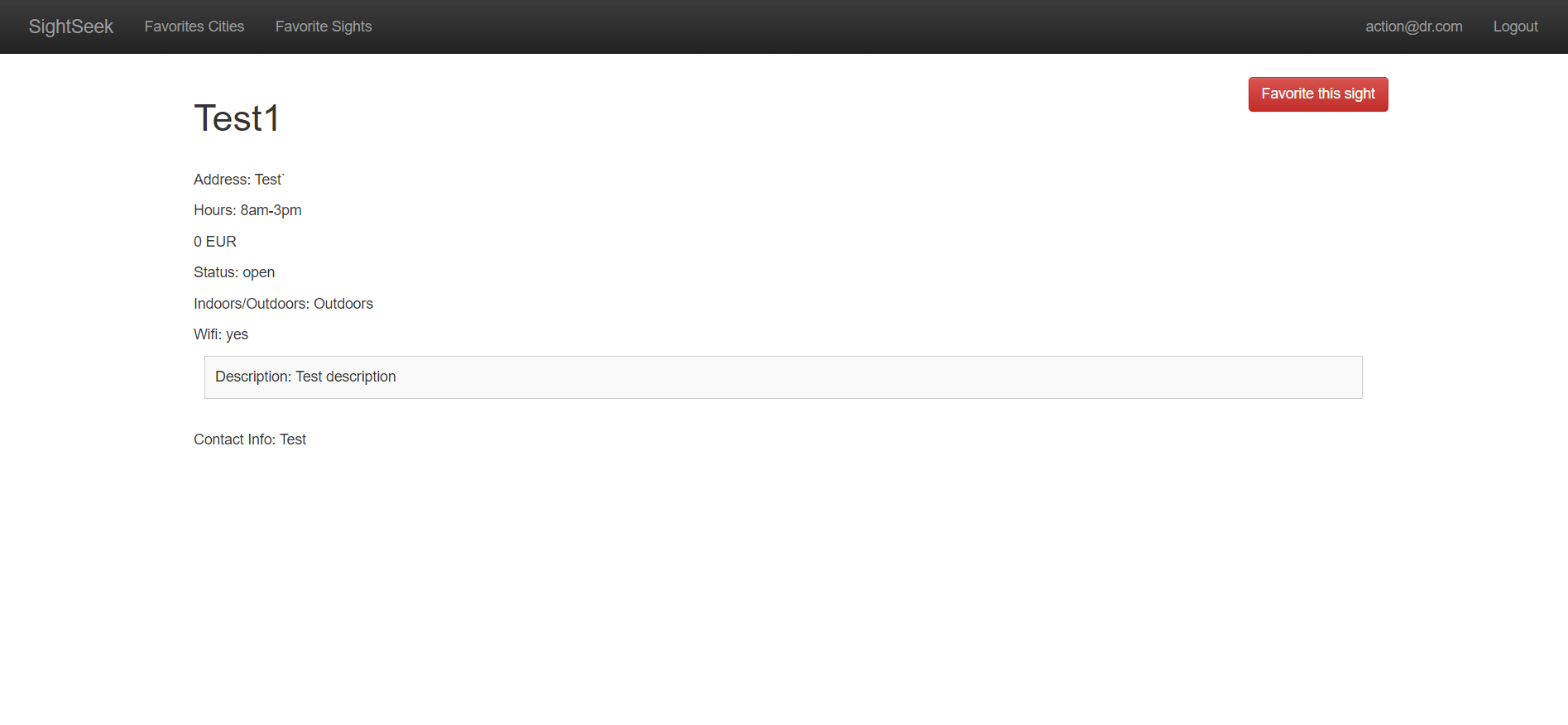


Image 13 - Omiljene znamenitosti



Ako pokušamo da stavimo nešto u omiljene što smo već sačuvali, dobićemo error.

Image 14 - Znamenitost

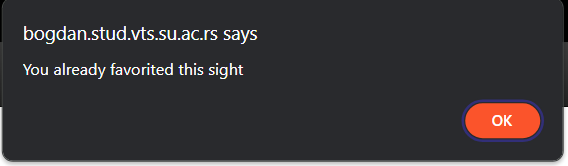


Image 15 - Block

Kada se admin uloguje header izgleda ovako:

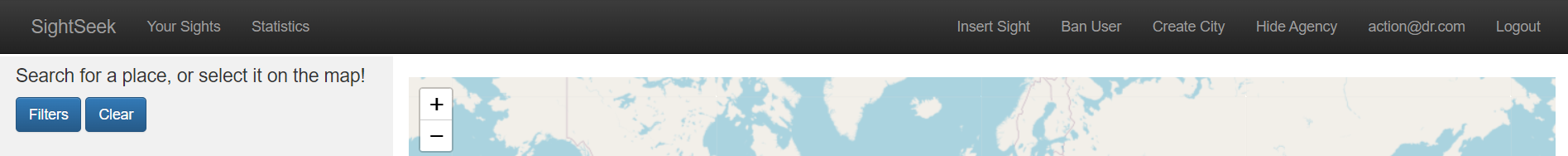


Image 16 - Admin header

Admin može da doda znamenitost, zabrani korisniku da uđe na sajt, kreira grad i isključi prikazivanje podataka agenciji

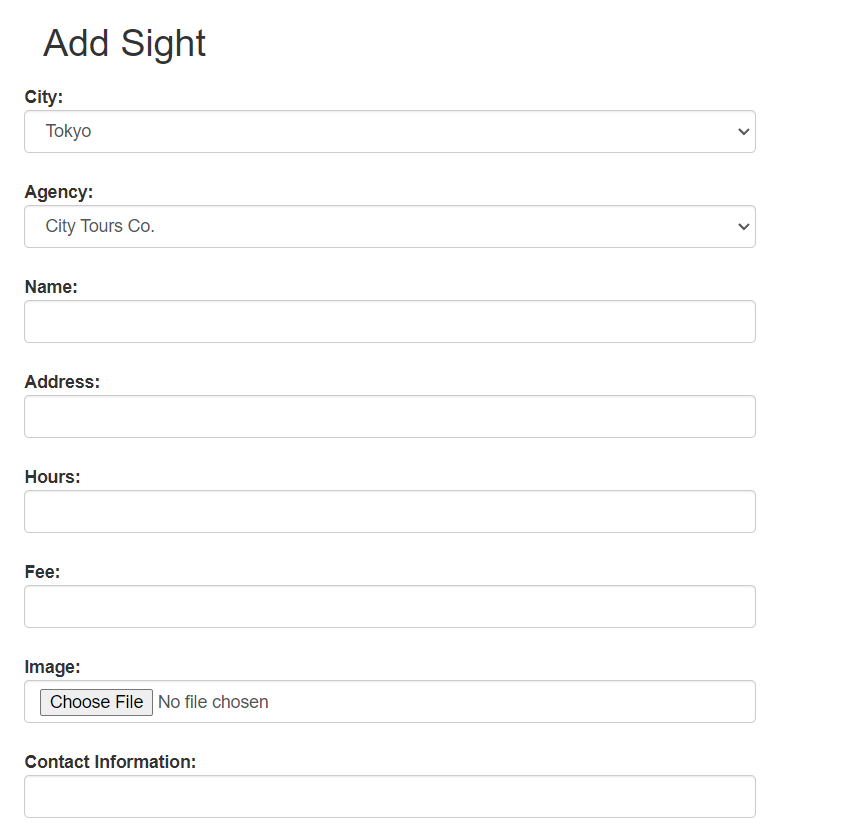


Image 17 - Dodavanje znamenitosti



Image 18 - Banovanje

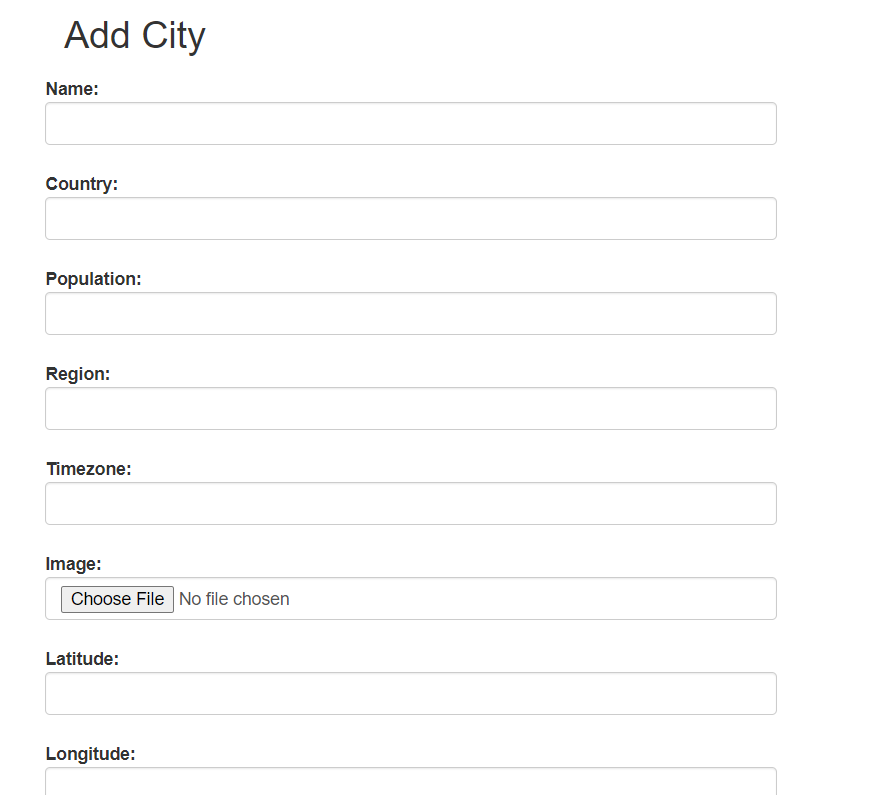


Image 19 - Dodavanje grada

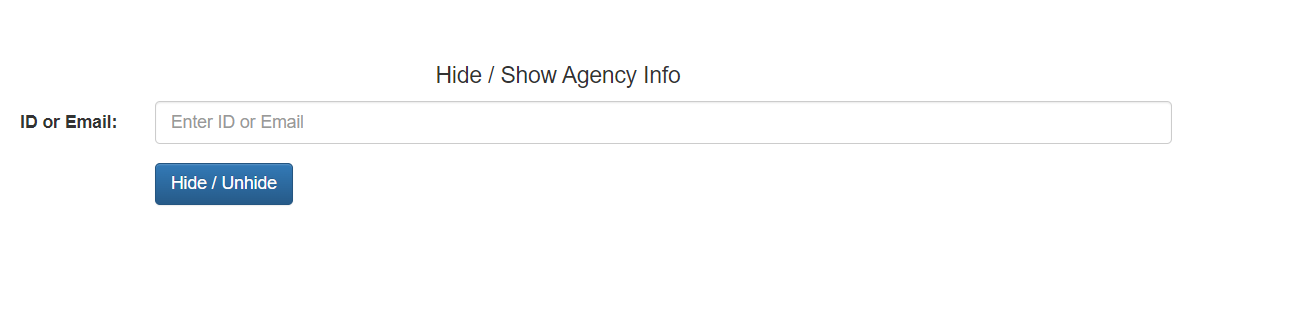


Image 20 - Sakrivanje agencije

### KORIŠĆENA LITERATURA

W1. <https://www.w3schools.com/>

W2. <https://phptherightway.com/>

W3. <https://leafletjs.com/>

W4. <https://www.openstreetmap.org/>

W5. <https://www.javatpoint.com/>

W6. <https://people.vts.su.ac.rs/~chole/nwp/examples/>