**VISOKA TEHNIČKA ŠKOLA**

**STRUKOVNIH STUDIJA**

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

**PROJEKAT**

**iz predmeta Web programiranje**

***IT BERZA***

**KANDIDAT MENTOR**

**Veljko Bogdan dr Zlatko Čović**

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

**SUBOTICA 2023.god.**

**SADRŽAJ**

[OPIS ZADATKA 3](#_Toc11643)

[REALIZACIJA ZADATKA 4](#_Toc13128)

[OPIS FUNKCIONALNOSTI 25](#_Toc18681)

**OPIS ZADATKA**

Cilj projekta je da se kreira web aplikacija-sajt za berzu IT poslova. Aplikacija razlikuje četiri nivoa pristupa: gost, registrovan/prijavljen korisnik, registrovan/prijavljen korisnik-kompanija i administrator.

Registracija korisnika se trebala uraditi 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.

Prilikom registracije je potrebno da korisnik izabere da li se registruje u ime kompanije ili kao običan korisnik. Ukoliko se registruje u ime kompanije, potebno je da unese ime kompanije kao i dodatne informacije o kompaniji-web sajt, adresa i opis kompanije).

Prijavljeni korisnik kompanije može da postavi oglas za posao. Da bi oglas bio aktivan na sajtu, administrator sistema treba da ga odobri. Nakon što je prijava za oglas istekla, vizuelno promeniti prikaz tog oglasa.

Na sajtu je potrebno omogućiti prikaz svih oglasa, oglasa koji su aktivni i isteklih oglasa.

Administratorski deo zaštiti upotrebom sesija (PHP).

Sve korisničke lozinke „hešovati“ bcrypt algoritmom.

Kreirati bazu podataka pod imenom job i unutar nje tabele koje će zadovoljiti sve funkcionalnosti projekta.

**REALIZACIJA ZADATKA**

**Prvo što sam uradio je povezao tabelu korišćenjem PDO-a:**

<?php

// Establish a connection $conn

try {

$conn = new PDO('mysql: host=localhost; dbname=job', 'root', '');

$conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

} catch (PDOException $e) {

echo $e->getMessage();

die();

}

**Onda sam uradio ban-check, koji proverava da li je trenutno prijavljen user banovan sa sajta, tako sto mu proverava is\_banned vrednost iz baze.**

<?php  
require\_once 'db-config.php'; // If a site doesn't have require db-config  
  
if (isset($\_SESSION['email'])) {  
 $email = $\_SESSION['email'];  
  
 // Prepare and execute the query  
 $stmt = $conn->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  
 }  
 }  
}

**Logout kod, briše sesije i vraća usera na login stranicu:**

<?php  
session\_start();  
session\_unset();  
session\_destroy();  
header("location:login.php");

**Ovaj kod se koristi da bi proverio da li je user ulogovan na sajt pre nego što pokuša da uđe na stranicu koja je dostupna samo ulogovanim korisnicima:**

<?php

if(!isset($\_SESSION['logged\_in']) && $\_SESSION['logged\_in'] !== true){

echo '<script>

alert("Access denied, login first");

window.location.href="login.php";

</script>';

}

**Ovaj kod se koristi da bi se video da li je trenutni user admin, ako nije šalje se na glavnu stranicu:**

if(!isset($\_SESSION['is\_admin']) || $\_SESSION['is\_admin'] !== true) {

header("Location: index.php");

}

**Ovaj PHP kod se koristi za prikaz poslovnih oglasa sa filtriranjem i straničenjem. Oglasi se dobavljaju iz baze podataka na osnovu odabranih filtera i prikazuju se u karticama. Takođe, prikazuju se linkovi za straničenje ako ima više od jedne stranice. Kod omogućava fleksibilno pretraživanje i pregled oglasa:**

<?php

// Define the number of boxes per page

$boxesPerPage = 5;

// Get the current page number

if (isset($\_GET['page'])) {

$currentPage = intval($\_GET['page']);

} else {

$currentPage = 1;

}

// Calculate the offset for the SQL query

$offset = ($currentPage - 1) \* $boxesPerPage;

$category = $\_GET['categories'] ?? '';

$location = $\_GET['locations'] ?? '';

$qualification = $\_GET['professional\_qualification'] ?? '';

$employmentType = $\_GET['employment\_type'] ?? '';

$empty = true;

if ($category !== '' || $location !== '' || $qualification !== '' || $employmentType !== '') {

$empty = false;

}

// Build the conditions based on the selected filter values

$conditions = [];

$params = [];

if (!empty($category)) {

$conditions[] = 'j.category = '.$category;

$params['category'] = $category;

}

if (!empty($location)) {

$conditions[] = 'j.city = '.$location;

$params['location'] = $location;

}

if (!empty($qualification)) {

$conditions[] = 'j.qualifications = '.$qualification;

$params['qualification'] = $qualification;

}

if (!empty($employmentType)) {

$conditions[] = 'j.employment\_type = '.$employmentType;

$params['employmentType'] = $employmentType;

}

// Add the conditions to the SQL statement if any exist

if (!empty($conditions)) {

$conditionsClause = ' WHERE j.period\_to > CURDATE() AND ' . implode(' AND ', $conditions);

} else {

$conditionsClause = ' WHERE j.period\_to > CURDATE() ';

}

// Query to fetch the job listings with pagination

$sql = "SQL SKRAĆEN";

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

// Fetch all rows to calculate the total number of pages

$totalResult = $conn->query("SELECT COUNT(\*) AS total FROM jobs");

$totalRows = $totalResult->fetch(PDO::FETCH\_ASSOC)['total'];

$totalPages = ceil($totalRows / $boxesPerPage);

// Display the job listings as clickable boxes

while ($row = $result->fetch(PDO::FETCH\_ASSOC)) {

// Convert from yyyy/mm/dd to yyyy - month name - dd

$dateFrom = date('Y-F-d', strtotime($row['period\_from']));

$dateTo = date('Y F d', strtotime($row['period\_to']));

echo '<a href="job-details.php?id=' . $row['id\_job'] . '&idCompany='.$row['id\_company'].'" class="link-disabled">';

echo '<br>';

echo '<div class="card card-rounded">';

echo '<div class="card-header card-rounded ';

if (strtotime($row['period\_to']) < strtotime(date('Y-m-d'))) {

echo 'card-header-expired';

} // echo expired class if the time is expired

echo '">';

echo '<h3 class="card-title lead">' . $row['position\_name'] . '</h3><h4 class="white"> ' . $row['company\_name'] . '</h4>';

echo '</div>';

echo '<div class="card-body">';

echo '<p class="card-text"><span class="glyphicon glyphicon-map-marker"></span> ' . strtoupper($row['city']) . '</p>';

echo '<p class="card-text ';

if (strtotime($row['period\_to']) < strtotime(date('Y-m-d'))) {

echo 'red';

} // echo red class if the time is expired

echo '"><span class="glyphicon glyphicon-time"></span> ' . $dateTo . '</p>';

echo '</div>';

echo '<div class="card-footer">';

if(isset($\_SESSION['is\_admin']) && $\_SESSION['is\_admin']===true && $row['is\_enabled']==0){

echo '<div class="text-center">

<a class="btn btn-success border" href="validate-job.php?id='.$row['id\_job'].' "><x button><h4>VALIDATE AD</h4></xbutton></a>

</div><br>';

}

echo '<div></div>';

echo '</div>';

echo '</div>';

echo '</a>';

}

// Display pagination links

if ($totalPages > 1) {

echo '<nav class="text-center">';

echo '<ul class="pagination justify-content-center">';

// Previous page link

if ($currentPage > 1) {

echo '<li class="page-item"><a class="page-link" href="?page=' . ($currentPage - 1) . '">Previous</a></li>';

}

// Page links

for ($i = 1; $i <= $totalPages; $i++) {

echo '<li class="page-item';

if ($i === $currentPage) {

echo ' active';

}

echo '"><a class="page-link" href="?page=' . $i . '">' . $i . '</a></li>';

}

// Next page link

if ($currentPage < $totalPages) {

echo '<li class="page-item"><a class="page-link" href="?page=' . ($currentPage + 1) . '">Next</a></li>';

}

echo '</ul>';

echo '</nav>';

}

?>

**Ovaj kod je ajax zahtjev zajedno sa php kodom koji pritiskom na dugme Expiring Soon ispisuje 5 poslova čiji se period prijavljivanja brzo zavrsava:**

<?php

include\_once 'db-config.php';

$s = "SELECT id\_job, position\_name, company\_name

FROM jobs

WHERE period\_to >= CURDATE()

ORDER BY period\_to ASC

LIMIT 5";

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

$response = array();

if ($result->rowCount() > 0) {

while ($row = $result->fetch(PDO::FETCH\_ASSOC)) {

$response[] = $row;

}

}

header('Content-Type: application/json');

echo json\_encode($response);

?>

var isCardBoxVisible = false;

function loadCardBoxes() {

$.ajax({

url: 'load-card-boxes.php',

type: 'GET',

dataType: 'json',

success: function(response) {

if (response && response.length > 0) {

if (isCardBoxVisible) {

$('.card-container').empty();

isCardBoxVisible = false;

} else {

$.each(response, function(index, item) {

var positionName = item.position\_name;

var companyName = item.company\_name;

var idJob = item.id\_job;

var cardBox = '<a class="link-disabled" href="job-details.php?id=' + idJob + '" class="card-link">' +

'<div class="card card-side card-rounded">' +

'<div class="card-body card-side-body">' +

'<h4 class="card-title">' + positionName + '</h4>' +

'<h5 class="card-subtitle mb-2 text-muted">' + companyName + '</h5>' +

'</div>' +

'</div>' +

'</a>';

$('.card-container').append(cardBox);

});

isCardBoxVisible = true;

}

}

},

error: function() {

console.log('Error occurred during AJAX request.');

}

});

}

$('.load-card-boxes-link').on('click', function(e) {

e.preventDefault();

loadCardBoxes();

});

$('#toggleFormBtn').click(function() {

$('#searchForm').slideToggle();

});

**Ovo je html kod koji sadrži kostur koji će služiti za čet:**

<div id="chatbox" class="content container">

<div id="message-container"></div>

<input type="text" id="message-input" placeholder="Type your message">

<button id="send-btn" class="btn btn-success">Send</button>

</div>

**Ovo je ajax i php kod koji koriste taj kostur kako bi ispisali poruke u četu:**

**Kada je ajax kod pozvan, on dobija trenutnog usera i usera sa kojim želi da četuje, enterom ili klikom na dugme send se poruke šalju. Kada se dobije nova poruka čet skroluje dole. Čet radi update svake 3 sekunde:**

var urlParams = new URLSearchParams(window.location.search);

var currentUser = urlParams.get('currentUser');

var otherUser = urlParams.get('otherUser');

// Load the initial messages

loadMessages();

// Load messages when the page is loaded and every 3 seconds

setInterval(function() {

loadMessages();

}, 3000);

// Handle sending a new message

$('#send-btn').click(function() {

var message = $('#message-input').val();

if (message.trim() !== '') {

message = "<small>" + currentUserEmail + "</small>: " + message;

sendMessage(message);

$('#message-input').val('');

}

});

// Handle pressing Enter key to send a message

$('#message-input').keypress(function(event) {

if (event.which === 13) {

var message = $('#message-input').val();

if (message.trim() !== '') {

message = "<small>" + currentUserEmail + "</small>: " + message;

sendMessage(message);

$('#message-input').val('');

}

return false;

}

});

// Load messages from the server

function loadMessages() {

$.ajax({

url: 'load\_messages.php',

type: 'POST',

data: {

currentUser: currentUser,

otherUser: otherUser

},

success: function(response) {

$('#message-container').html(response);

scrollChatToBottom();

},

error: function(xhr, status, error) {

console.error(xhr.responseText);

}

});

}

// Send a new message to the server

function sendMessage(message) {

$.ajax({

url: 'send\_message.php',

type: 'POST',

data: {

currentUser: currentUser,

otherUser: otherUser,

message: message

},

success: function(response) {

loadMessages(); // Reload messages after sending

},

error: function(xhr, status, error) {

console.error(xhr.responseText);

}

});

}

// Scroll the chat container to the bottom

function scrollChatToBottom() {

var container = document.getElementById('message-container');

container.scrollTop = container.scrollHeight;

}

**PHP kod koji uzima podatke iz tabele messagess:**

<?php

require\_once 'db-config.php';

// Get the current user and other user IDs from the AJAX request

$currentUser = $\_POST['currentUser'];

$otherUser = $\_POST['otherUser'];

// Fetch messages from the database

$sql = "SELECT \* FROM messages WHERE (sender\_id = $currentUser AND receiver\_id = $otherUser) OR (sender\_id = $otherUser AND receiver\_id = $currentUser) ORDER BY timestamp ASC";

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

// Display the messages

if ($result->rowCount() > 0) {

while ($row = $result->fetch(PDO::FETCH\_ASSOC)) {

$message = $row['message'];

$senderID = $row['sender\_id'];

// Determine the CSS class for the message based on the sender

$messageClass = ($senderID == $currentUser) ? 'sent' : 'received';

// Display the message with the appropriate class

echo '<div class="' . $messageClass . '">' . $message . '</div>';

}

} else {

echo '<div>No messages yet.</div>';

}

?>

**Confirmation.php sadrži više funkcija:**

**sendMail - koji šalje mejl i verifikacioni kod na mail korisnika koji se želi registrovati  
passwordChange - koji šalje mejl i verifikacioni kod na mail korisnika koji želi promenu lozinke jer ju je zaboravio.**

**Ovaj kod se koristi za prikaz pojedinosti o poslovnom oglasu na web stranici.**

Koristi se HTML kod za prikazivanje kartice (<div class="card card-rounded">) koja će sadržati informacije o oglasu.

U card-header se prikazuje naziv pozicije i ime kompanije, čiji podaci se dobavljaju iz baze podataka.

U card-body se prikazuju detalji oglasa, kao što su grad, rok za prijavu, vrsta zaposlenja, kvalifikacija i informacija da li je moguće raditi od kuće.

Nakon toga, u content card se prikazuje tekst oglasa.

U card-footer se prikazuju dodatne opcije, uključujući kontaktiranje kompanije ako je korisnik prijavljen, kao i mogućnost omogućavanja ili onemogućavanja oglasa, koja je dostupna samo administratorima.

<div class="card card-rounded">  
 <div class="card-header card-rounded">  
 <h3 class="text-left">  
 <?php  
 $id = $\_GET['id'];  
 if(isset($\_GET['idCompany'])){  
 $idCompany = $\_GET['idCompany'];  
 }  
 else{  
 $idCompany=0;  
 }  
 $dateTo = "";  
  
 include\_once 'job-details-sql.php';  
  
 if ($result) {  
 $row = $result->fetch(PDO::FETCH\_ASSOC);  
 echo $row['position\_name'];  
 $dateTo = date('Y F d', strtotime($row['period\_to']));  
 }  
 ?>  
 </h3>  
 <div>  
 <h4 class="white"> <?php echo $row['company\_name']; ?> </h4>  
 </div>  
 </div>  
 <div class="card-body card-rounded">  
 <hr>  
 <h3 class="text-center">Job Information</h3>  
 <div>  
 <?php echo '<span class="glyphicon glyphicon-map-marker"></span> '. strtoupper($row['city']) ; ?>  
 </div>  
 <div>  
 <?php echo 'Registration Deadline: <b><span class="glyphicon glyphicon-time"></span> '.$dateTo.'</b><br><br>'; ?>  
 </div>  
 <div>  
 <?php echo '<span class="glyphicon glyphicon-map-marker"></span> '. $row['employment\_type\_name'] ; ?>  
 </div>  
 <div>  
 <?php echo '<span class="glyphicon glyphicon-map-marker"></span> '. $row['qualification\_name'] ; ?>  
 </div>  
 <div>  
 <?php echo $row['remote']? "REMOTE": "NOT REMOTE" ; ?>  
 </div>  
 <div class="content card">  
 <hr>  
 <div class="card-body"><?php echo $row['text'] ; ?></div>  
 <hr>  
 </div>  
 </div>  
 <div class="card-footer card-rounded">  
 <div class="text-center">  
 <?php  
 if(isset($\_SESSION['logged\_in']) && $\_SESSION['logged\_in']===true){  
  
 if (!isset($\_SESSION['is\_company'])) {  
 echo '<div class="text-center">  
 <a href="chat.php?currentUser=' . $\_SESSION['id'] . '&otherUser=' . $idCompany . '">Contact Company</a>  
 </div><br>';  
 }  
  
 if(isset($\_SESSION['is\_admin']) && $\_SESSION['is\_admin']===true){  
 $sql = "SELECT is\_enabled  
 FROM jobs  
 WHERE id\_job = :id";  
 $stmt = $conn->prepare($sql);  
 $stmt->bindParam(':id', $id);  
 $stmt->execute();  
  
 if ($stmt->rowCount() > 0) {  
 $row = $stmt->fetch(PDO::FETCH\_ASSOC);  
 $isEnabled = $row['is\_enabled'];  
  
 // Determine the cube color based on the value of is\_enabled  
 $cubeColor = ($isEnabled == 1) ? 'green' : 'red';  
  
 echo '<a class="cube link-disabled '.$cubeColor. '" href="validate-job.php?id='.$id.'&isEnabled='.$isEnabled.'"><button><h4>ENABLE / DISABLE</h4></button></a><br><small>Green is enabled, red is disabled</small>';  
 }  
 }  
 }  
 ?>  
 </div>  
 </div>  
 <br>  
</div>

**Ovaj kod predstavlja PHP klasu JobFormHandler koja se koristi za obradu podataka poslatih putem forme za unos poslovnog oglasa.**

**Klasa ima konstruktor koji prima PDO objekt za povezivanje s bazom podataka.**

**Metoda handleFormSubmission() se koristi za obradu podataka poslatih putem forme. Prvo se provjerava, validira i obrezuje svako polje podataka poslanih POST metodom.**

**Nakon provjere valjanosti svih polja, provjerava se jesu li sva polja popunjena. Ako bilo koje polje nije popunjeno, korisnik se preusmjerava natrag na formu s prikazm pogreške.**

**Ako su sva polja ispravno popunjena, izvršava se SQL upit za umetanje podataka u bazu podataka. Ako je umetanje uspješno, korisnik se preusmjerava na početnu stranicu. Ako se dogodi greška pri umetanju, korisnik se preusmjerava natrag na formu s prikazom pogreške.**

**Metoda validateAndTrimData() se koristi za validaciju i obrezivanje podataka. Podaci se obrezuju kako bi se uklonili vodeći i slijedeći razmaci, a zatim se provjerava je li podatak prazan. Ako je podatak prazan, vraća se false, inače se vraća validirani i obrezani podatak.**

<?php  
  
class JobFormHandler  
{  
 private $conn;  
  
 public function \_\_construct($conn)  
 {  
 $this->conn = $conn;  
 }  
  
// Function to validate and trim the data  
  
 public function handleFormSubmission()  
 {  
// Validate and trim each field  
 $contactPerson = $this->validateAndTrimData($\_POST['contact\_name']) ?? null;  
 $contactEmail = $this->validateAndTrimData($\_POST['contact\_email']) ?? null;  
 $telephone = $this->validateAndTrimData($\_POST['contact\_phone']) ?? null;  
 $companyName = $this->validateAndTrimData($\_POST['company\_name']) ?? null;  
 $tin = $this->validateAndTrimData($\_POST['pib']) ?? null;  
 $position = $this->validateAndTrimData($\_POST['position\_name']) ?? null;  
 $category = $this->validateAndTrimData($\_POST['categories']) ?? null;  
 $city = $this->validateAndTrimData($\_POST['locations']) ?? null;  
 $remote = $this->validateAndTrimData($\_POST['remote\_work']) ?? null;  
 $qualifications = $this->validateAndTrimData($\_POST['professional\_qualification']) ?? null;  
 $employmentType = $this->validateAndTrimData($\_POST['employment\_type']) ?? null;  
 $text = $this->validateAndTrimData($\_POST['text']) ?? null;  
 $signupEmail = $this->validateAndTrimData($\_POST['application\_address']) ?? null;  
 $signupPhone = $this->validateAndTrimData($\_POST['application\_phone']) ?? null;  
 $duration = $this->validateAndTrimData($\_POST['ad\_period']) ?? null;  
 $signupPeriodFrom = $this->validateAndTrimData($\_POST['visible\_from']) ?? null;  
 $signupPeriodTo = $this->validateAndTrimData($\_POST['visible\_to']) ?? null;  
  
// Check if any field is null, indicating validation failure  
 if (  
 $contactPerson === null ||  
 $contactEmail === null ||  
 $telephone === null ||  
 $companyName === null ||  
 $tin === null ||  
 $position === null ||  
 $category === null ||  
 $city === null ||  
 $remote === null ||  
 $qualifications === null ||  
 $employmentType === null ||  
 $text === null ||  
 $signupEmail === null ||  
 $signupPhone === null ||  
 $duration === null ||  
 $signupPeriodFrom === null ||  
 $signupPeriodTo === null  
 ) {  
// Handle validation error, e.g., redirect back with an error message  
 header('Location: job-form.php?error=1');  
 exit;  
 }  
  
 $query = "INSERT INTO `jobs`(`contact\_person`, `contact\_email`, `contact\_telephone`, `company\_name`, `poster\_company\_name`,  
`tax\_id\_number`,`position\_name`, `category`, `city`,`remote`, `qualifications`, `employment\_type`,  
`text`, `signup\_email`, `signup\_telephone`, `duration`, `period\_from`, `period\_to`, `id\_company`)  
VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)";  
 $stmt = $this->conn->prepare($query);  
 $insertConfirmation = $stmt->execute([$contactPerson, $contactEmail, $telephone, $companyName, $companyName, $tin, $position,  
 $category, $city, $remote, $qualifications, $employmentType, $text, $signupEmail, $signupPhone, $duration, $signupPeriodFrom, $signupPeriodTo, $\_SESSION['id']]);  
  
 if ($insertConfirmation === true) {  
 header('Location: index.php');  
 exit;  
 } else {  
 header('Location: job-form.php?error=2');  
 exit;  
 }  
 }  
  
 private function validateAndTrimData($data)  
 {  
 $data = trim($data); // Remove leading/trailing whitespaces  
  
 if (empty($data)) {  
 return false; // Empty data  
 }  
  
 return $data; // Validated and trimmed data  
 }  
}

**Ovaj kod je dio skripte koja se koristi za prikazivanje podataka korisnika na web stranici te omogućuje ažuriranje određenih informacija.**

**Na početku, skripta dobija e-poštu korisnika iz sesije i izvršava upit prema bazi podataka kako bi dobila odgovarajuće podatke za tu e-poštu. Ako je rezultat upita uspješan i vraća barem jedan red, tada se prikazuju informacije.**

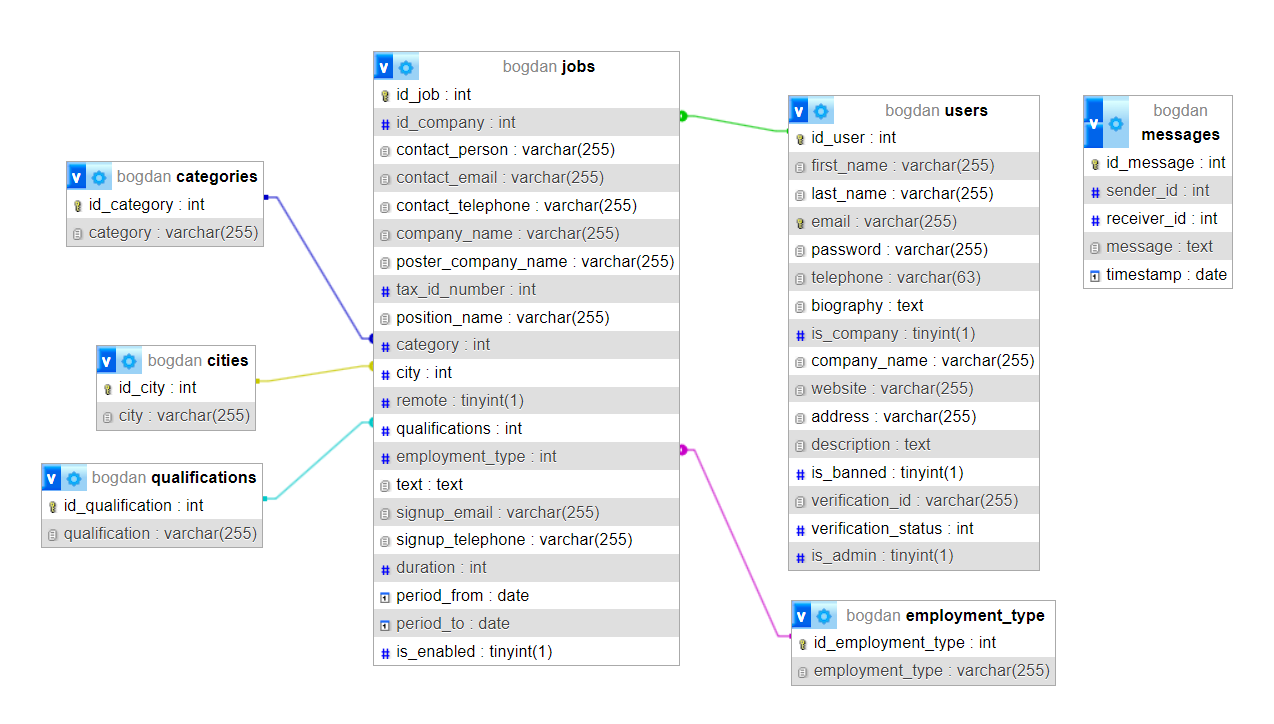
**Ako je korisnik ili tvrtka, prikazuju se različite informacije. Za korisnika se prikazuje tip računa (korisnik ili kompaniju), e-pošta, te se omogućuje ažuriranje imena, prezimena i telefonskog broja putem obrasca. Također se prikazuje biografija korisnika s mogućnošću ažuriranja.**

**Za kompaniju se prikazuju informacije kao što su naziv, web stranica i adresa. Također se prikazuje opis kompanijes mogućnošću ažuriranja.**

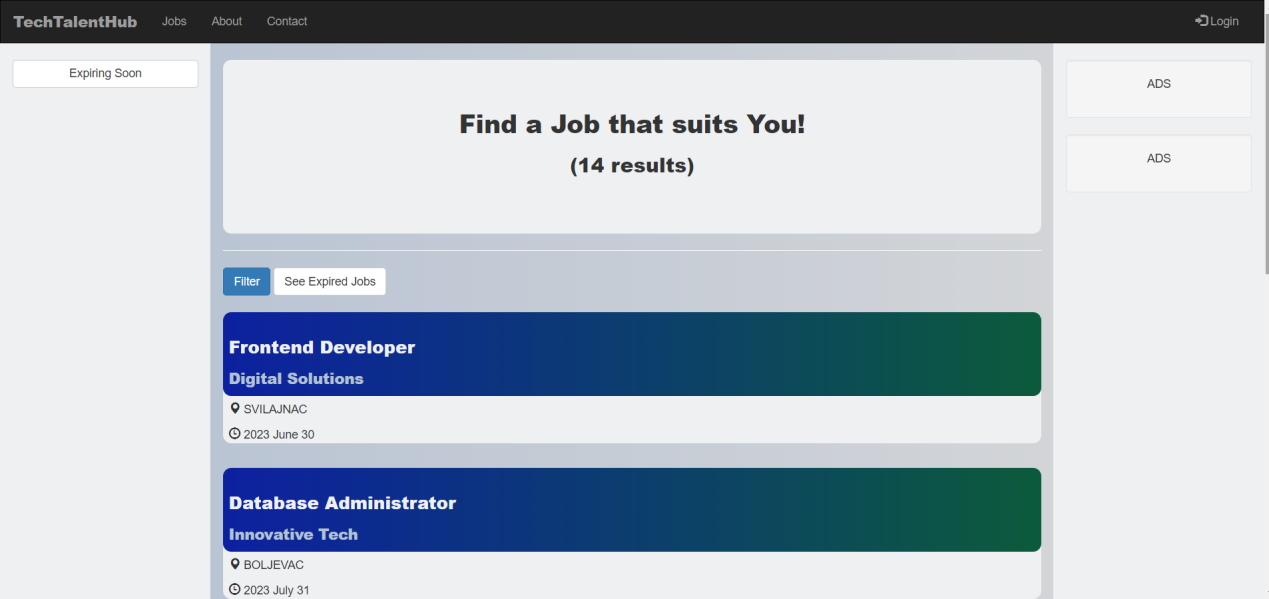
**Ako upit ne vrati nijedan red ili se dogodi pogreška, korisnik se preusmjerava na stranicu za prijavu s prikazom upozorenja.**

<?php  
require\_once 'db-config.php';  
  
$email = $\_SESSION['email'];  
$sql = "SELECT \* FROM users WHERE email='$email'";  
$result = $conn->query($sql);  
  
// Check if any rows were returned  
if ($result->rowCount() > 0) {  
 // Loop through each row  
 while ($row = $result->fetch(PDO::FETCH\_ASSOC)) {  
 // Display the account type based on the is\_company field  
 $accountType = $row['is\_company'] == 1 ? "Company Account" : "User Account";  
 $bio = $row['biography'];  
  
 // Output the data  
 echo '<div class="lead"><b>' . $accountType . "</b></div>";  
 echo '<div class="email">Email: ' . $row['email'] . "</div>";  
 echo '<form class="form-horizontal" action="update-bio.php" method="post" onsubmit="return validateUpdateInfo()">';  
 echo '<label class="name control-label" for="name">First Name: </label><input class="well form-control" type="text" id="name" name="name" value=' . $row['first\_name'] . '>';  
 echo '<div id="first-name-error" class="red"></div>';  
 echo '<label class="last-name control-label" for="last-name">Last Name: </label><input class="well form-control" type="text" id="last-name" name="last-name" value=' . $row['last\_name'] . '>';  
 echo '<div id="last-name-error" class="red"></div>';  
 echo '<label class="phone control-label" for="phone">Telephone: </label><input class="well form-control" type="text" id="phone" name="phone" value=' . $row['telephone'] . '>';  
 echo '<div id="phone-error" class="red"></div>';  
 echo '<button class="btn btn-default border" name="update-info" type="submit">Update Info</button><br>';  
 echo '</form>';  
 echo '<hr>';  
 if ($row['is\_company'] == 1) {  
 echo '<hr>';  
 echo '<div class="name">Company\_Name: ' . $row['company\_name'] . "</div>";  
 echo '<div class="website">Website: ' . $row['website'] . "</div>";  
 echo '<div class="address">Address: ' . $row['address'] . "</div>";  
  
 echo '<form class="form-horizontal" action="update-bio.php" method="post">';  
 echo '<label class="biography control-label" for="biography">Biography:</label><input class="well form-control" type="text" id="description" name="description" value=' . $row['description'] . '>';  
 echo '<button class="btn btn-default border" name="update-description" type="submit">Update Description</button><br>';  
 echo '</form>';  
 } else {  
 echo '<form class="form-horizontal" action="update-bio.php" method="post">';  
 echo '<label class="biography control-label" for="biography">Biography:</label><input class="well form-control" type="text" id="biography" name="biography" value="' . $bio . '">';  
 echo '<button class="btn btn-default border" name="update-biography" id="update-biography" type="submit">Update Bio</button><br>';  
 echo '</form>';  
 }  
 echo "<br>";  
 }  
} else {  
 echo "  
 <script>  
 alert('Error!');  
 window.location.href='login.php';  
 </script>";  
}  
?>

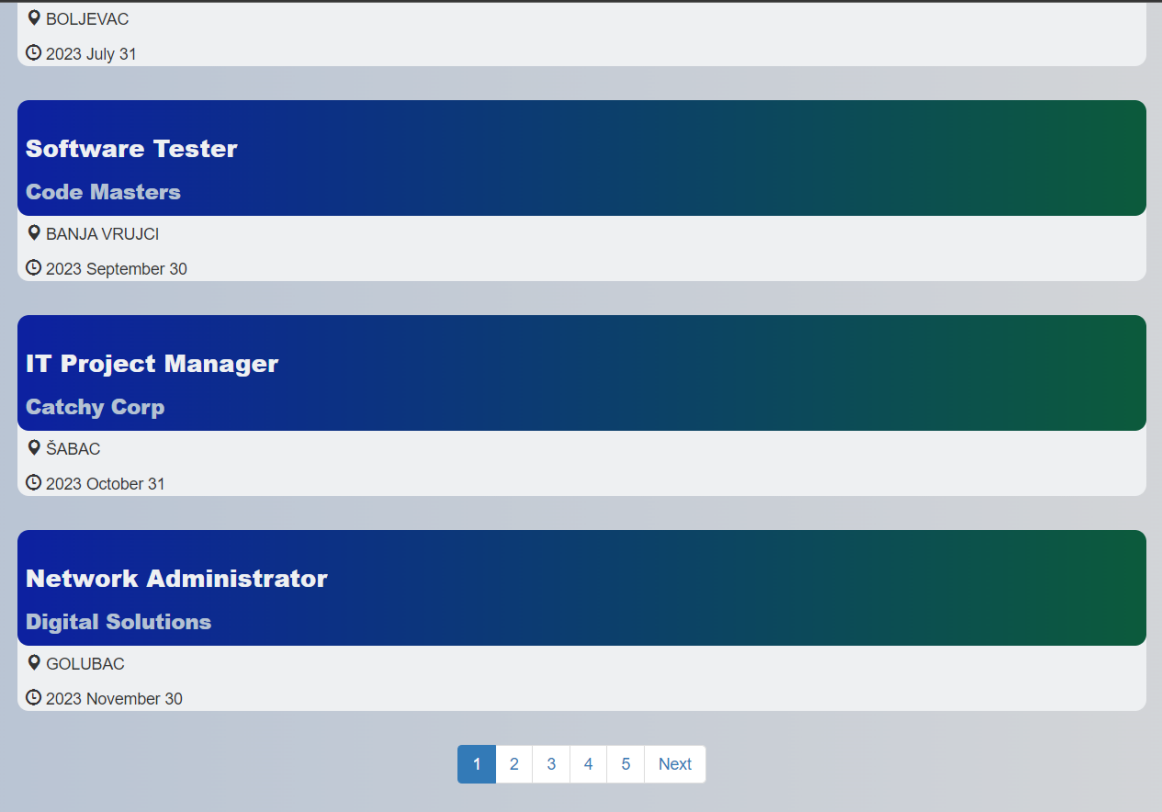
**STRUKTURA BAZE PODATAKA (MySQL Workbench)**

****

**OPIS FUNKCIONALNOSTI**



**Index.php stranica - gost nivo pristupa**



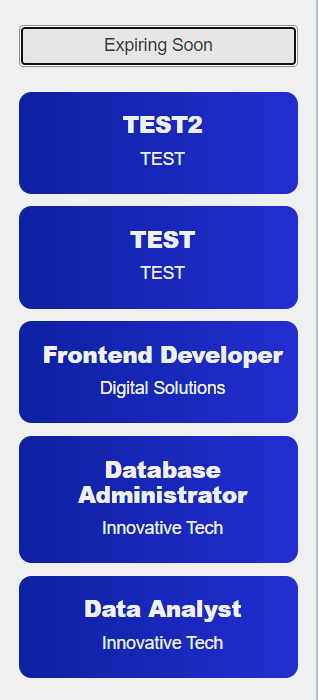
**Stranica izlistava 5 oglasa, sa stranicama koje prolaze kroz stranice pomoću LIMIT sql statementa.**



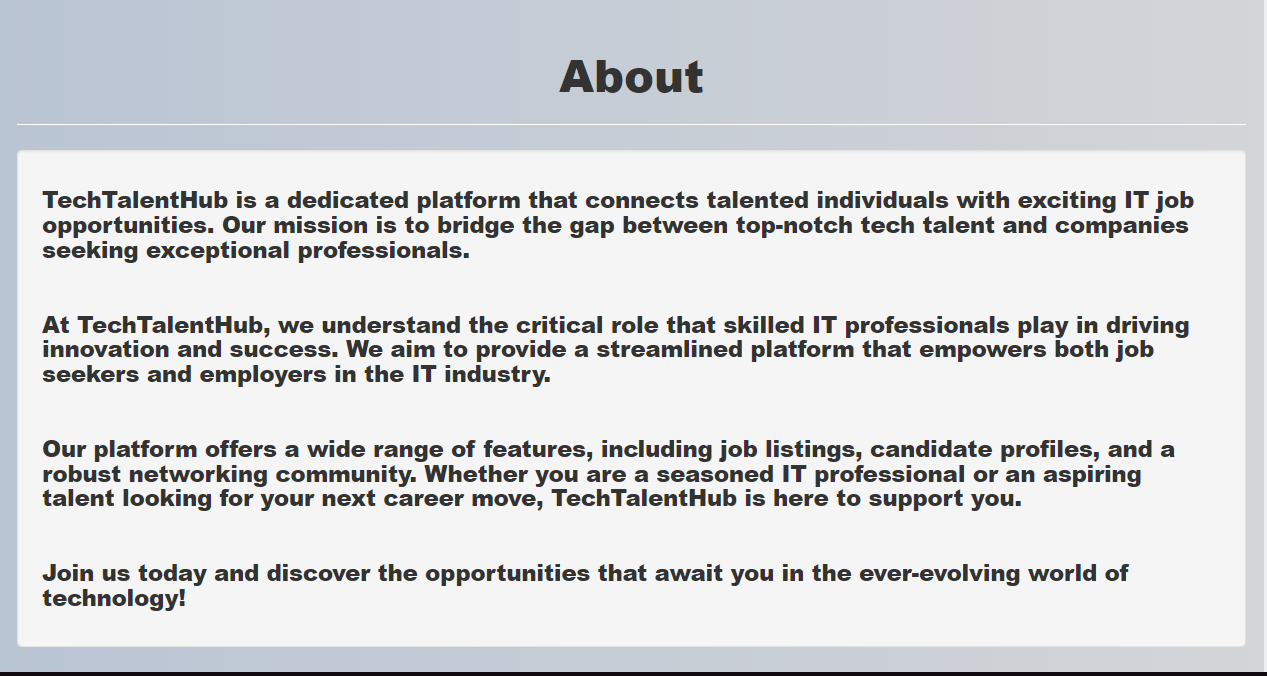
**Svi korisnici mogu da pretražuju preko FILTER dugmeta, koji slajduje prema dole, prikazivajući odabir filtera.**



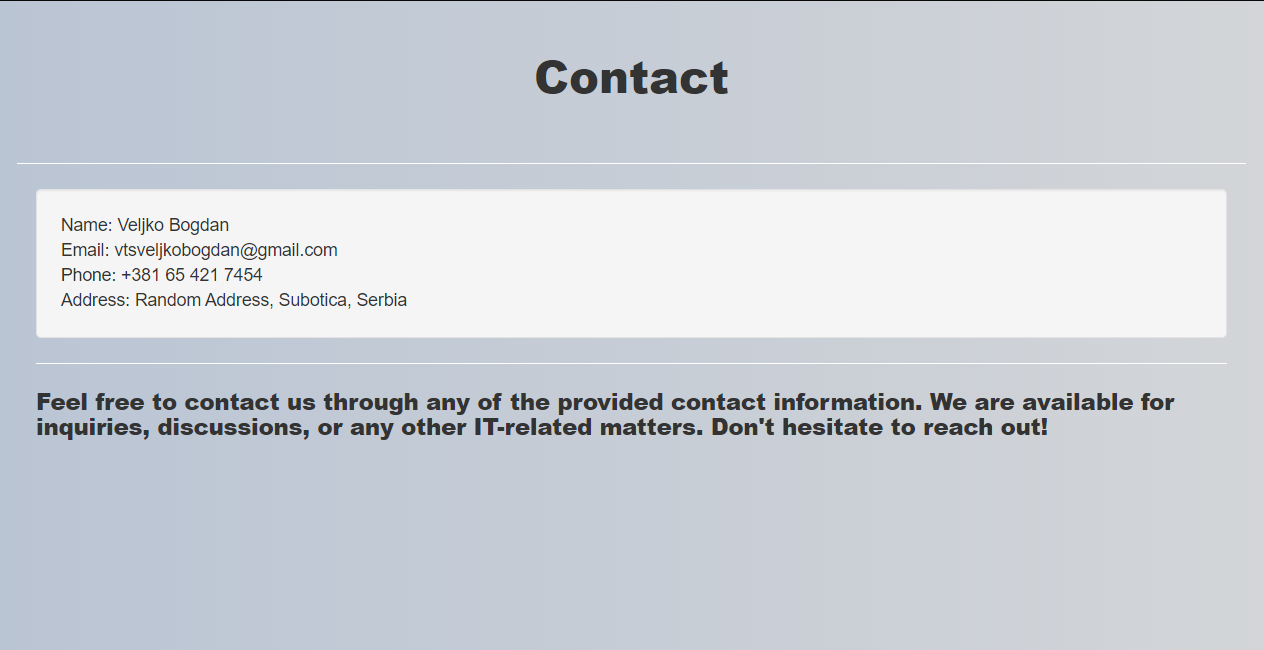
Klikom na dugme SEE EXPIRED JOBS idemo na stranicu list-archived koja izlistava sve jobove koji su istekli. Sajt gleda da li je današnji datum veći od roka prijave, ako jeste, promeni im boju.



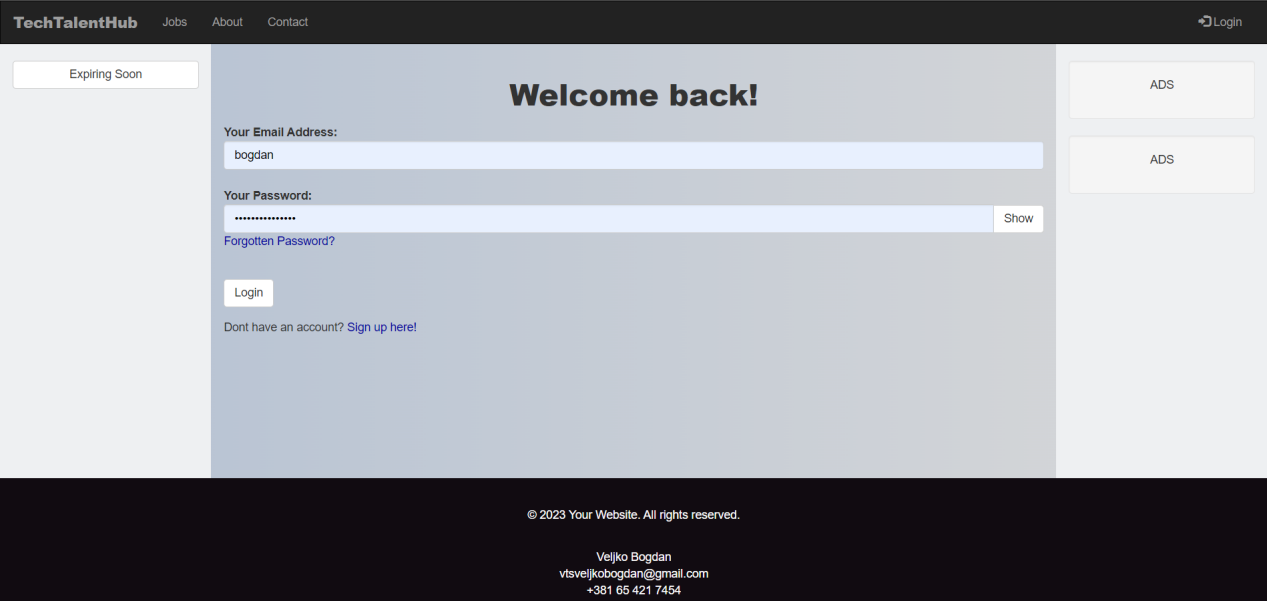
Klikom na Expiring Soon dugme, pojavljuju se 5 oglasa čiji je period prijave najbliži današnjem datumu. Na telefonima je prikaz manji i može se skrolovati.



About page

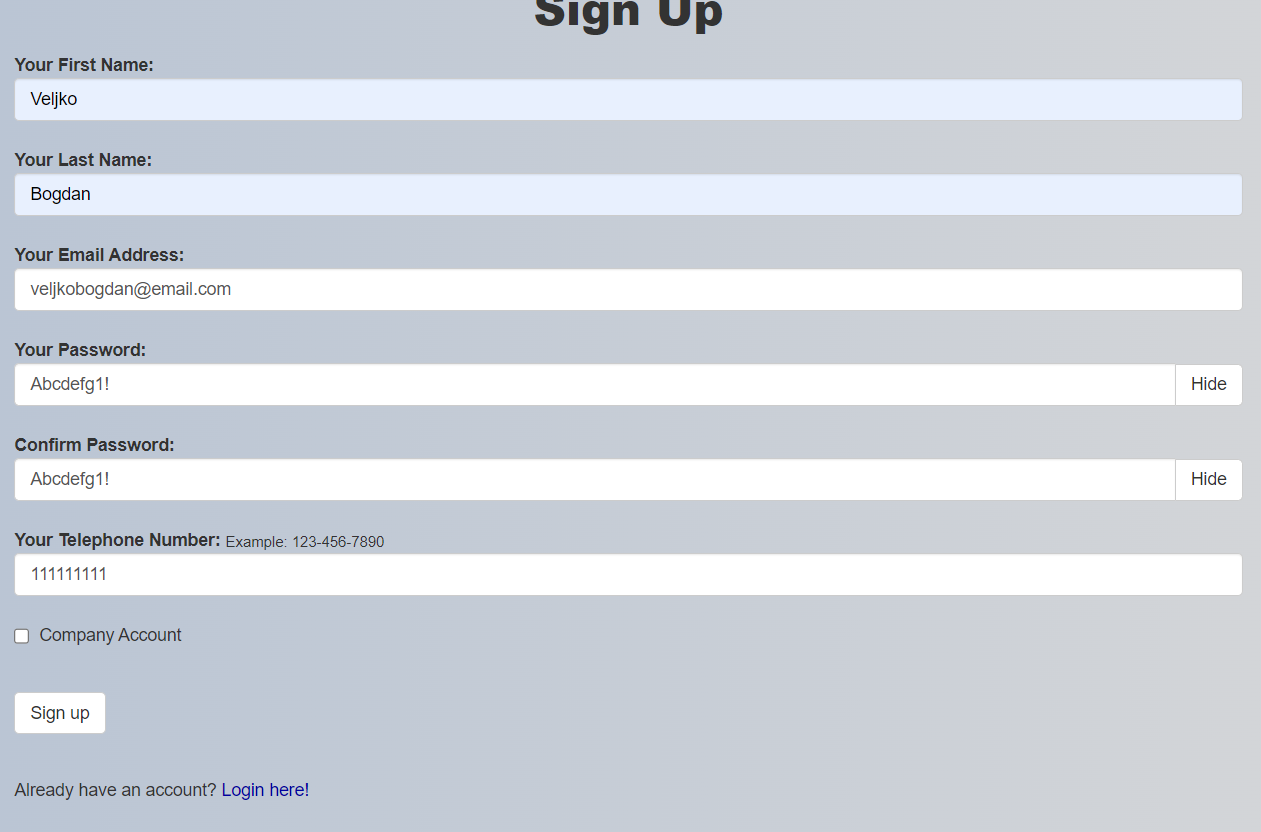


Contact page

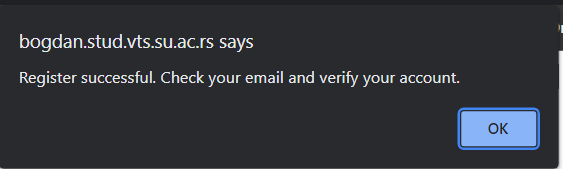
Login page, ako želimo da idemo na signup, kliknemo na link ispod forme.



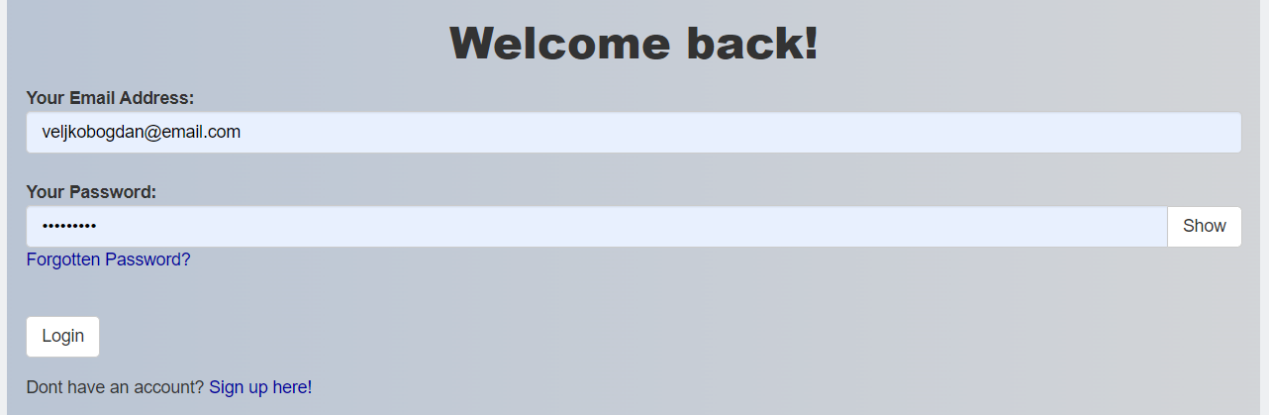
Ako pritisnemo checkbox Companz Account, prikazuje nam se dodatni form input-i. Skripta neće proveravati ove unose ako checkbox nije izabran, i staviće ih na NULL, a is\_company na 0.



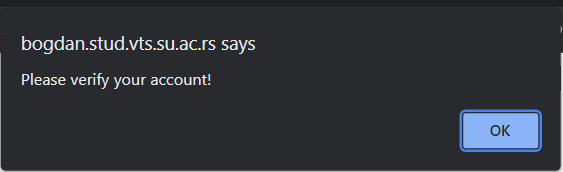
Ubacimo neke random podatke...



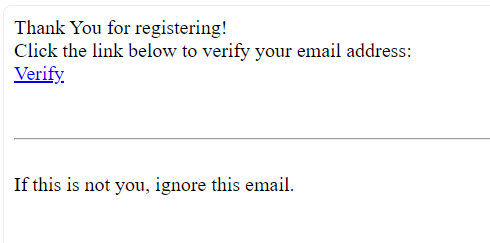
Izbacuje nam da trebamo da potvrdimo našu email adresu. Na email se poslao link sa GET promenjlivama email i v\_cod, koji je random heksadecimalni niz od 32 karaktera, koji je takođe unet u user tabelu u verification\_id.

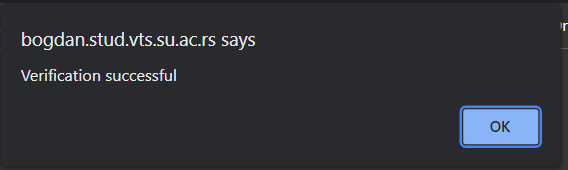


Ako se pokušam ulogovati u taj profil...

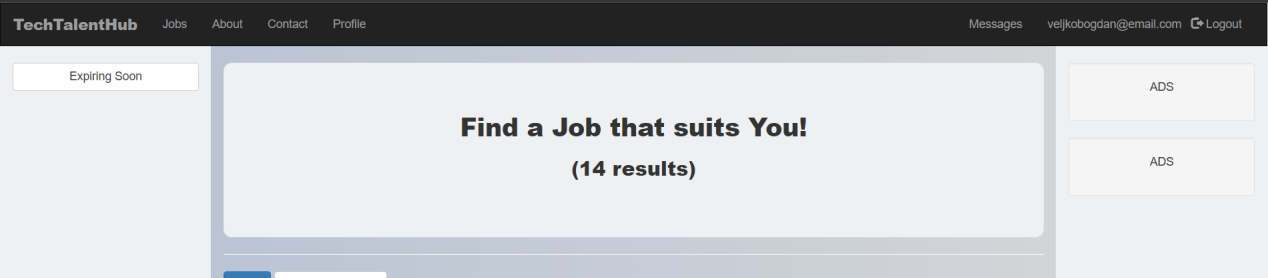


Reći će nam da trebam prvo da verifikujem profil, do tog zaključka dođe tako što pogleda da li je verification\_status jednak 1 tamo gdje je moj email.

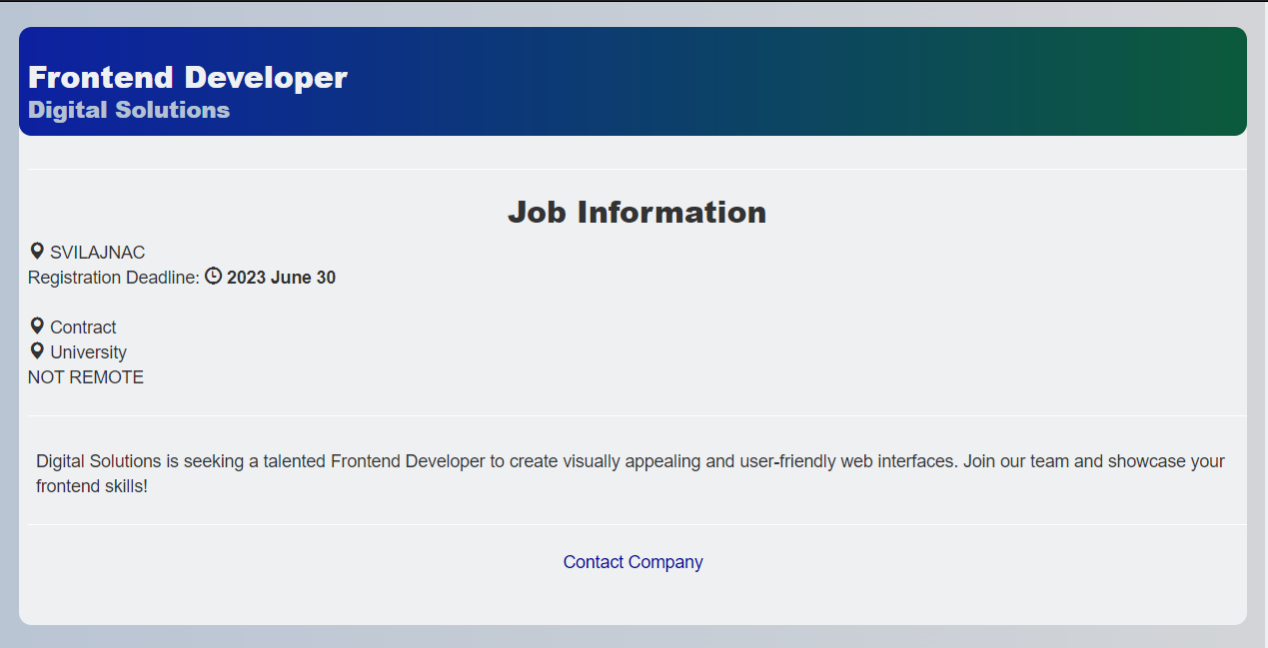




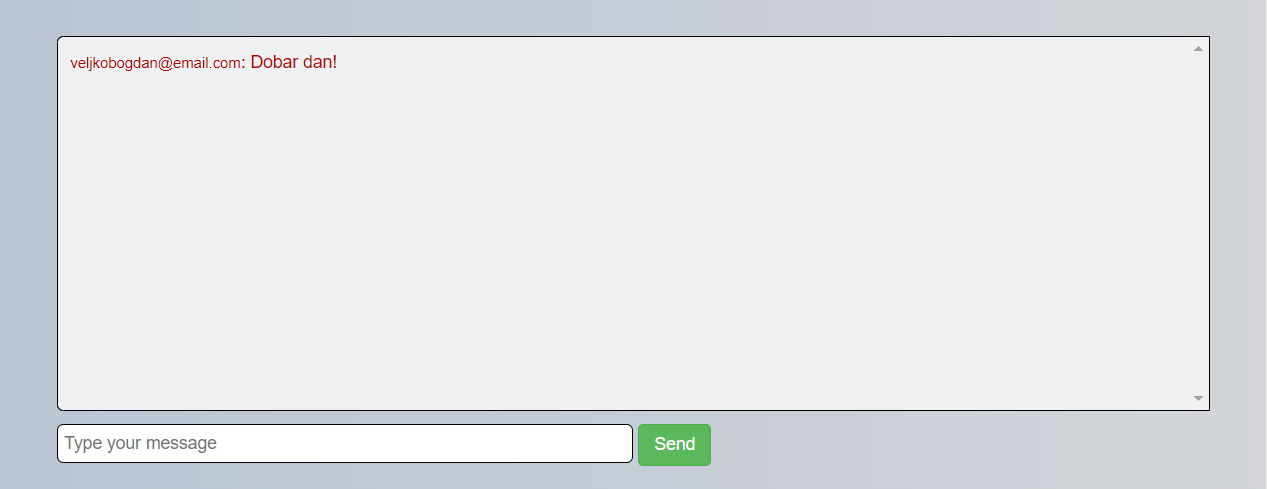
Klikom na VERIFY, sajt pogleda da li je verification\_status jednak 0, a verification\_id jednak v\_cod-u koji smo dobili, ako jeste, verification\_status se stavlja na 1, i sada se korisnik može ulogovati.



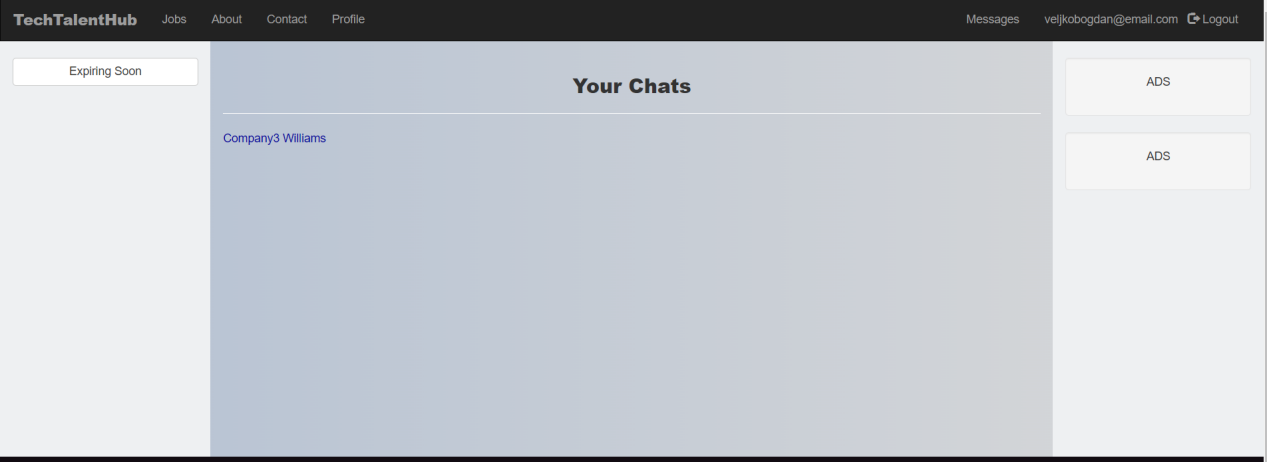
Ulogovani korisnik ima PROFILE i MESSAGES tab u headeru.



Klikom na karticu sada imamo i Contact Company link koji nas vodi na čet sa tom kompanijom.



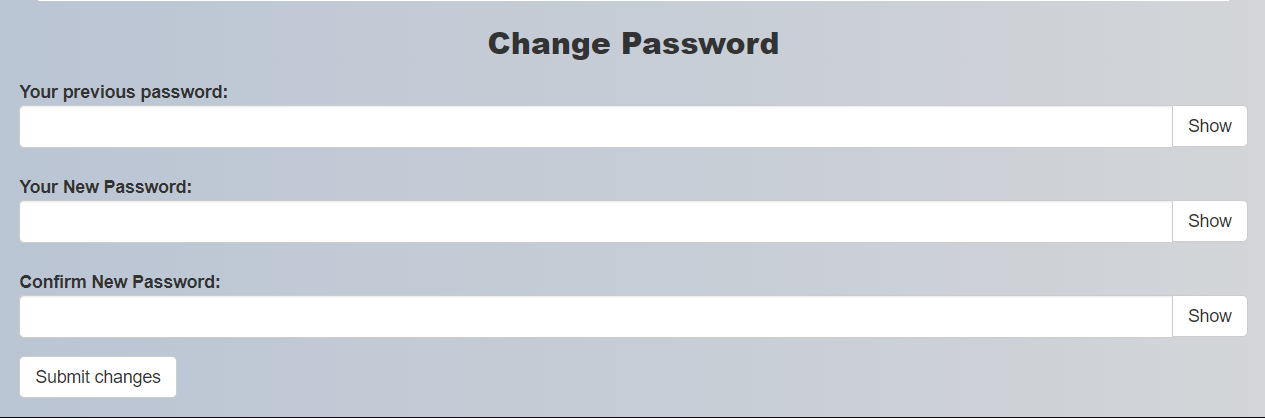
Čet, apdejtuje se svake 3 sekunde i poredan je po vremenu slanja.



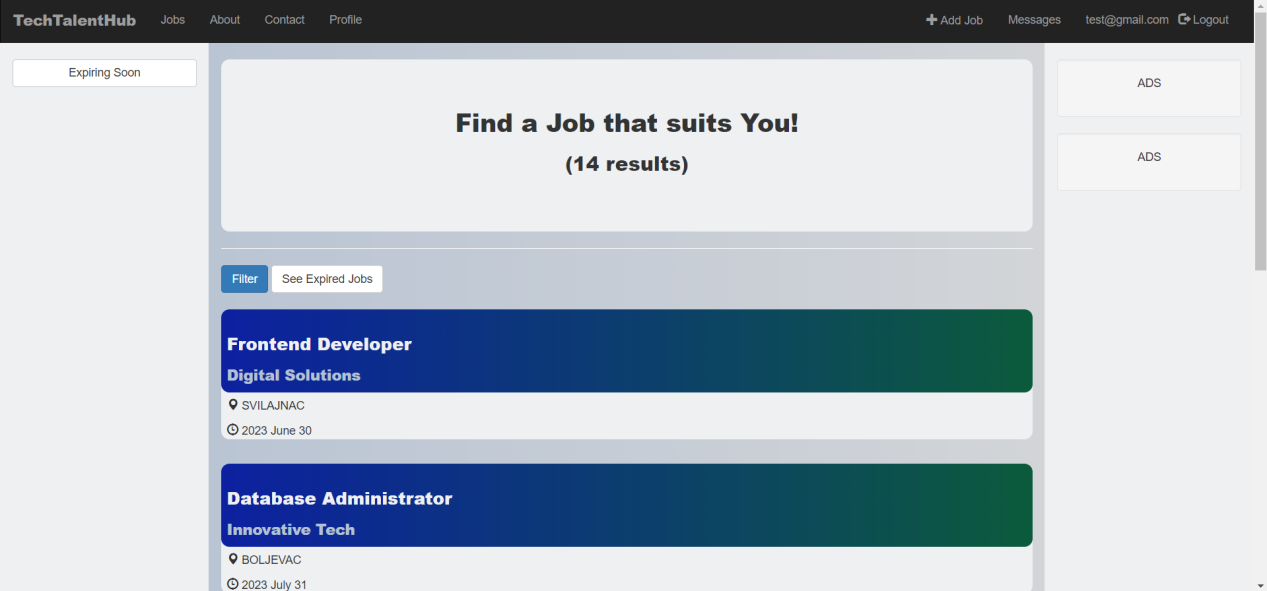
Sada je u Messages tabu otvoren link na čet sa tom kompanijom.



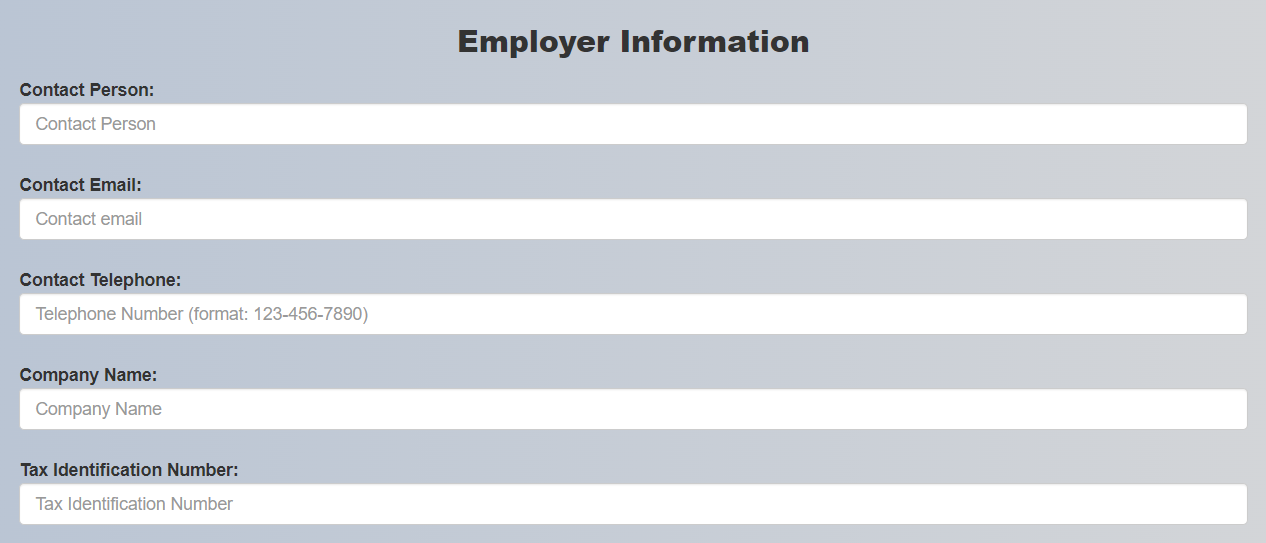
U profile tabu, možemo da menjamo svoje ime, prezime i telefon, kao i biografiju.



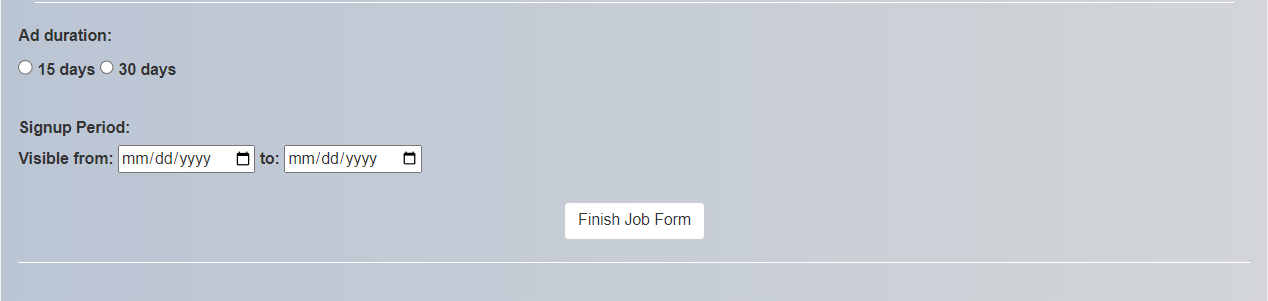
Možemo i da menjamo lozinku na istoj stranici.



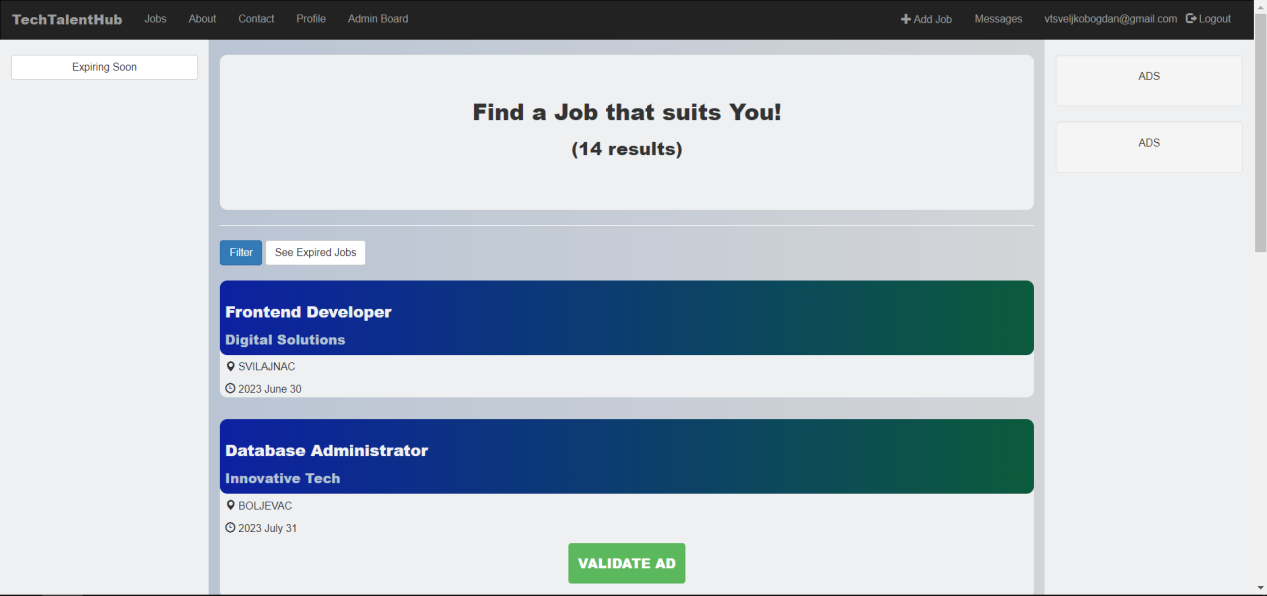
Sada smo ulogovani u korisnika koji se vodi kao kompanija. On može da doda posao pomoću + ADD JOBB tab-om



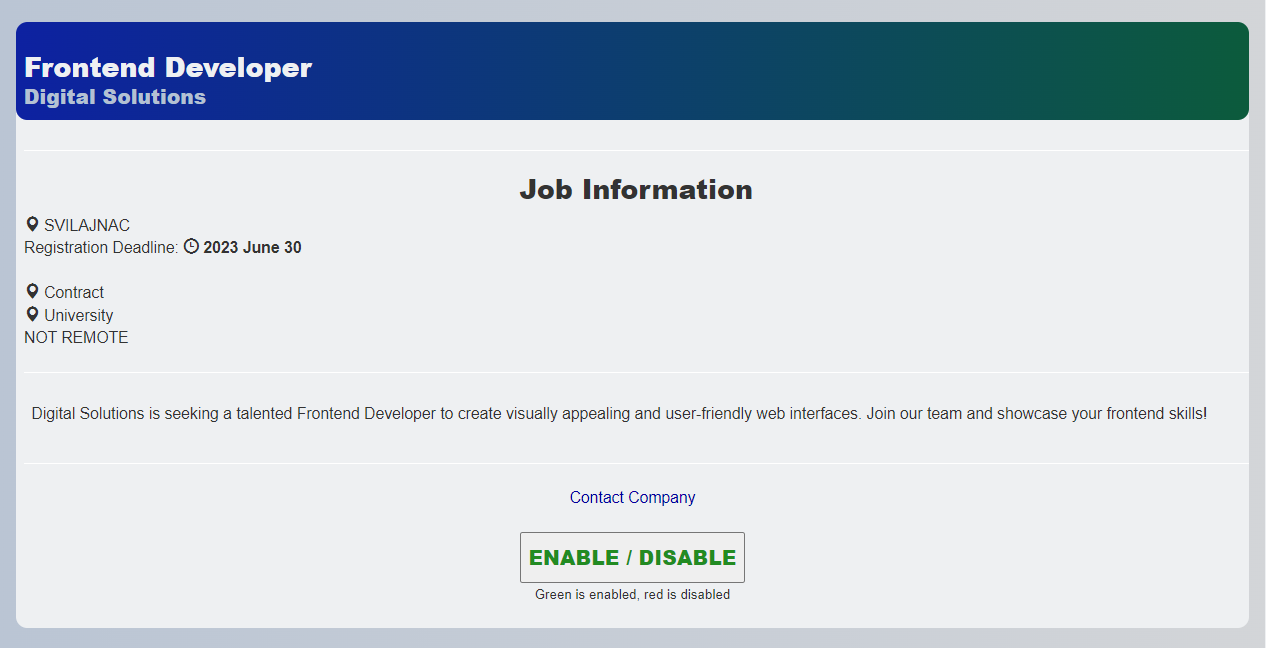




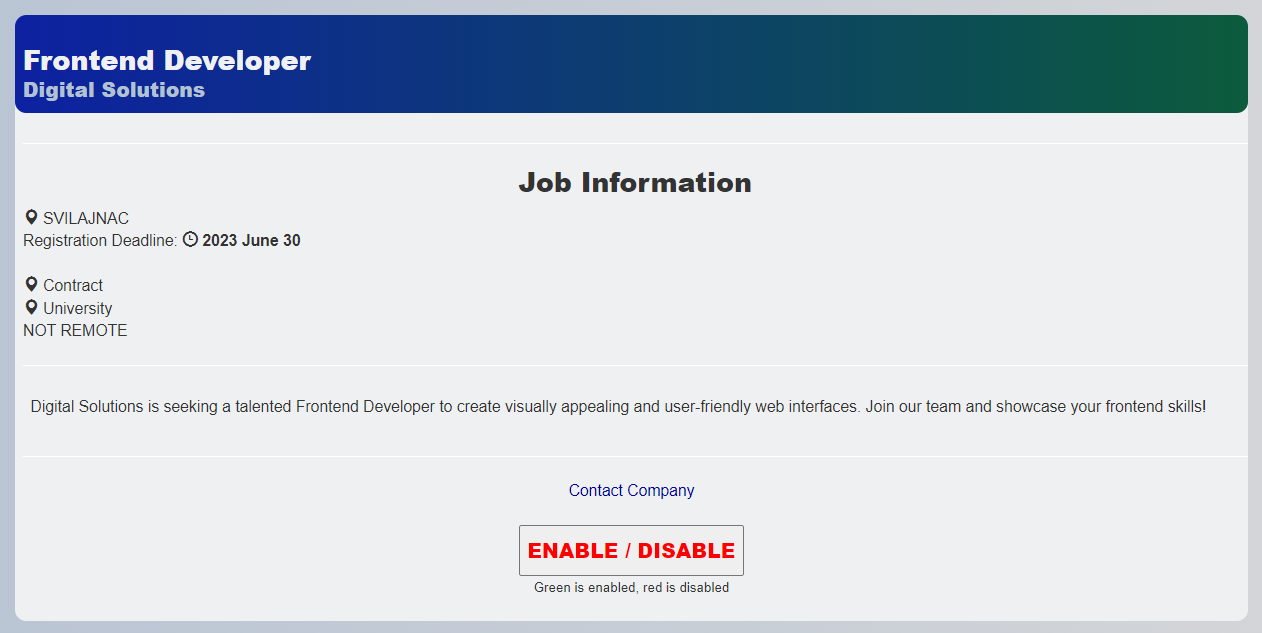
Moramo upisati sve podatke kako bi se posao uneo u bazu, čak i tad, treba potvrdu admina da bi se prikazivao.



Ovo je korisnik-admin. Ima ADMIN BOARD tab i na karticama poslova ima VALIDATE AD, koji će da odobri posao, to jest da mu stavi is\_enabled na 1.



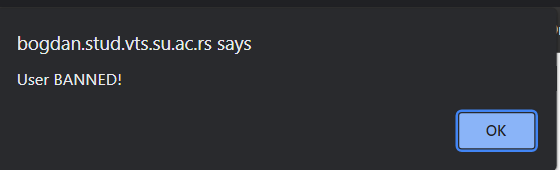
Dugme ENABLE / DISABLE odobrava to jest povlači posao. Ako je zeleno, posao je trenutno odobren, ako je crveno, nije



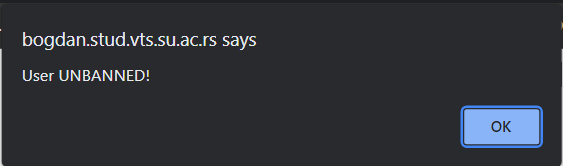
Klikom se promeni na crveno.



Preko admin boarda, admin može da zabrani / odobri ulaz usera, tako što će upisati njegov email ili user\_id.



Ako je user banovan.



Ako je user unbanovan.

**KORIŠĆENA LITERATURA**

W1. <https://mailtrap.io/home>

W2. <https://www.w3schools.com/bootstrap/default.asp>

W3. <https://www.w3schools.com/js/default.asp>

W4. <https://www.w3schools.com/sql/default.asp>

W5. <https://www.w3schools.com/php/default.asp>

W6. <https://www.w3schools.com/html/default.asp>

W7. <http://www.jquery.com>

W8. <https://www.w3schools.com/js/js_ajax_intro.asp>

W9. <https://people.vts.su.ac.rs/~chole/wp/primeri-peldak/>