**SZABADKAI MŰSZAKI SZAKFŐISKOLA**

**IT ÁLLÁS BÖRZE -  PROJEKTUM**

**Web programozás tantárgyból**

|  |  |
| --- | --- |
| **HALLGATÓ(K)**  **Tojzán Sándor** | **MENTOR**  **prof. dr Zlatko Čović** |
| **SZABADKA, 2024.** | |

Tartalom

[1. A feladatkör meghatározása 3](#_Toc156903382)

[2. Kidolgozás 4](#_Toc156903383)

[3. Az adatbázis struktúrája (MySQL Workbench) 5](#_Toc156903384)

[4. A projekt működésének leírása 6](#_Toc156903385)

[5. Felhasznált irodalom 7](#_Toc156903386)

# A feladatkör meghatározása

A projekt lényege az volt, hogy össze kellett rakni egy olyan weboldalt, ahol a cégek felrakhatnak állásokat, amikre a felhasználók jelentkezhetnek.

Ezt azáltal teszik meg, hogy üzenetet írnak a cégeknek. Így megbírják beszélni a részleteket, és hogy fel-e fogják venni a felhasználót, vagy sem.

A weboldalnak van számos hasznos funkciója, ami kisegít ebben a célban, és megkönnyíti a felhasználását.

És természetesen van egy admin felhasználó is, akinek engedélyeznie kell a munkákat, amiket a cégek felakarnak rakni, és aki még képes kitiltani felhasználókat, akik gondot okoznának.

És ha van már egy account az email-jén, akkor a localhost/web/login.php-vel bebír jelentkezni.

# Kidolgozás

Urlap.php:

?php

session\_start();

if (empty($\_SESSION['csrf\_token'])) {

$\_SESSION['csrf\_token'] = bin2hex(random\_bytes(32));

}

$csrf\_token = $\_SESSION['csrf\_token'];

?>

<!DOCTYPE html>

<html lang="hu">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Regisztráció</title>

<link href="startbootstrap-agency-gh-pages\css\styles.css" rel="stylesheet" type="text/css">

</head>

<body>

<div class="container mt-5">

<h2>Regisztráció</h2>

<form id="registrationForm" action="register.php" method="POST">

<input type="hidden" name="csrf\_token" value="<?php echo htmlspecialchars($csrf\_token, ENT\_QUOTES, 'UTF-8'); ?>">

<div class="mb-3">

<label for="email" class="form-label">Email</label>

<input type="email" class="form-control" id="email" name="email" required>

</div>

<div class="mb-3">

<label for="username" class="form-label">Felhasználónév</label>

<input type="text" class="form-control" id="username" name="username" required minlength="3" maxlength="16" pattern="^[a-zA-Z0-9\_]+$">

</div>

<div class="mb-3">

<label for="password" class="form-label">Jelszó</label>

<input type="password" class="form-control" id="password" name="password" required minlength="8">

</div>

<div class="mb-3">

<label for="confirm\_password" class="form-label">Jelszó megerősítése</label>

<input type="password" class="form-control" id="confirm\_password" name="confirm\_password" required>

</div>

<div class="mb-3">

<label for="userType" class="form-label">Regisztráció típusa</label>

<select class="form-select" id="userType" name="userType" required>

<option value="individual">Egyéni Felhasználó</option>

<option value="company">Céges Felhasználó</option>

</select>

</div>

<div id="companyFields" style="display: none;">

<div class="mb-3">

<label for="companyName" class="form-label">Cég neve</label>

<input type="text" class="form-control" id="companyName" name="companyName">

</div>

<div class="mb-3">

<label for="companyWebsite" class="form-label">Cég weboldala</label>

<input type="url" class="form-control" id="companyWebsite" name="companyWebsite" pattern="https?://.+">

</div>

<div class="mb-3">

<label for="companyAddress" class="form-label">Cég címe</label>

<input type="text" class="form-control" id="companyAddress" name="companyAddress">

</div>

<div class="mb-3">

<label for="companyDescription" class="form-label">Cégleírás</label>

<textarea class="form-control" id="companyDescription" name="companyDescription"></textarea>

</div>

</div>

<button type="submit" class="btn btn-primary">Regisztráció</button>

</form>

</div>

<script>

document.getElementById('userType').addEventListener('change', function() {

var companyFields = document.getElementById('companyFields');

if (this.value === 'company') {

companyFields.style.display = 'block';

} else {

companyFields.style.display = 'none';

}

});

</script>

</body>

</html>  
  
----------------------------------

register.php:

<?php

require 'db\_config.php';

require 'vendor/autoload.php';

use PHPMailer\PHPMailer\PHPMailer;

use PHPMailer\PHPMailer\Exception;

session\_start();

if (empty($\_SESSION['csrf\_token'])) {

$\_SESSION['csrf\_token'] = bin2hex(random\_bytes(32));

}

$csrf\_token = $\_SESSION['csrf\_token'];

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

if (!isset($\_POST['csrf\_token']) || $\_POST['csrf\_token'] !== $\_SESSION['csrf\_token']) {

die("Invalid CSRF token");

}

function validateInput($data) {

return htmlspecialchars(stripslashes(trim($data)));

}

$errors = [];

$success = '';

$email = validateInput($\_POST["email"]);

$username = validateInput($\_POST["username"]);

$password = validateInput($\_POST["password"]);

$confirm\_password = validateInput($\_POST["confirm\_password"]);

$userType = validateInput($\_POST["userType"]);

$companyName = $companyWebsite = $companyAddress = $companyDescription = "";

if (!filter\_var($email, FILTER\_VALIDATE\_EMAIL)) {

$errors[] = "Invalid email format.";

}

if (!preg\_match("/^(?=.\*[A-Za-z])(?=.\*\d)[A-Za-z\d]{8,}$/", $password)) {

$errors[] = "Password must be at least 8 characters long and include a mix of letters and numbers.";

}

if ($userType == "company") {

$companyName = validateInput($\_POST["companyName"]);

$companyWebsite = validateInput($\_POST["companyWebsite"]);

$companyAddress = validateInput($\_POST["companyAddress"]);

$companyDescription = validateInput($\_POST["companyDescription"]);

}

if ($password !== $confirm\_password) {

$errors[] = "A jelszavak nem egyeznek!";

}

try {

$pdo = new PDO($dsn, $db\_username, $db\_password, $options);

$stmt = $pdo->prepare("SELECT id FROM users WHERE email = :email LIMIT 1");

$stmt->execute(['email' => $email]);

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

$errors[] = "Ez az e-mail cím már foglalt!";

}

} catch (PDOException $e) {

$errors[] = "Adatbázis hiba: " . $e->getMessage();

}

if (empty($errors)) {

$activation\_token = bin2hex(random\_bytes(16));

$hashed\_password = password\_hash($password, PASSWORD\_BCRYPT);

try {

$pdo->beginTransaction();

$query = "INSERT INTO users (email, username, password\_hash, is\_active, activation\_token, role, company\_name, company\_website, company\_address, company\_description)

VALUES (:email, :username, :password\_hash, :is\_active, :activation\_token, :role, :company\_name, :company\_website, :company\_address, :company\_description)";

$stmt = $pdo->prepare($query);

$stmt->execute([

':email' => $email,

':username' => $username,

':password\_hash' => $hashed\_password,

':is\_active' => 0,

':activation\_token' => $activation\_token,

':role' => $userType,

':company\_name' => $companyName,

':company\_website' => $companyWebsite,

':company\_address' => $companyAddress,

':company\_description' => $companyDescription,

]);

$pdo->commit();

$mail = new PHPMailer(true);

try {

$mail->isSMTP();

$mail->Host = 'smtp.gmail.com';

$mail->SMTPAuth = true;

$mail->Username = 'allasborzeit@gmail.com';

$mail->Password = 'yfhofepazbuqtnpb';

$mail->SMTPSecure = PHPMailer::ENCRYPTION\_STARTTLS;

$mail->Port = 587;

$mail->setFrom('allasborzeit@gmail.com', 'Allas borze IT');

$mail->addAddress($email, $username);

$activation\_link = "http://localhost/web/activate.php?token=" . $activation\_token;

$mail->isHTML(true);

$mail->Subject = 'Account Activation';

$mail->Body = "Please click the following link to activate your account: <a href='$activation\_link'>$activation\_link</a>";

$mail->AltBody = "Please click the following link to activate your account: $activation\_link";

$mail->send();

echo 'Registration successful! Please check your email to activate your account.';

} catch (Exception $e) {

echo "Registration successful, but the activation email could not be sent. Mailer Error: {$mail->ErrorInfo}";

}

} catch (PDOException $e) {

$pdo->rollBack();

$errors[] = "Database error: " . $e->getMessage();

}

}

}

?>

<!DOCTYPE html>

<html lang="hu">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Regisztráció</title>

<link href="startbootstrap-agency-gh-pages\css\styles.css" rel="stylesheet" type="text/css">

</head>

<body>

<div class="container mt-5">

<h2>Regisztráció</h2>

<?php if (!empty($errors)): ?>

<div class="alert alert-danger">

<?php foreach ($errors as $error): ?>

<p><?php echo $error; ?></p>

<?php endforeach; ?>

</div>

<?php endif; ?>

<?php if (!empty($success)): ?>

<div class="alert alert-success">

<p><?php echo $success; ?></p>

</div>

<?php endif; ?>

<form method="POST" action="register.php">

<input type="hidden" name="csrf\_token" value="<?php echo htmlspecialchars($csrf\_token, ENT\_QUOTES, 'UTF-8'); ?>">

</form>

<a href="Urlap.php" class="btn btn-primary">Vissza a regisztrációhoz</a>

</div>

</body>

</html>  
  
---------------------------------------------  
  
post\_job.php:  
  
  
<?php

session\_start();

require 'db\_config.php';

if ($\_SESSION['role'] !== 'company') {

header("Location: login.php");

exit;

}

if (!isset($\_SESSION['user\_id']) || $\_SESSION['role'] !== 'company') {

echo "You must be logged in as a company to post a job.";

exit;

}

$errors = [];

$success = '';

function validateInput($data) {

return htmlspecialchars(stripslashes(trim($data)));

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$position = validateInput($\_POST["position"]);

$description = validateInput($\_POST["description"]);

$location = validateInput($\_POST["location"]);

$deadline = validateInput($\_POST["deadline"]);

$company\_name = validateInput($\_POST["company\_name"]);

$company\_id = $\_SESSION['user\_id'];

try {

$pdo = new PDO($dsn, $db\_username, $db\_password, $options);

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

$stmt = $pdo->prepare("INSERT INTO jobs (company\_id, job\_title, job\_description, location, application\_deadline, is\_active, is\_expired, company\_name) VALUES (:company\_id, :position, :description, :location, :deadline, 0, 0, :company\_name)");

$stmt->execute([

'company\_id' => $company\_id,

'position' => $position,

'description' => $description,

'location' => $location,

'deadline' => $deadline,

'company\_name' => $company\_name

]);

echo "Állásajánlat sikeresen feladva. Várja meg, amíg az adminisztrátor jóváhagyja.";

} catch (PDOException $e) {

echo "Adatbázis hiba: " . $e->getMessage();

}

}

?>

<!DOCTYPE html>

<html lang="hu">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Álláshirdetés Közzététele</title>

<link href="startbootstrap-agency-gh-pages\css\styles.css" rel="stylesheet" type="text/css">

</head>

<body>

<div class="container mt-5">

<h2>Új Álláshirdetés Közzététele</h2>

<form action="post\_job.php" method="POST">

<div class="mb-3">

<label for="position" class="form-label">Állás címe</label>

<input type="text" class="form-control" id="position" name="position" required>

</div>

<div class="mb-3">

<label for="description" class="form-label">Állásleírás</label>

<textarea class="form-control" id="description" name="description" required></textarea>

</div>

<div class="mb-3">

<label for="location" class="form-label">Helyszín</label>

<input type="text" class="form-control" id="location" name="location" required>

</div>

<div class="mb-3">

<label for="deadline" class="form-label">Jelentkezési határidő</label>

<input type="date" class="form-control" id="deadline" name="deadline" required>

</div>

<div class="mb-3">

<label for="company\_name" class="form-label">Cég neve</label>

<input type="text" class="form-control" id="company\_name" name="company\_name" required>

</div>

<button type="submit" class="btn btn-primary">Hirdetés Közzététele</button>

</form>

</div>

</body>

</html>  
  
-----------------------------------------------------  
  
password\_reset.php:  
  
  
<?php

require 'db\_config.php';

session\_start();

function validateInput($data) {

return htmlspecialchars(trim($data));

}

$errors = [];

$success = '';

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$token = validateInput($\_POST["token"]);

$password = validateInput($\_POST["password"]);

$confirm\_password = validateInput($\_POST["confirm\_password"]);

if ($password !== $confirm\_password) {

$errors[] = 'A jelszavak nem egyeznek.';

} else {

try {

$pdo = new PDO($dsn, $db\_username, $db\_password, $options);

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

$stmt = $pdo->prepare("SELECT \* FROM password\_resets WHERE token = :token AND expires > NOW() LIMIT 1");

$stmt->execute(['token' => $token]);

$reset = $stmt->fetch(PDO::FETCH\_ASSOC);

if ($reset) {

$hashed\_password = password\_hash($password, PASSWORD\_BCRYPT);

$stmt = $pdo->prepare("UPDATE users SET password\_hash = :password WHERE email = :email");

$stmt->execute(['password' => $hashed\_password, 'email' => $reset['email']]);

$stmt = $pdo->prepare("DELETE FROM password\_resets WHERE token = :token");

$stmt->execute(['token' => $token]);

$success = 'A jelszavad sikeresen visszaállítva. Most már bejelentkezhetsz az új jelszóval.';

} else {

$errors[] = 'Érvénytelen vagy lejárt token.';

}

} catch (PDOException $e) {

$errors[] = "Adatbázis hiba: " . $e->getMessage();

}

}

} else if (isset($\_GET['token'])) {

$token = validateInput($\_GET['token']);

} else {

$errors[] = 'Érvénytelen token.';

}

?>

<!DOCTYPE html>

<html lang="hu">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Jelszó Visszaállítása</title>

<link href="startbootstrap-agency-gh-pages\css\styles.css" rel="stylesheet" type="text/css">

</head>

<body>

<div class="container mt-5">

<h2>Jelszó Visszaállítása</h2>

<?php if (!empty($errors)): ?>

<div class="alert alert-danger">

<?php foreach ($errors as $error): ?>

<p><?php echo $error; ?></p>

<?php endforeach; ?>

</div>

<?php endif; ?>

<?php if ($success): ?>

<div class="alert alert-success">

<p><?php echo $success; ?></p>

</div>

<?php elseif (isset($token)): ?>

<form action="password\_reset.php" method="POST">

<input type="hidden" name="token" value="<?php echo $token; ?>">

<div class="mb-3">

<label for="password" class="form-label">Új Jelszó</label>

<input type="password" class="form-control" id="password" name="password" required>

</div>

<div class="mb-3">

<label for="confirm\_password" class="form-label">Jelszó Megerősítése</label>

<input type="password" class="form-control" id="confirm\_password" name="confirm\_password" required>

</div>

<button type="submit" class="btn btn-primary">Jelszó Visszaállítása</button>

</form>

<?php endif; ?>

</div>

</body>

</html>

-----------------------------------------------  
  
  
activate.php:  
  
<?php

require 'db\_config.php';

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

$token = $\_GET['token'];

$stmt = $pdo->prepare("UPDATE users SET is\_active = 1, activation\_token = NULL WHERE activation\_token = :token");

$stmt->execute(['token' => $token]);

if ($stmt->rowCount()) {

echo 'Sikeres account aktiváció!';

} else {

echo 'Téves aktivációs link, vagy az account már aktiválva van';

}

} else {

echo 'Nincs megadva az aktivációs token';

}

?>

--------------------------------------------------  
  
change-password.php:  
  
  
<?php

require 'db\_config.php';

session\_start();

if (!isset($\_SESSION['user\_id'])) {

header("Location: login.php");

exit;

}

$errors = [];

$success = '';

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$current\_password = htmlspecialchars(trim($\_POST["current\_password"]));

$new\_password = htmlspecialchars(trim($\_POST["new\_password"]));

$confirm\_password = htmlspecialchars(trim($\_POST["confirm\_password"]));

$user\_id = $\_SESSION['user\_id'];

try {

$pdo = new PDO($dsn, $db\_username, $db\_password, $options);

$stmt = $pdo->prepare("SELECT password\_hash FROM users WHERE id = :user\_id");

$stmt->execute(['user\_id' => $user\_id]);

$user = $stmt->fetch(PDO::FETCH\_ASSOC);

if ($user && password\_verify($current\_password, $user['password\_hash'])) {

if ($new\_password === $confirm\_password) {

$hashed\_password = password\_hash($new\_password, PASSWORD\_BCRYPT);

$stmt = $pdo->prepare("UPDATE users SET password\_hash = :password\_hash WHERE id = :user\_id");

$stmt->execute(['password\_hash' => $hashed\_password, 'user\_id' => $user\_id]);

$success = "Jelszó sikeresen megváltoztatva!";

} else {

$errors[] = "Az új jelszavak nem egyeznek!";

}

} else {

$errors[] = "A jelenlegi jelszó helytelen!";

}

} catch (PDOException $e) {

$errors[] = "Adatbázis hiba: " . $e->getMessage();

}

}

?>

<!DOCTYPE html>

<html lang="hu">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Jelszó megváltoztatása</title>

<link href="startbootstrap-agency-gh-pages\css\styles.css" rel="stylesheet" type="text/css">

</head>

<body>

<div class="container mt-5">

<h2>Jelszó megváltoztatása</h2>

<?php if (!empty($errors)): ?>

<div class="alert alert-danger">

<?php foreach ($errors as $error): ?>

<p><?php echo $error; ?></p>

<?php endforeach; ?>

</div>

<?php endif; ?>

<?php if (!empty($success)): ?>

<div class="alert alert-success">

<p><?php echo $success; ?></p>

</div>

<?php endif; ?>

<form action="change\_password.php" method="POST">

<div class="mb-3">

<label for="current\_password" class="form-label">Jelenlegi jelszó</label>

<input type="password" class="form-control" id="current\_password" name="current\_password" required>

</div>

<div class="mb-3">

<label for="new\_password" class="form-label">Új jelszó</label>

<input type="password" class="form-control" id="new\_password" name="new\_password" required>

</div>

<div class="mb-3">

<label for="confirm\_password" class="form-label">Új jelszó megerősítése</label>

<input type="password" class="form-control" id="confirm\_password" name="confirm\_password" required>

</div>

<button type="submit" class="btn btn-primary">Jelszó megváltoztatása</button>

</form>

<a href="user\_dashboard.php" class="btn btn-secondary mt-3">Vissza a profilhoz</a>

</div>

</body>

</html>  
------------------------------------------------  
  
edit\_profile.php:  
  
  
<?php

require 'db\_config.php';

session\_start();

if (!isset($\_SESSION['user\_id'])) {

header("Location: login.html");

exit;

}

$errors = [];

$success = '';

$user\_id = $\_SESSION['user\_id'];

try {

$pdo = new PDO($dsn, $db\_username, $db\_password, $options);

$stmt = $pdo->prepare("SELECT \* FROM users WHERE id = :user\_id");

$stmt->execute(['user\_id' => $user\_id]);

$user = $stmt->fetch(PDO::FETCH\_ASSOC);

if (!$user) {

$errors[] = "Felhasználó nem található.";

}

} catch (PDOException $e) {

$errors[] = "Adatbázis hiba: " . $e->getMessage();

}

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$username = htmlspecialchars(trim($\_POST["username"]));

$email = htmlspecialchars(trim($\_POST["email"]));

$companyName = htmlspecialchars(trim($\_POST["companyName"]));

$companyWebsite = htmlspecialchars(trim($\_POST["companyWebsite"]));

$companyAddress = htmlspecialchars(trim($\_POST["companyAddress"]));

$companyDescription = htmlspecialchars(trim($\_POST["companyDescription"]));

try {

$stmt = $pdo->prepare("SELECT id FROM users WHERE email = :email AND id != :user\_id");

$stmt->execute(['email' => $email, 'user\_id' => $user\_id]);

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

$errors[] = "Ez az e-mail cím már foglalt!";

}

if (empty($errors)) {

$stmt = $pdo->prepare("UPDATE users SET username = :username, email = :email, company\_name = :companyName, company\_website = :companyWebsite, company\_address = :companyAddress, company\_description = :companyDescription WHERE id = :user\_id");

$stmt->execute([

'username' => $username,

'email' => $email,

'companyName' => $companyName,

'companyWebsite' => $companyWebsite,

'companyAddress' => $companyAddress,

'companyDescription' => $companyDescription,

'user\_id' => $user\_id

]);

$success = "Profil sikeresen frissítve!";

$\_SESSION['username'] = $username;

}

} catch (PDOException $e) {

$errors[] = "Adatbázis hiba: " . $e->getMessage();

}

}

?>

<!DOCTYPE html>

<html lang="hu">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Profil szerkesztése</title>

<link href="startbootstrap-agency-gh-pages\css\styles.css" rel="stylesheet" type="text/css">

</head>

<body>

<div class="container mt-5">

<h2>Profil szerkesztése</h2>

<?php if (!empty($errors)): ?>

<div class="alert alert-danger">

<?php foreach ($errors as $error): ?>

<p><?php echo $error; ?></p>

<?php endforeach; ?>

</div>

<?php endif; ?>

<?php if (!empty($success)): ?>

<div class="alert alert-success">

<p><?php echo $success; ?></p>

</div>

<?php endif; ?>

<form action="edit\_profile.php" method="POST">

<div class="mb-3">

<label for="username" class="form-label">Felhasználónév</label>

<input type="text" class="form-control" id="username" name="username" value="<?php echo htmlspecialchars($user['username']); ?>" required>

</div>

<div class="mb-3">

<label for="email" class="form-label">Email</label>

<input type="email" class="form-control" id="email" name="email" value="<?php echo htmlspecialchars($user['email']); ?>" required>

</div>

<div class="mb-3">

<label for="companyName" class="form-label">Cég neve</label>

<input type="text" class="form-control" id="companyName" name="companyName" value="<?php echo htmlspecialchars($user['company\_name']); ?>">

</div>

<div class="mb-3">

<label for="companyWebsite" class="form-label">Cég weboldala</label>

<input type="url" class="form-control" id="companyWebsite" name="companyWebsite" value="<?php echo htmlspecialchars($user['company\_website']); ?>">

</div>

<div class="mb-3">

<label for="companyAddress" class="form-label">Cég címe</label>

<input type="text" class="form-control" id="companyAddress" name="companyAddress" value="<?php echo htmlspecialchars($user['company\_address']); ?>">

</div>

<div class="mb-3">

<label for="companyDescription" class="form-label">Cég leírása</label>

<textarea class="form-control" id="companyDescription" name="companyDescription"><?php echo htmlspecialchars($user['company\_description']); ?></textarea>

</div>

<button type="submit" class="btn btn-primary">Mentés</button>

</form>

<a href="user\_dashboard.php" class="btn btn-secondary mt-3">Vissza a profilhoz</a>

</div>

</body>

</html>  
  
------------------------------------------------  
  
  
logout.php:  
  
<?php

session\_start();

session\_unset();

session\_destroy();

header("Location: login.php");

exit();

?>  
------------------------------------------------  
  
  
job\_details.php:  
  
<?php

session\_start();

require 'db\_config.php';

if (!isset($\_GET['id']) || empty($\_GET['id'])) {

die("Érvénytelen kérés.");

}

$job\_id = intval($\_GET['id']);

try {

$pdo = new PDO($dsn, $db\_username, $db\_password, $options);

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

$stmt = $pdo->prepare("SELECT \* FROM jobs WHERE id = :id");

$stmt->execute(['id' => $job\_id]);

$job = $stmt->fetch(PDO::FETCH\_ASSOC);

if (!$job) {

die("A munka nem található.");

}

} catch (PDOException $e) {

die("Adatbázis hiba: " . $e->getMessage());

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href="startbootstrap-agency-gh-pages\css\styles.css" rel="stylesheet" type="text/css">

<title>Munka Részletek</title>

</head>

<body>

<div class="container mt-5">

<h1><?php echo htmlspecialchars($job['job\_title']); ?></h1>

<p><strong>Cég:</strong> <?php echo htmlspecialchars($job['company\_name']); ?></p>

<p><strong>Leírás:</strong> <?php echo nl2br(htmlspecialchars($job['job\_description'])); ?></p>

<p><strong>Lokáció:</strong> <?php echo htmlspecialchars($job['location']); ?></p>

<p><strong>Kategória:</strong> <?php echo htmlspecialchars($job['category']); ?></p>

<p><strong>Jelentkezési Határidő:</strong> <?php echo htmlspecialchars($job['application\_deadline']); ?></p>

<a href="send\_message.php?receiver\_id=<?php echo $job['company\_id']; ?>" class="btn btn-primary">Üzenet Küldés</a>

<a href="job\_listings.php" class="btn btn-primary">Vissza a Munka Listázásához</a>

</div>

</body>

</html>  
  
----------------------------------------------------  
  
  
user\_dashboard.php:  
  
  
<?php

session\_start();

if (!isset($\_SESSION['user\_id'])) {

header("Location: login.html");

exit;

}

require 'db\_config.php';

$user\_id = $\_SESSION['user\_id'];

try {

$pdo = new PDO($dsn, $db\_username, $db\_password, $options);

$stmt = $pdo->prepare("SELECT \* FROM users WHERE id = :user\_id");

$stmt->execute(['user\_id' => $user\_id]);

$user = $stmt->fetch(PDO::FETCH\_ASSOC);

if (!$user) {

die("Felhasználó nem található.");

}

} catch (PDOException $e) {

die("Adatbázis hiba: " . $e->getMessage());

}

?>

<!DOCTYPE html>

<html lang="hu">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Felhasználói felület</title>

<link href="startbootstrap-agency-gh-pages\css\styles.css" rel="stylesheet" type="text/css">

</head>

<body>

<div class="container mt-5">

<h2>Felhasználói profil</h2>

<table class="table">

<tr>

<th>Felhasználónév</th>

<td><?php echo htmlspecialchars($user['username']); ?></td>

</tr>

<tr>

<th>Email</th>

<td><?php echo htmlspecialchars($user['email']); ?></td>

</tr>

<?php if ($user['role'] == 'company'): ?>

<tr>

<th>Cég neve</th>

<td><?php echo htmlspecialchars($user['company\_name']); ?></td>

</tr>

<tr>

<th>Cég weboldala</th>

<td><?php echo htmlspecialchars($user['company\_website']); ?></td>

</tr>

<tr>

<th>Cég címe</th>

<td><?php echo htmlspecialchars($user['company\_address']); ?></td>

</tr>

<tr>

<th>Cég leírása</th>

<td><?php echo htmlspecialchars($user['company\_description']); ?></td>

</tr>

<?php endif; ?>

</table>

<a href="edit\_profile.php" class="btn btn-primary">Profil szerkesztése</a>

<a href="change\_password.php" class="btn btn-warning">Jelszó megváltoztatása</a>

<a href="logout.php" class="btn btn-danger">Kijelentkezés</a>

<a href="job\_listings.php" class="btn btn-primary">Állásajánlatok megtekintése</a>

<nav class="navbar navbar-expand-lg navbar-light bg-light">

<a class="navbar-brand" href="#">Állásportál</a>

<div class="collapse navbar-collapse">

<ul class="navbar-nav ml-auto">

<li class="nav-item">

<a class="nav-link" href="inbox.php">Inbox</a>

</li>

<li class="nav-item">

<a class="nav-link" href="sent\_messages.php">Elküldött Üzenetek</a>

</li>

</ul>

</div>

</nav>

</div>

</body>

</html>  
  
----------------------------------------------------  
  
job\_listings.php:  
  
  
<?php

session\_start();

require 'db\_config.php';

try {

$pdo = new PDO($dsn, $db\_username, $db\_password, $options);

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

$query = "SELECT \* FROM jobs WHERE is\_active = 1";

$params = [];

if (!empty($\_GET['title'])) {

$query .= " AND job\_title LIKE :title";

$params[':title'] = "%" . $\_GET['title'] . "%";

}

if (!empty($\_GET['location'])) {

$query .= " AND location LIKE :location";

$params[':location'] = "%" . $\_GET['location'] . "%";

}

if (!empty($\_GET['category'])) {

$query .= " AND category = :category";

$params[':category'] = $\_GET['category'];

}

if (!empty($\_GET['deadline'])) {

$query .= " AND application\_deadline <= :deadline";

$params[':deadline'] = $\_GET['deadline'];

}

$stmt = $pdo->prepare($query);

$stmt->execute($params);

$jobs = $stmt->fetchAll(PDO::FETCH\_ASSOC);

$stmt = $pdo->query("SELECT \* FROM jobs WHERE is\_active = 1 AND is\_expired = 0");

$jobs = $stmt->fetchAll(PDO::FETCH\_ASSOC);

} catch (PDOException $e) {

echo "Adatbázis hiba: " . $e->getMessage();

$jobs = [];

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href="startbootstrap-agency-gh-pages\css\styles.css" rel="stylesheet" type="text/css">

<title>Munka Listázás</title>

<style>

.expired {

background-color: #f8d7da;

color: #721c24;

}

.expired .card-title {

text-decoration: line-through;

}

</style>

</head>

<body>

<div class="container mt-5">

<h1>Munka Listázás</h1>

<form method="GET" action="job\_listings.php" class="mb-4">

<div class="row">

<div class="col-md-3">

<input type="text" name="title" class="form-control" placeholder="Munka Cím">

</div>

<div class="col-md-3">

<input type="text" name="location" class="form-control" placeholder="Lokáció">

</div>

<div class="col-md-3">

<select name="category" class="form-control">

<option value="">Kategória Kiválasztás</option>

<option value="IT">IT</option>

<option value="Marketing">Marketing</option>

</select>

</div>

<div class="col-md-3">

<input type="date" name="deadline" class="form-control" placeholder="Határidő">

</div>

</div>

<button type="submit" class="btn btn-primary mt-3">Keresés</button>

</form>

<div class="row">

<?php if (!empty($jobs)): ?>

<?php foreach ($jobs as $job): ?>

<?php

$is\_expired = strtotime($job['application\_deadline']) < time();

$card\_class = $is\_expired ? 'expired' : '';

?>

<div class="col-md-4 mb-3">

<div class="card <?php echo $card\_class; ?>">

<div class="card-body">

<h5 class="card-title"><?php echo htmlspecialchars($job['job\_title']); ?></h5>

<p class="card-text"><?php echo htmlspecialchars($job['job\_description']); ?></p>

<p><strong>Location:</strong> <?php echo htmlspecialchars($job['location']); ?></p>

<p><strong>Category:</strong> <?php echo htmlspecialchars($job['category']); ?></p>

<?php if ($is\_expired): ?>

<p><strong>Status:</strong> Expired</p>

<?php endif; ?>

<a href="job\_details.php?id=<?php echo $job['id']; ?>" class="btn btn-primary">Részletek Megtekintése</a>

<a href="send\_message.php?receiver\_id=<?php echo $job['company\_id']; ?>" class="btn btn-primary">Üzenet Küldés</a>

</div>

</div>

</div>

<?php endforeach; ?>

<?php else: ?>

<p>Jelenleg nincs elérhető munka</p>

<?php endif; ?>

</div>

<div class="mt-4">

<?php if (isset($\_SESSION['role'])): ?>

<?php if ($\_SESSION['role'] === 'admin'): ?>

<a href="admin\_categories.php" class="btn btn-secondary">Visszatérés az Admin Műszerfalára</a>

<?php elseif ($\_SESSION['role'] === 'company'): ?>

<a href="company\_dashboard.php" class="btn btn-secondary">Visszatérés a Cég Műszerfalára</a>

<?php else: ?>

<a href="user\_dashboard.php" class="btn btn-secondary">Visszatérés a Felhasználó Műszerfalára</a>

<?php endif; ?>

<?php else: ?>

<a href="Urlap.php" class="btn btn-secondary">Regisztrálás</a>

<?php endif; ?>

</div>

</div>

</body>

</html>  
  
-----------------------------------------------------------  
  
  
company\_dashboard.php:  
  
  
<?php

session\_start();

if (!isset($\_SESSION['user\_id']) || $\_SESSION['role'] !== 'company') {

header("Location: login.php");

exit;

}

require 'db\_config.php';

$user\_id = $\_SESSION['user\_id'];

try {

$pdo = new PDO($dsn, $db\_username, $db\_password, $options);

$stmt = $pdo->prepare("SELECT \* FROM users WHERE id = :user\_id");

$stmt->execute(['user\_id' => $user\_id]);

$user = $stmt->fetch(PDO::FETCH\_ASSOC);

if (!$user) {

die("Felhasználó nem található.");

}

} catch (PDOException $e) {

die("Adatbázis hiba: " . $e->getMessage());

}

?>

<!DOCTYPE html>

<html lang="hu">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Cég Dashboard</title>

<link href="startbootstrap-agency-gh-pages\css\styles.css" rel="stylesheet" type="text/css">

</head>

<body>

<div class="container mt-5">

<h2>Üdvözöljük a Céges Irányítópulton, <?php echo htmlspecialchars($\_SESSION['username']); ?>!</h2>

<table class="table">

<tr>

<th>Felhasználónév</th>

<td><?php echo htmlspecialchars($user['username']); ?></td>

</tr>

<tr>

<th>Email</th>

<td><?php echo htmlspecialchars($user['email']); ?></td>

</tr>

<?php if ($user['role'] == 'company'): ?>

<tr>

<th>Cég neve</th>

<td><?php echo htmlspecialchars($user['company\_name']); ?></td>

</tr>

<tr>

<th>Cég weboldala</th>

<td><?php echo htmlspecialchars($user['company\_website']); ?></td>

</tr>

<tr>

<th>Cég címe</th>

<td><?php echo htmlspecialchars($user['company\_address']); ?></td>

</tr>

<tr>

<th>Cég leírása</th>

<td><?php echo htmlspecialchars($user['company\_description']); ?></td>

</tr>

<?php endif; ?>

</table>

<a href="edit\_profile.php" class="btn btn-primary mt-3">Profil szerkesztése</a>

<a href="change\_password.php" class="btn btn-warning mt-3">Jelszó megváltoztatása</a>

<a href="logout.php" class="btn btn-danger mt-3">Kijelentkezés</a>

<a href="post\_job.php" class="btn btn-primary mt-3">Álláshirdetés közzététele</a>

<a href="job\_listings.php" class="btn btn-primary mt-3">Állásajánlatok megtekintése</a>

<nav class="navbar navbar-expand-lg navbar-light bg-light">

<a class="navbar-brand" href="#">Állásportál</a>

<div class="collapse navbar-collapse">

<ul class="navbar-nav ml-auto">

<li class="nav-item">

<a class="nav-link" href="inbox.php">Inbox</a>

</li>

<li class="nav-item">

<a class="nav-link" href="sent\_messages.php">Elküldött Üzenetek</a>

</li>

</ul>

</div>

</nav>

</div>

</body>

</html>  
  
----------------------------------------------------  
  
password\_reset\_request.php:  
  
  
<?php

require 'db\_config.php';

require 'vendor/autoload.php';

use PHPMailer\PHPMailer\PHPMailer;

use PHPMailer\PHPMailer\Exception;

session\_start();

function validateInput($data) {

return htmlspecialchars(trim($data));

}

$errors = [];

$success = '';

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$email = validateInput($\_POST["email"]);

try {

$pdo = new PDO($dsn, $db\_username, $db\_password, $options);

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

$stmt = $pdo->prepare("SELECT \* FROM users WHERE email = :email LIMIT 1");

$stmt->execute(['email' => $email]);

$user = $stmt->fetch(PDO::FETCH\_ASSOC);

if ($user) {

$token = bin2hex(random\_bytes(32));

$expires = date('Y-m-d H:i:s', strtotime('+1 hour'));

$stmt = $pdo->prepare("INSERT INTO password\_resets (email, token, expires) VALUES (:email, :token, :expires)");

$stmt->execute(['email' => $email, 'token' => $token, 'expires' => $expires]);

$mail = new PHPMailer();

$mail->isSMTP();

$mail->Host = 'smtp.gmail.com';

$mail->SMTPAuth = true;

$mail->Username = 'allasborzeit@gmail.com';

$mail->Password = 'yfhofepazbuqtnpb';

$mail->SMTPSecure = 'tls';

$mail->Port = 587;

$mail->setFrom('allasborzeit@gmail.com', 'Allas borze IT');

$mail->addAddress($email);

$mail->isHTML(true);

$mail->Subject = 'Password Reset Request';

$mail->Body = "Kattints ide a jelszavad visszaállításához: <a href='https://localhost/web/password\_reset.php?token=$token'>Jelszó visszaállítása</a>";

if ($mail->send()) {

$success = 'Egy e-mailt küldtünk a jelszó visszaállítási utasításokkal.';

} else {

$errors[] = 'Nem sikerült elküldeni az e-mailt. Próbáld újra később.';

}

} else {

$errors[] = 'Ez az e-mail cím nincs regisztrálva.';

}

} catch (PDOException $e) {

$errors[] = "Adatbázis hiba: " . $e->getMessage();

}

}

?>

<!DOCTYPE html>

<html lang="hu">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Jelszó visszaállítása</title>

<link href="startbootstrap-agency-gh-pages\css\styles.css" rel="stylesheet" type="text/css">

</head>

<body>

<div class="container mt-5">

<h2>Jelszó visszaállítása</h2>

<?php if (!empty($errors)): ?>

<div class="alert alert-danger">

<?php foreach ($errors as $error): ?>

<p><?php echo $error; ?></p>

<?php endforeach; ?>

</div>

<?php endif; ?>

<?php if ($success): ?>

<div class="alert alert-success">

<p><?php echo $success; ?></p>

</div>

<?php else: ?>

<form action="password\_reset\_request.php" method="POST">

<div class="mb-3">

<label for="email" class="form-label">Email cím</label>

<input type="email" class="form-control" id="email" name="email" required>

</div>

<button type="submit" class="btn btn-primary">Küldés</button>

</form>

<?php endif; ?>

</div>

</body>

</html>  
  
------------------------------------------------

login.php:  
  
<?php

require 'db\_config.php';

session\_start();

function validateInput($data) {

return htmlspecialchars(trim($data));

}

$errors = [];

$success = '';

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$username = validateInput($\_POST["username"]);

$password = validateInput($\_POST["password"]);

try {

$pdo = new PDO($dsn, $db\_username, $db\_password, $options);

$stmt = $pdo->prepare("SELECT \* FROM users WHERE username = :username OR email = :email LIMIT 1");

$stmt->execute(['username' => $username, 'email' => $username]);

$user = $stmt->fetch(PDO::FETCH\_ASSOC);

if ($user && password\_verify($password, $user['password\_hash'])) {

if ($user['banned']) {

$errors[] = "This account has been banned. Please contact the administrator.";

} elseif (!$user['is\_active']) {

$errors[] = "The account is not yet active. Please check your email for the activation link.";

} else {

$\_SESSION['user\_id'] = $user['id'];

$\_SESSION['username'] = $user['username'];

$\_SESSION['role'] = $user['role'];

if ($user['role'] === 'company') {

header("Location: company\_dashboard.php");

} else {

header("Location: user\_dashboard.php");

}

exit;

}

} else {

$errors[] = "Invalid username/email or password!";

}

} catch (PDOException $e) {

$errors[] = "Database error: " . $e->getMessage();

}

}

$token = bin2hex(random\_bytes(32));

$\_SESSION['csrf\_token'] = $token;

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Bejelentkezés</title>

<link href="startbootstrap-agency-gh-pages\css\styles.css" rel="stylesheet" type="text/css">

</head>

<body>

<div class="container mt-5">

<h2>Bejelentkezés</h2>

<?php if (!empty($errors)): ?>

<div class="alert alert-danger">

<?php foreach ($errors as $error): ?>

<p><?php echo $error; ?></p>

<?php endforeach; ?>

</div>

<?php endif; ?>

<form action="login.php" method="POST">

<input type="hidden" name="csrf\_token" value="<?php echo $token; ?>">

<div class="mb-3">

<label for="username" class="form-label">Felhasználónév vagy Email</label>

<input type="text" class="form-control" id="username" name="username" required>

</div>

<div class="mb-3">

<label for="password" class="form-label">Jelszó</label>

<input type="password" class="form-control" id="password" name="password" required>

</div>

<button type="submit" class="btn btn-primary">Bejelentkezés</button>

</form>

<p>Még nincs fiókod? <a href="urlap.php">Regisztráció</a></p>

<p>Elfelejtett jelszó? <a href="password\_reset\_request.php">Jelszó visszaállítása</a></p>

</div>

</body>

</html>  
  
-----------------------------------------------------  
  
admin\_login.php:  
  
  
<?php

require 'db\_config.php';

session\_start();

$errors = [];

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$username = trim($\_POST["username"]);

$password = trim($\_POST["password"]);

if (empty($username) || empty($password)) {

$errors[] = "Please fill in both fields.";

} else {

try {

$pdo = new PDO($dsn, $db\_username, $db\_password, $options);

$stmt = $pdo->prepare("SELECT \* FROM admin\_users WHERE username = :username LIMIT 1");

$stmt->execute(['username' => $username]);

$admin = $stmt->fetch(PDO::FETCH\_ASSOC);

if ($admin && ($password == $admin['password\_hash'])) {

$\_SESSION['role'] = 'admin';

$\_SESSION['admin\_username'] = $admin['username'];

header("Location: admin\_categories.php");

exit();

} else {

$errors[] = "Invalid username or password.";

}

} catch (PDOException $e) {

$errors[] = "Database error: " . $e->getMessage();

}

}

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Admin Bejelentkezés</title>

<link href="startbootstrap-agency-gh-pages\css\styles.css" rel="stylesheet" type="text/css">

</head>

<body>

<div class="container mt-5">

<h2>Admin Bejelentkezés</h2>

<?php if (!empty($errors)): ?>

<div class="alert alert-danger">

<?php foreach ($errors as $error): ?>

<p><?php echo $error; ?></p>

<?php endforeach; ?>

</div>

<?php endif; ?>

<form method="POST">

<div class="mb-3">

<label for="username" class="form-label">Felhasználónév</label>

<input type="text" class="form-control" id="username" name="username" required>

</div>

<div class="mb-3">

<label for="password" class="form-label">Jelszó</label>

<input type="password" class="form-control" id="password" name="password" required>

</div>

<button type="submit" class="btn btn-primary">Bejelentkezés</button>

</form>

</div>

</body>

</html>  
  
--------------------------------------------------  
  
db\_config.php:  
  
  
<?php

$host = 'localhost';

$dbname = 'at';

$username = 'at';

$password = 'VxcKnLGUSlR4ZO0';

$dsn = 'mysql:host=localhost;dbname=at;charset=utf8mb4';

$db\_username = 'at';

$db\_password = 'VxcKnLGUSlR4ZO0';

$options = [

PDO::ATTR\_ERRMODE => PDO::ERRMODE\_EXCEPTION,

PDO::ATTR\_DEFAULT\_FETCH\_MODE => PDO::FETCH\_ASSOC,

PDO::ATTR\_EMULATE\_PREPARES => false,

];

try {

$pdo = new PDO("mysql:host=$host;dbname=$dbname", $username, $password);

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

} catch (PDOException $e) {

die("Database connection failed: " . $e->getMessage());

}

?>  
  
------------------------------------------------  
  
sent\_messages.php:  
  
  
<?php

session\_start();

require 'db\_config.php';

if (!isset($\_SESSION['user\_id'])) {

header("Location: login.php");

exit;

}

$user\_id = $\_SESSION['user\_id'];

try {

$pdo = new PDO($dsn, $db\_username, $db\_password, $options);

$stmt = $pdo->prepare("SELECT m.id, m.message\_text, timestamp, u.username as receiver\_name

FROM messages m

JOIN users u ON m.receiver\_id = u.id

WHERE m.sender\_id = :user\_id

ORDER BY timestamp DESC");

$stmt->execute(['user\_id' => $user\_id]);

$messages = $stmt->fetchAll(PDO::FETCH\_ASSOC);

} catch (PDOException $e) {

echo "Adatbázis hiba: " . $e->getMessage();

$messages = [];

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href="startbootstrap-agency-gh-pages\css\styles.css" rel="stylesheet" type="text/css">

<title>Elküldött Üzenetek</title>

</head>

<body>

<div class="container mt-5">

<h1>Elküldött Üzenetek</h1>

<?php if (!empty($messages)): ?>

<ul class="list-group">

<?php foreach ($messages as $message): ?>

<li class="list-group-item">

<strong>Célzott Személy:</strong> <?php echo htmlspecialchars($message['receiver\_name']); ?><br>

<strong>Üzenet:</strong> <?php echo htmlspecialchars($message['message\_text']); ?><br>

<small><strong>Küldési Dátum:</strong> <?php echo htmlspecialchars($message['timestamp']); ?></small>

</li>

<?php endforeach; ?>

</ul>

<?php else: ?>

<p>Nincs elküldött üzenet</p>

<?php endif; ?>

</div>

</body>

</html>  
  
------------------------------------------------------------  
  
inbox.php:  
  
  
<?php

session\_start();

require 'db\_config.php';

if (!isset($\_SESSION['user\_id'])) {

header("Location: login.php");

exit;

}

$user\_id = $\_SESSION['user\_id'];

try {

$pdo = new PDO($dsn, $db\_username, $db\_password, $options);

$stmt = $pdo->prepare("SELECT m.id, m.message\_text, timestamp, u.username as receiver\_name

FROM messages m

JOIN users u ON m.receiver\_id = u.id

WHERE m.sender\_id = :user\_id

ORDER BY timestamp DESC");

$stmt->execute(['user\_id' => $user\_id]);

$messages = $stmt->fetchAll(PDO::FETCH\_ASSOC);

} catch (PDOException $e) {

echo "Adatbázis hiba: " . $e->getMessage();

$messages = [];

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href="startbootstrap-agency-gh-pages\css\styles.css" rel="stylesheet" type="text/css">

<title>Elküldött Üzenetek</title>

</head>

<body>

<div class="container mt-5">

<h1>Elküldött Üzenetek</h1>

<?php if (!empty($messages)): ?>

<ul class="list-group">

<?php foreach ($messages as $message): ?>

<li class="list-group-item">

<strong>Célzott Személy:</strong> <?php echo htmlspecialchars($message['receiver\_name']); ?><br>

<strong>Üzenet:</strong> <?php echo htmlspecialchars($message['message\_text']); ?><br>

<small><strong>Küldési Dátum:</strong> <?php echo htmlspecialchars($message['timestamp']); ?></small>

</li>

<?php endforeach; ?>

</ul>

<?php else: ?>

<p>Nincs elküldött üzenet</p>

<?php endif; ?>

</div>

</body>

</html>  
  
-----------------------------------------------------------  
  
send\_message.php:  
<?php

session\_start();

require 'db\_config.php';

if (!isset($\_SESSION['user\_id'])) {

header("Location: login.php");

exit;

}

$receiver\_id\_prefilled = isset($\_GET['receiver\_id']) ? intval($\_GET['receiver\_id']) : 0;

$errors = [];

$success = '';

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$receiver\_id = intval($\_POST['receiver\_id']);

$message\_text = trim($\_POST['message\_text']);

$sender\_id = $\_SESSION['user\_id'];

if (empty($message\_text)) {

$errors[] = "Message cannot be empty.";

}

if (empty($errors)) {

try {

$pdo = new PDO($dsn, $db\_username, $db\_password, $options);

$stmt = $pdo->prepare("INSERT INTO messages (sender\_id, receiver\_id, message\_text) VALUES (:sender\_id, :receiver\_id, :message\_text)");

$stmt->execute([

'sender\_id' => $sender\_id,

'receiver\_id' => $receiver\_id,

'message\_text' => $message\_text

]);

$success = "Message sent successfully.";

} catch (PDOException $e) {

$errors[] = "Database error: " . $e->getMessage();

}

}

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href="startbootstrap-agency-gh-pages\css\styles.css" rel="stylesheet" type="text/css">

<title>Üzenet Küldés</title>

</head>

<body>

<div class="container mt-5">

<h1>Üzenet Küldés</h1>

<?php if (!empty($errors)): ?>

<div class="alert alert-danger">

<?php echo implode('<br>', $errors); ?>

</div>

<?php endif; ?>

<?php if ($success): ?>

<div class="alert alert-success">

<?php echo $success; ?>

</div>

<?php endif; ?>

<form action="send\_message.php" method="POST">

<div class="mb-3">

<label for="receiver\_id" class="form-label">Recipiens</label>

<select name="receiver\_id" id="receiver\_id" class="form-control" required>

<?php

$stmt = $pdo->query("SELECT id, username FROM users WHERE id != {$\_SESSION['user\_id']}");

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

$selected = ($row['id'] == $receiver\_id\_prefilled) ? 'selected' : '';

echo "<option value='{$row['id']}' $selected>{$row['username']}</option>";

}

?>

</select>

</div>

<div class="mb-3">

<label for="message\_text" class="form-label">Üzenet</label>

<textarea name="message\_text" id="message\_text" class="form-control" required></textarea>

</div>

<button type="submit" class="btn btn-primary">Üzenet Küldés</button>

<a href="job\_listings.php" class="btn btn-primary">Vissza a Listához</a>

</form>

</div>

</body>

</html>  
  
---------------------------------------------------  
  
reject\_job.php:  
  
<?php

require 'db\_config.php';

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

$job\_id = intval($\_GET['id']);

try {

$pdo = new PDO($dsn, $db\_username, $db\_password, $options);

$stmt = $pdo->prepare("DELETE FROM jobs WHERE id = :id");

$stmt->execute(['id' => $job\_id]);

header("Location: admin\_job\_approval.php?status=rejected");

exit;

} catch (PDOException $e) {

die("Adatbázis hiba: " . $e->getMessage());

}

} else {

die("Érvénytelen kérés.");

}

?>  
-----------------------------------------------------  
  
approve\_job.php:  
  
  
<?php

session\_start();

require 'db\_config.php';

if ($\_SESSION['role'] !== 'admin') {

header("Location: login.php");

exit;

}

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

$job\_id = $\_GET['id'];

try {

$pdo = new PDO($dsn, $db\_username, $db\_password, $options);

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

$stmt = $pdo->prepare("UPDATE jobs SET is\_active = 1 WHERE id = :job\_id");

$stmt->execute(['job\_id' => $job\_id]);

header("Location: admin\_job\_approval.php?status=approved");

exit;

} catch (PDOException $e) {

echo "Database error: " . $e->getMessage();

}

} else {

header("Location: admin\_job\_approval.php?status=error");

exit;

}

?>  
----------------------------------------------------------  
  
admin\_job\_approval.php:  
  
  
<?php

require 'db\_config.php';

session\_start();

if (!isset($\_SESSION['user\_id']) || $\_SESSION['role'] !== 'admin') {

header('Location: login.php');

exit;

}

try {

$pdo = new PDO($dsn, $db\_username, $db\_password, $options);

$stmt = $pdo->prepare("SELECT \* FROM jobs WHERE is\_active = 0 ORDER BY application\_deadline DESC");

$stmt->execute();

$pending\_jobs = $stmt->fetchAll(PDO::FETCH\_ASSOC);

if (!$pending\_jobs) {

echo "Nincsenek jóváhagyásra váró állások.";

}

} catch (PDOException $e) {

die("Adatbázis hiba: " . $e->getMessage());

}

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

echo "<tr>";

echo "<td>" . htmlspecialchars($row['position']) . "</td>";

echo "<td>" . htmlspecialchars($row['description']) . "</td>";

echo "<td>

<a href='approve\_job.php?id=" . intval($row['id']) . "' class='btn btn-success'>Elfogadás</a>

<a href='reject\_job.php?id=" . intval($row['id']) . "' class='btn btn-danger'>Elutasítás</a>

</td>";

echo "</tr>";

}

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

if ($\_GET['status'] == 'approved') {

echo "<div class='alert alert-success'>A munka elfogadva.</div>";

} elseif ($\_GET['status'] == 'rejected') {

echo "<div class='alert alert-danger'>A munka elutasítva.</div>";

}

}

?>

<!DOCTYPE html>

<html lang="hu">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Állásajánlatok Jóváhagyása</title>

<link href="startbootstrap-agency-gh-pages\css\styles.css" rel="stylesheet" type="text/css">

</head>

<body>

<div class="container mt-5">

<h2>Állásajánlatok Jóváhagyása</h2>

<table class="table table-bordered">

<thead>

<tr>

<th>Pozíció</th>

<th>Leírás</th>

<th>Cég</th>

<th>Helyszín</th>

<th>Határidő</th>

<th>Műveletek</th>

</tr>

</thead>

<tbody>

<?php if (!empty($pending\_jobs)) : ?>

<?php foreach ($pending\_jobs as $job) : ?>

<tr>

<td><?php echo htmlspecialchars($job['job\_title']); ?></td>

<td><?php echo htmlspecialchars($job['job\_description']); ?></td>

<td><?php echo htmlspecialchars($job['company\_name']); ?></td>

<td><?php echo htmlspecialchars($job['location']); ?></td>

<td><?php echo htmlspecialchars($job['application\_deadline']); ?></td>

<td>

<a href="approve\_job.php?id=<?php echo $job['id']; ?>" class="btn btn-success">Jóváhagyás</a>

<a href="reject\_job.php?id=<?php echo $job['id']; ?>" class="btn btn-danger">Elutasítás</a>

</td>

</tr>

<?php endforeach; ?>

<?php endif; ?>

</tbody>

</table>

<a href="admin\_categories.php" class="btn btn-secondary mt-3">Vissza a Kategóriákhoz</a>

</div>

</body>

</html>  
  
-----------------------------------------------------------  
  
manage\_users.php:  
  
  
<?php

require 'db\_config.php';

session\_start();

if (!isset($\_SESSION['role']) || $\_SESSION['role'] !== 'admin') {

header("Location: admin\_login.php");

exit();

}

$errors = [];

$success = '';

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

if (isset($\_POST['ban\_user'])) {

$user\_id = $\_POST['user\_id'];

try {

$pdo = new PDO($dsn, $db\_username, $db\_password, $options);

$stmt = $pdo->prepare("UPDATE users SET banned = 1 WHERE id = :id");

$stmt->execute(['id' => $user\_id]);

$success = "User has been banned successfully!";

} catch (PDOException $e) {

$errors[] = "Database error: " . $e->getMessage();

}

} elseif (isset($\_POST['unban\_user'])) {

$user\_id = $\_POST['user\_id'];

try {

$pdo = new PDO($dsn, $db\_username, $db\_password, $options);

$stmt = $pdo->prepare("UPDATE users SET banned = 0 WHERE id = :id");

$stmt->execute(['id' => $user\_id]);

$success = "User has been unbanned successfully!";

} catch (PDOException $e) {

$errors[] = "Database error: " . $e->getMessage();

}

}

}

try {

$pdo = new PDO($dsn, $db\_username, $db\_password, $options);

$stmt = $pdo->query("SELECT id, username, email, banned FROM users");

$users = $stmt->fetchAll(PDO::FETCH\_ASSOC);

} catch (PDOException $e) {

$errors[] = "Database error: " . $e->getMessage();

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Admin - Felhasználók Kezelése</title>

<link href="startbootstrap-agency-gh-pages\css\styles.css" rel="stylesheet" type="text/css">

</head>

<body>

<div class="container mt-5">

<h2>Felhasználók Kezelése</h2>

<?php if (!empty($errors)): ?>

<div class="alert alert-danger">

<?php foreach ($errors as $error): ?>

<p><?php echo $error; ?></p>

<?php endforeach; ?>

</div>

<?php endif; ?>

<?php if (!empty($success)): ?>

<div class="alert alert-success">

<p><?php echo $success; ?></p>

</div>

<?php endif; ?>

<table class="table table-bordered">

<thead>

<tr>

<th>ID</th>

<th>Felhasználónév</th>

<th>Email</th>

<th>Státusz</th>

<th>Ban/Unban</th>

</tr>

</thead>

<tbody>

<?php foreach ($users as $user): ?>

<tr>

<td><?php echo $user['id']; ?></td>

<td><?php echo htmlspecialchars($user['username']); ?></td>

<td><?php echo htmlspecialchars($user['email']); ?></td>

<td><?php echo $user['banned'] ? 'Banned' : 'Active'; ?></td>

<td>

<?php if (!$user['banned']): ?>

<form method="POST" style="display:inline;">

<input type="hidden" name="user\_id" value="<?php echo $user['id']; ?>">

<button type="submit" name="ban\_user" class="btn btn-danger" onclick="return confirm('Are you sure you want to ban this user?');">Ban</button>

</form>

<?php else: ?>

<form method="POST" style="display:inline;">

<input type="hidden" name="user\_id" value="<?php echo $user['id']; ?>">

<button type="submit" name="unban\_user" class="btn btn-success" onclick="return confirm('Are you sure you want to unban this user?');">Unban</button>

</form>

<?php endif; ?>

</td>

</tr>

<?php endforeach; ?>

</tbody>

</table>

<a href="admin\_categories.php" class="btn btn-secondary mt-3">Vissza a Dashboard-ra</a>

</div>

</body>

</html>  
  
------------------------------------------------------------------  
  
admin\_categories.php:  
  
<?php

require 'db\_config.php';

session\_start();

if (!isset($\_SESSION['role']) || $\_SESSION['role'] !== 'admin') {

header("Location: admin\_login.php");

exit();

}

$errors = [];

$success = '';

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

if (isset($\_POST['add\_category'])) {

$category\_name = trim($\_POST['category\_name']);

if (empty($category\_name)) {

$errors[] = "The category name cannot be empty.";

} else {

try {

$pdo = new PDO($dsn, $db\_username, $db\_password, $options);

$stmt = $pdo->prepare("INSERT INTO job\_categories (category\_name) VALUES (:category\_name)");

$stmt->execute(['category\_name' => $category\_name]);

$success = "Category added successfully!";

} catch (PDOException $e) {

$errors[] = "Database error: " . $e->getMessage();

}

}

} elseif (isset($\_POST['edit\_category'])) {

$category\_id = $\_POST['category\_id'];

$category\_name = trim($\_POST['category\_name']);

if (empty($category\_name)) {

$errors[] = "A kategória név mezője nem lehet üres";

} else {

try {

$pdo = new PDO($dsn, $db\_username, $db\_password, $options);

$stmt = $pdo->prepare("UPDATE job\_categories SET category\_name = :category\_name WHERE id = :id");

$stmt->execute(['category\_name' => $category\_name, 'id' => $category\_id]);

$success = "Sikeres kategória változtatás!";

} catch (PDOException $e) {

$errors[] = "Database error: " . $e->getMessage();

}

}

} elseif (isset($\_POST['delete\_category'])) {

$category\_id = $\_POST['category\_id'];

try {

$pdo = new PDO($dsn, $db\_username, $db\_password, $options);

$stmt = $pdo->prepare("DELETE FROM job\_categories WHERE id = :id");

$stmt->execute(['id' => $category\_id]);

$success = "Sikeres kategória kitörlés!";

} catch (PDOException $e) {

$errors[] = "Database error: " . $e->getMessage();

}

}

}

try {

$pdo = new PDO($dsn, $db\_username, $db\_password, $options);

$stmt = $pdo->query("SELECT \* FROM job\_categories");

$categories = $stmt->fetchAll(PDO::FETCH\_ASSOC);

} catch (PDOException $e) {

$errors[] = "Database error: " . $e->getMessage();

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Admin - Munka Kategóriák Kezelése</title>

<link href="startbootstrap-agency-gh-pages\css\styles.css" rel="stylesheet" type="text/css">

</head>

<body>

<div class="container mt-5">

<h2>Munka Kategóriák Kezelése</h2>

<?php if (!empty($errors)): ?>

<div class="alert alert-danger">

<?php foreach ($errors as $error): ?>

<p><?php echo $error; ?></p>

<?php endforeach; ?>

</div>

<?php endif; ?>

<?php if (!empty($success)): ?>

<div class="alert alert-success">

<p><?php echo $success; ?></p>

</div>

<?php endif; ?>

<form method="POST">

<div class="mb-3">

<label for="category\_name" class="form-label">Kategória Neve</label>

<input type="text" class="form-control" id="category\_name" name="category\_name" required>

</div>

<button type="submit" name="add\_category" class="btn btn-primary">Kategória Hozzáadása</button>

</form>

<h3 class="mt-5">Kategóriák</h3>

<table class="table table-bordered">

<thead>

<tr>

<th>ID</th>

<th>Kategória Neve</th>

<th>Műveletek</th>

</tr>

</thead>

<tbody>

<?php foreach ($categories as $category): ?>

<tr>

<td><?php echo $category['id']; ?></td>

<td><?php echo htmlspecialchars($category['category\_name']); ?></td>

<td>

<form method="POST" style="display:inline;">

<input type="hidden" name="category\_id" value="<?php echo $category['id']; ?>">

<input type="text" name="category\_name" value="<?php echo htmlspecialchars($category['category\_name']); ?>" required>

<button type="submit" name="edit\_category" class="btn btn-warning">Edit</button>

</form>

<form method="POST" style="display:inline;">

<input type="hidden" name="category\_id" value="<?php echo $category['id']; ?>">

<button type="submit" name="delete\_category" class="btn btn-danger" onclick="return confirm('Biztos ki akarja törölni ezt a kategóriát?');">Törlés</button>

</form>

</td>

</tr>

<?php endforeach; ?>

</tbody>

</table>

<a href="admin\_job\_approval.php" class="btn btn-secondary mt-3">Állások Jóváhagyása</a>

<a href="job\_listings.php" class="btn btn-primary mt-3">Állásajánlatok megtekintése</a>

<a href="manage\_users.php" class="btn btn-primary mt-3">Manage Users</a>

<a href="logout.php" class="btn btn-danger mt-3">Kijelentkezés</a>

</div>

</body>

</html>  
  
-----------------------------------------------------  
  
message\_form.php:  
  
<?php

session\_start();

require 'db\_config.php';

if (!isset($\_SESSION['user\_id'])) {

header("Location: login.php");

exit;

}

$errors = [];

$success = '';

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$receiver\_id = intval($\_POST['receiver\_id']);

$message\_text = trim($\_POST['message\_text']);

$sender\_id = $\_SESSION['user\_id'];

if (empty($message\_text)) {

$errors[] = "Message cannot be empty.";

}

if (empty($errors)) {

try {

$pdo = new PDO($dsn, $db\_username, $db\_password, $options);

$stmt = $pdo->prepare("INSERT INTO messages (sender\_id, receiver\_id, message\_text) VALUES (:sender\_id, :receiver\_id, :message\_text)");

$stmt->execute([

'sender\_id' => $sender\_id,

'receiver\_id' => $receiver\_id,

'message\_text' => $message\_text

]);

$success = "Message sent successfully.";

} catch (PDOException $e) {

$errors[] = "Database error: " . $e->getMessage();

}

}

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link href="startbootstrap-agency-gh-pages\css\styles.css" rel="stylesheet" type="text/css">

<title>Üzenet Küldés</title>

</head>

<body>

<div class="container mt-5">

<h1>Üzenet Küldés</h1>

<?php if (!empty($errors)): ?>

<div class="alert alert-danger">

<?php echo implode('<br>', $errors); ?>

</div>

<?php endif; ?>

<?php if ($success): ?>

<div class="alert alert-success">

<?php echo $success; ?>

</div>

<?php endif; ?>

<form action="send\_message.php" method="POST">

<div class="mb-3">

<label for="receiver\_id" class="form-label">Recipiens</label>

<select name="receiver\_id" id="receiver\_id" class="form-control" required>

<?php

$stmt = $pdo->query("SELECT id, username FROM users WHERE id != {$\_SESSION['user\_id']}");

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

echo "<option value='{$row['id']}'>{$row['username']}</option>";

}

?>

</select>

</div>

<div class="mb-3">

<label for="message\_text" class="form-label">Üzenet</label>

<textarea name="message\_text" id="message\_text" class="form-control" required></textarea>

</div>

<button type="submit" class="btn btn-primary">Küldés</button>

</form>

</div>

</body>

</html>  
  
-----------------------------------------------------  
  
job.sql:  
  
CREATE DATABASE at;

USE at;

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

email VARCHAR(255) UNIQUE NOT NULL,

username VARCHAR(255) UNIQUE NOT NULL,

password\_hash VARCHAR(255) NOT NULL,

is\_active BOOLEAN DEFAULT FALSE,

activation\_token VARCHAR(255),

role ENUM('user', 'company') NOT NULL,

company\_name VARCHAR(255),

company\_website VARCHAR(255),

company\_address TEXT,

company\_description TEXT

);

CREATE TABLE jobs (

id INT AUTO\_INCREMENT PRIMARY KEY,

company\_id INT,

job\_title VARCHAR(255) NOT NULL,

job\_description TEXT NOT NULL,

category VARCHAR(255),

location VARCHAR(255),

application\_deadline DATE,

is\_active BOOLEAN DEFAULT FALSE,

is\_expired BOOLEAN DEFAULT FALSE,,

company\_name VARCHAR(255),

FOREIGN KEY (company\_id) REFERENCES users(id)

);

CREATE TABLE job\_applications (

id INT AUTO\_INCREMENT PRIMARY KEY,

job\_id INT,

user\_id INT,

application\_date DATETIME DEFAULT CURRENT\_TIMESTAMP,

status ENUM('pending', 'approved', 'rejected') DEFAULT 'pending',

FOREIGN KEY (job\_id) REFERENCES jobs(id),

FOREIGN KEY (user\_id) REFERENCES users(id)

);

CREATE TABLE messages (

id INT AUTO\_INCREMENT PRIMARY KEY,

sender\_id INT NOT NULL,

receiver\_id INT NOT NULL,

message\_text TEXT NOT NULL,

timestamp DATETIME DEFAULT CURRENT\_TIMESTAMP,

is\_read BOOLEAN DEFAULT FALSE,

FOREIGN KEY (sender\_id) REFERENCES users(id) ON DELETE CASCADE,

FOREIGN KEY (receiver\_id) REFERENCES users(id) ON DELETE CASCADE

);

CREATE TABLE job\_categories (

id INT AUTO\_INCREMENT PRIMARY KEY,

category\_name VARCHAR(255) UNIQUE NOT NULL

);

CREATE TABLE admin\_users (

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(255) UNIQUE NOT NULL,

password\_hash VARCHAR(255) NOT NULL,

role VARCHAR(50) DEFAULT 'admin'

);

CREATE TABLE job\_posts (

id INT AUTO\_INCREMENT PRIMARY KEY,

position VARCHAR(255) NOT NULL,

description TEXT NOT NULL,

location VARCHAR(255) NOT NULL,

deadline DATE NOT NULL,

company\_name VARCHAR(255) NOT NULL,

is\_approved TINYINT(1) DEFAULT 0,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE password\_resets (

id INT AUTO\_INCREMENT PRIMARY KEY,

email VARCHAR(255) NOT NULL,

token VARCHAR(64) NOT NULL,

expires DATETIME NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

ALTER TABLE users ADD COLUMN banned TINYINT(1) DEFAULT 0;

INSERT INTO admin\_users (username, password\_hash, role) VALUES ('AT', 'sandor3fiam', 'admin');

ALTER TABLE job\_posts ADD COLUMN is\_approved TINYINT(1) DEFAULT 0;

ALTER TABLE job\_posts ADD COLUMN is\_active TINYINT(1) DEFAULT 1;

SELECT \* FROM job\_posts WHERE is\_approved = 0;

SELECT \* FROM jobs WHERE is\_active = 1 AND is\_expired = 0;  
  
------------------------------------------------------------  
  
main.js:  
  
document.getElementById('registrationForm').addEventListener('submit', function(event) {

var errors = [];

var username = document.getElementById('username').value;

if (!/^[a-zA-Z0-9\_]{3,16}$/.test(username)) {

errors.push("A felhasználónév 3-16 betű hosszú kell, hogy legyen, és lehetnek benne betűk, és számok");

}

var email = document.getElementById('email').value;

if (!/^[^@\s]+@[^@\s]+\.[^@\s]+$/.test(email)) {

errors.push("Helytelen email formátum");

}

var password = document.getElementById('password').value;

var confirmPassword = document.getElementById('confirm\_password').value;

if (password.length < 8) {

errors.push("A jelszó legalább 8 betű hosszúságú kell, hogy legyen");

}

if (password !== confirmPassword) {

errors.push("Jelszavak nem egyeznek");

}

if (errors.length > 0) {

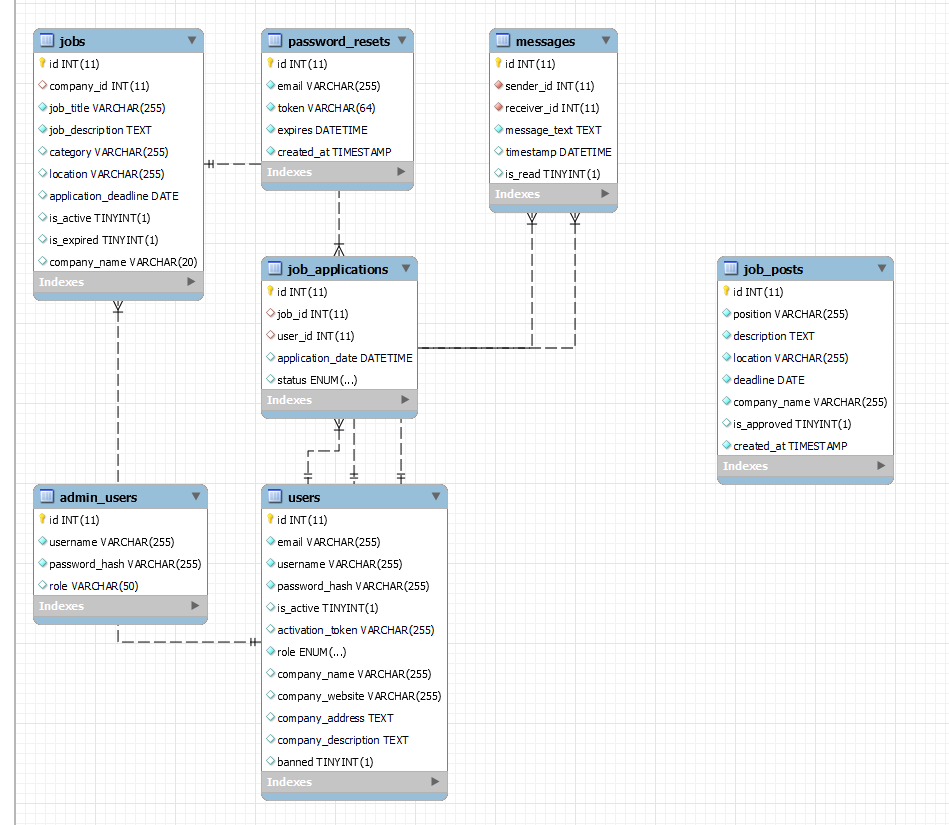
event.preventDefault();

alert(errors.join("\n"));

}

});  
  
  
---------------------------------------------

# Az adatbázis struktúrája (MySQL Workbench)

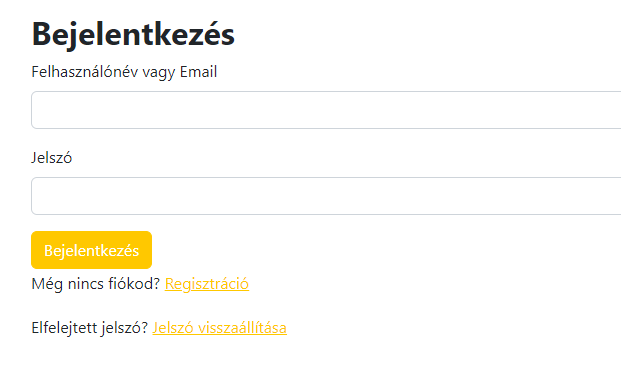
****

# A projekt működésének leírása

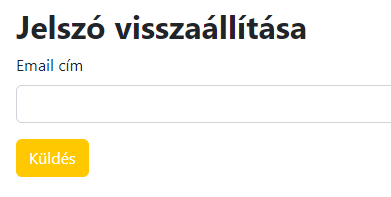
Először is be kell regisztrálni. Azt az Urlap.php oldalon lehet megtenni. Miután a felhasználó kitölti az összes szükséges mezőt, a rendszer küld neki egy email-t, ami tartalmaz egy linket. A felhasználó a link rákattintásával fejezi be a regisztrációt.



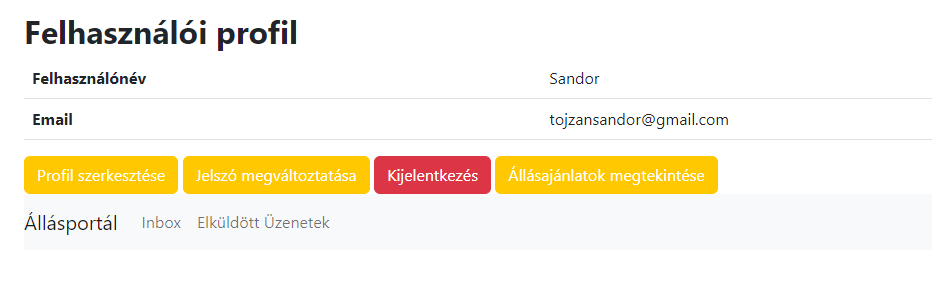
Ha már bevan egy email cím regisztrálva, az már nem bír új account-ot csinálni, úgyhogy a login.php-n kereszül kell visszamennie az oldalra. Az admin-nek van külön login felülete.

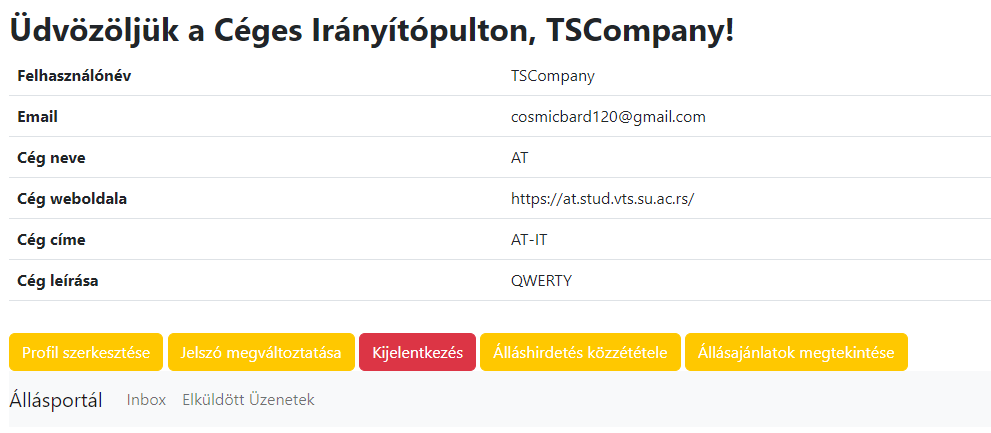
****

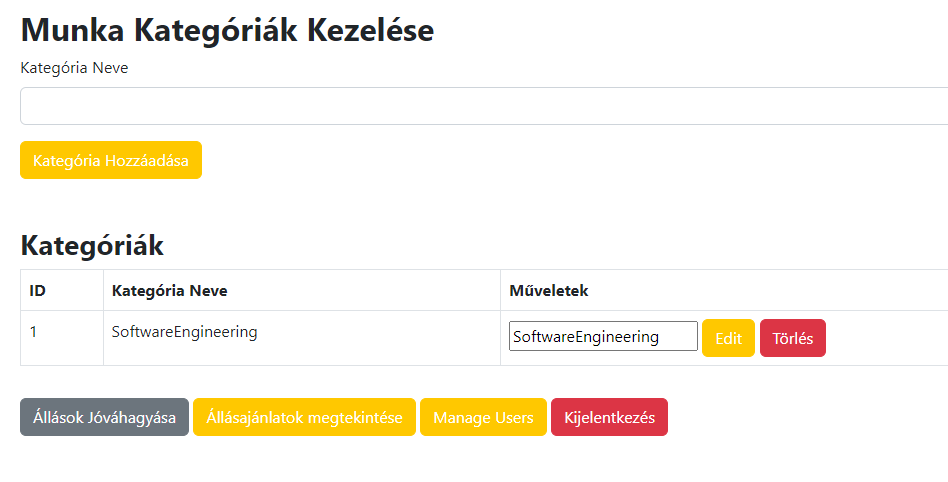
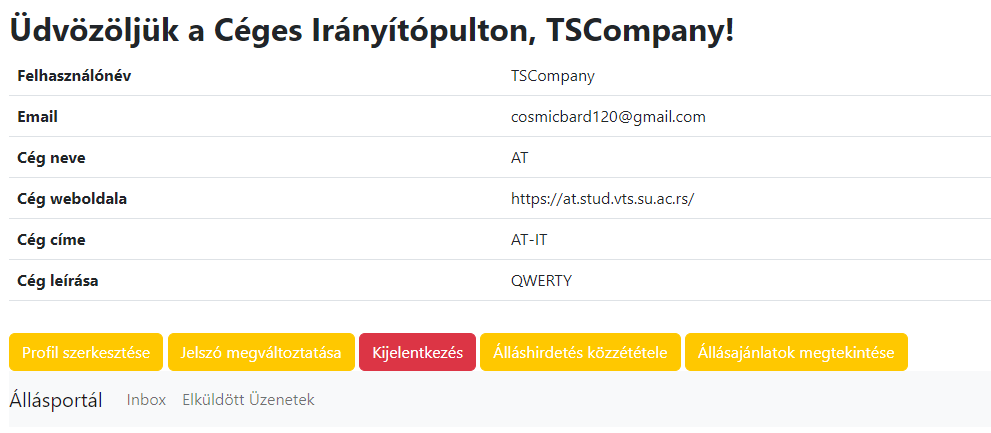
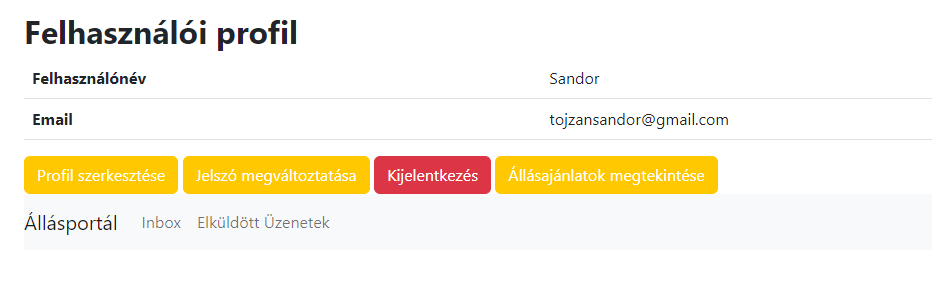
****

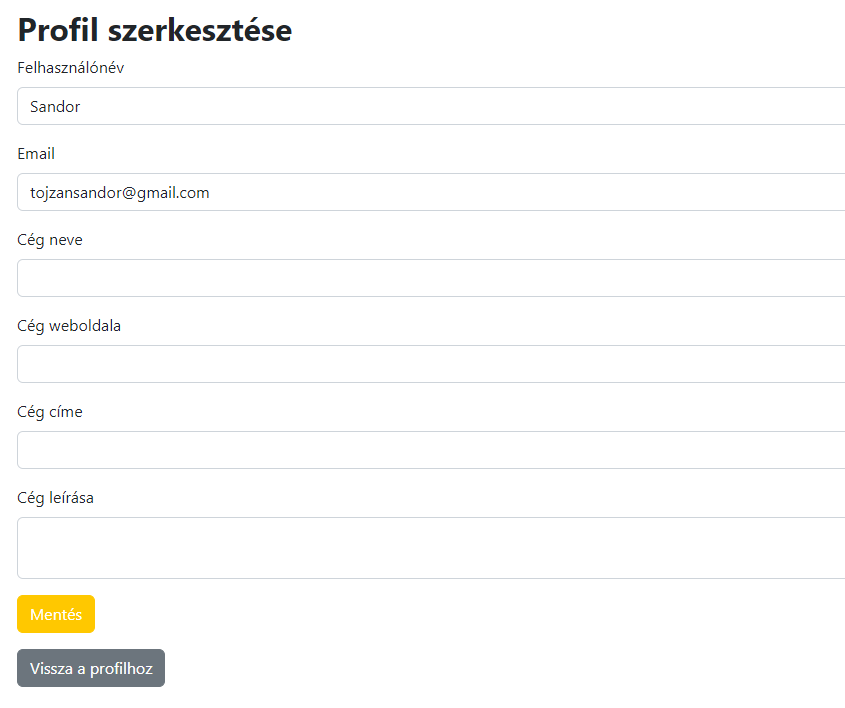
Ha netán egy felhasználó elfelejtené a jelszavát, van lehetősége újat kérni. A rendszer azt a folyamatot is egy email-en keresztül intézi el.

Mikor a felhasználó bejelentkezik, a rendszer a fő felületre küldi őtet. Ez a felület más a cégeknek, az admin-nek, és a reguláris felhasználóknak.



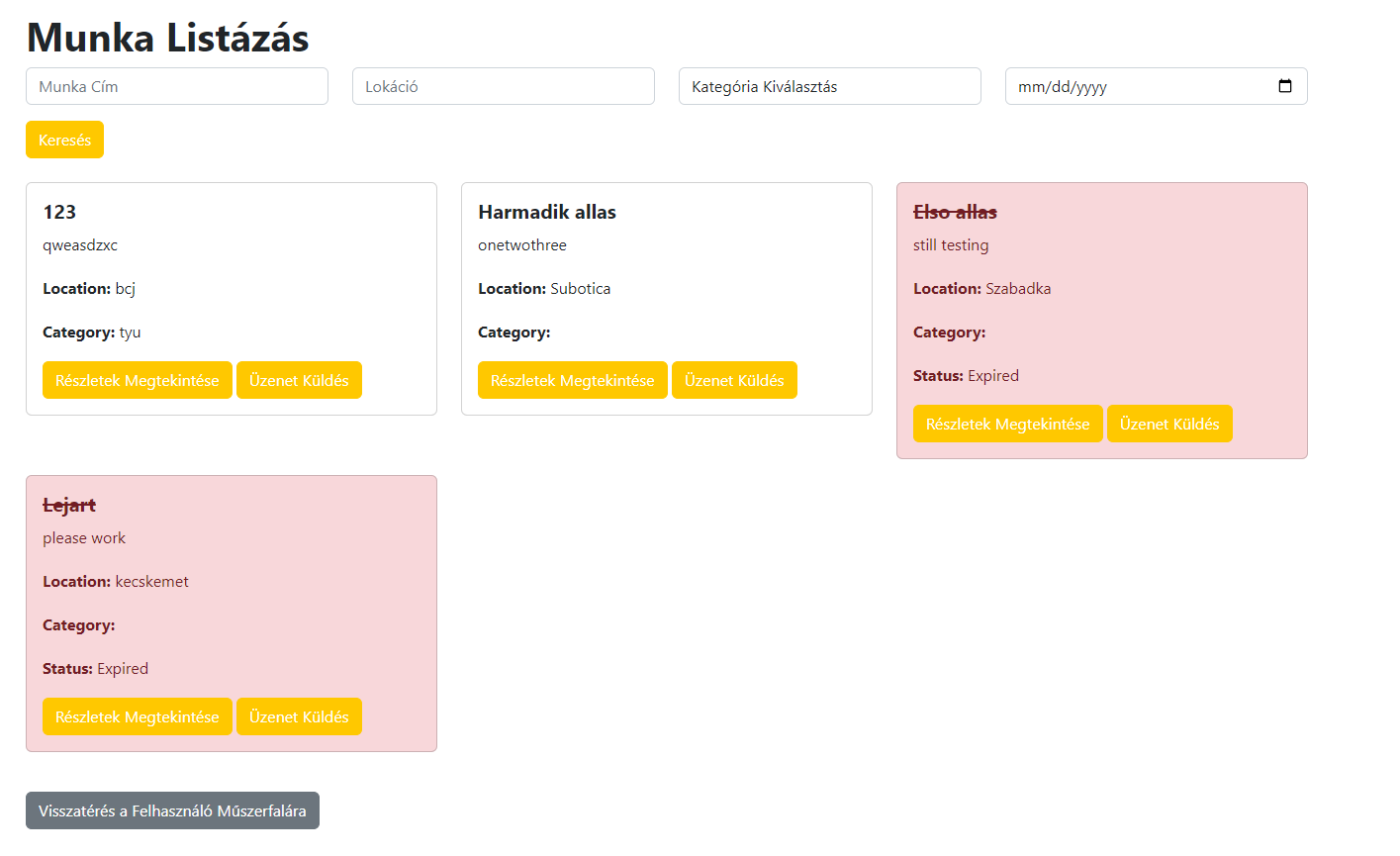


Felhasználóként jó pár elérhető funkció van. Külön megemlítendő a profil szerkesztés, és a jelszó változtatás.

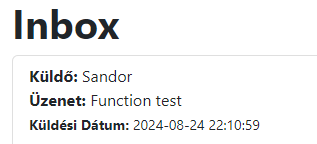




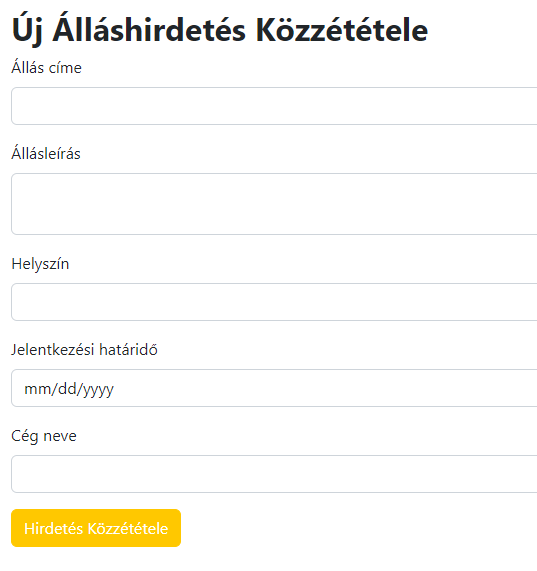
A weboldal fő funkciója viszont az állások átnézése. Erre van egy nagy felület, amihez mindenki hozzáférhet. Itt látszódik, hogy mi minden rett felrakva, és hogy közülük melyikeken nem járt még le a határidő.

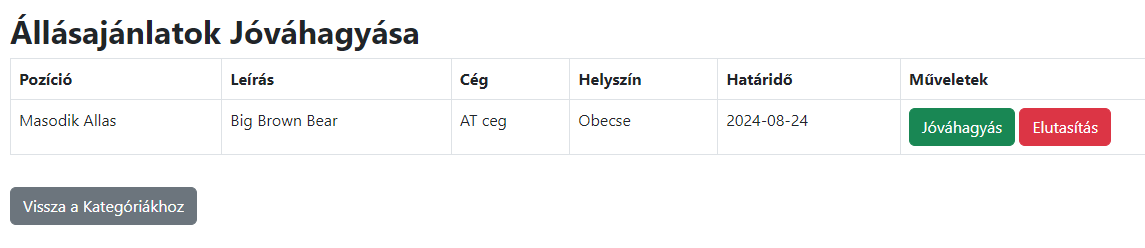
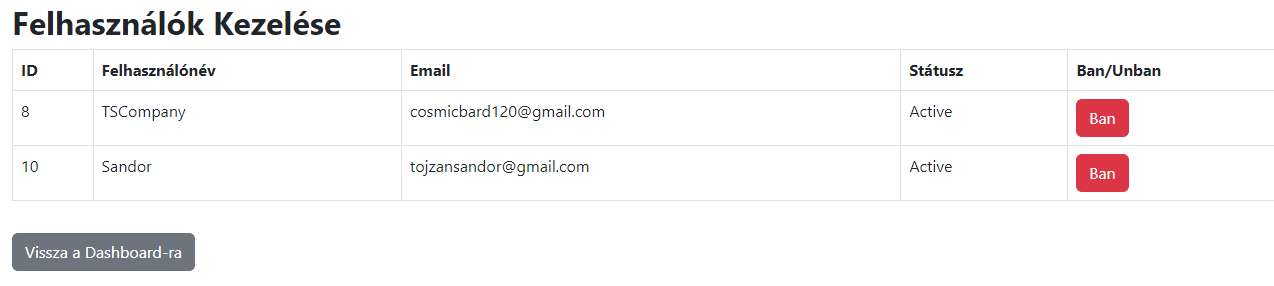
Az individuális munkáknak meglehet tekinteni a régszleteiket, és ha a felhasználónak tetszik, kontaktálhatja a céget, ami felrakta. Minden felhasználónak (cégnek, és regulárisnak is) vannak külön felületeik az inbox-ra, é az elküldött üzeneteikre, ahol követketik a beszélgetéseiket.





A céges felhasználóknak van egy külön oldaljuk, amivel felrakhatják a munkalehetőségeket.



Ezeket a munkalehetőségeket pedig az admin-oknak kell jóváhagyniuk vagy elutasítaniuk, amit ők az egyik saját oldalukon megtehetik.    
  
Viszont az admin-ok nem csak az állásokkal bírnak foglalkozni. A gondot okozó felhasználók letiltására is van egy külön felületük.

# Felhasznált irodalom

1. https://startbootstrap.com/theme/agency
2. https://mailtrap.io/blog/setup-smtp-server/
3. <https://www.geeksforgeeks.org/how-to-send-an-email-using-phpmailer/>
4. Több https://stackoverflow.com/-ra küldött válasz.

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

W2. [http://www.jquery.com](http://www.jquery.com/)