

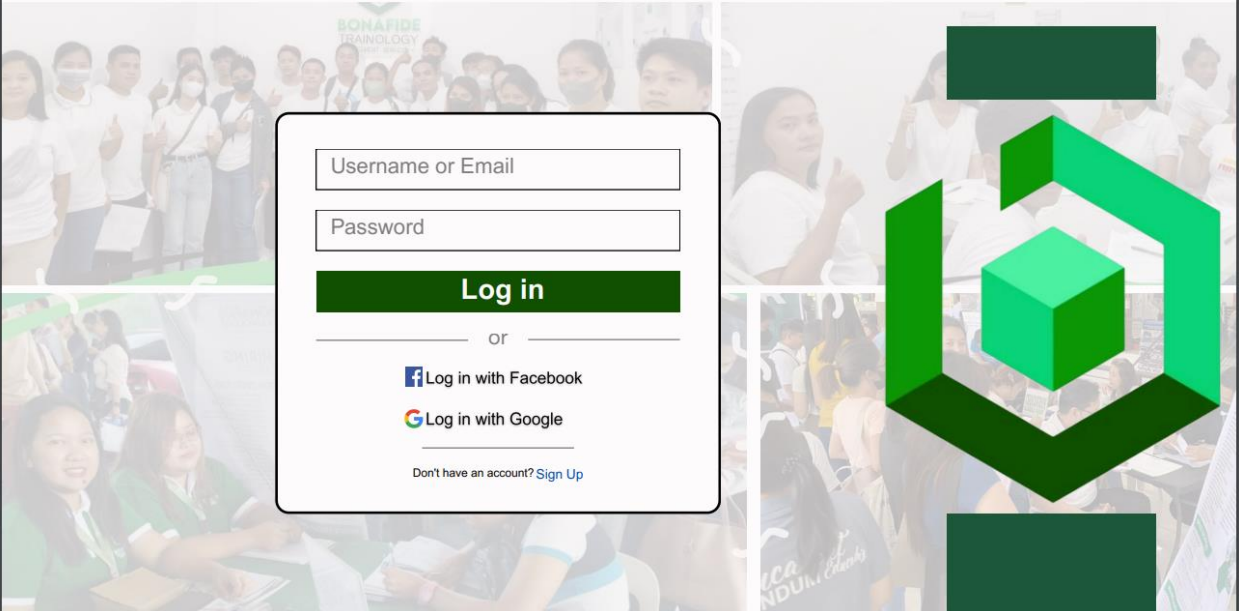





Make a ui design for all of this codes with a responsive features. Based the ui design in the given pictures.

**Bonafide Trainology Placement Services**


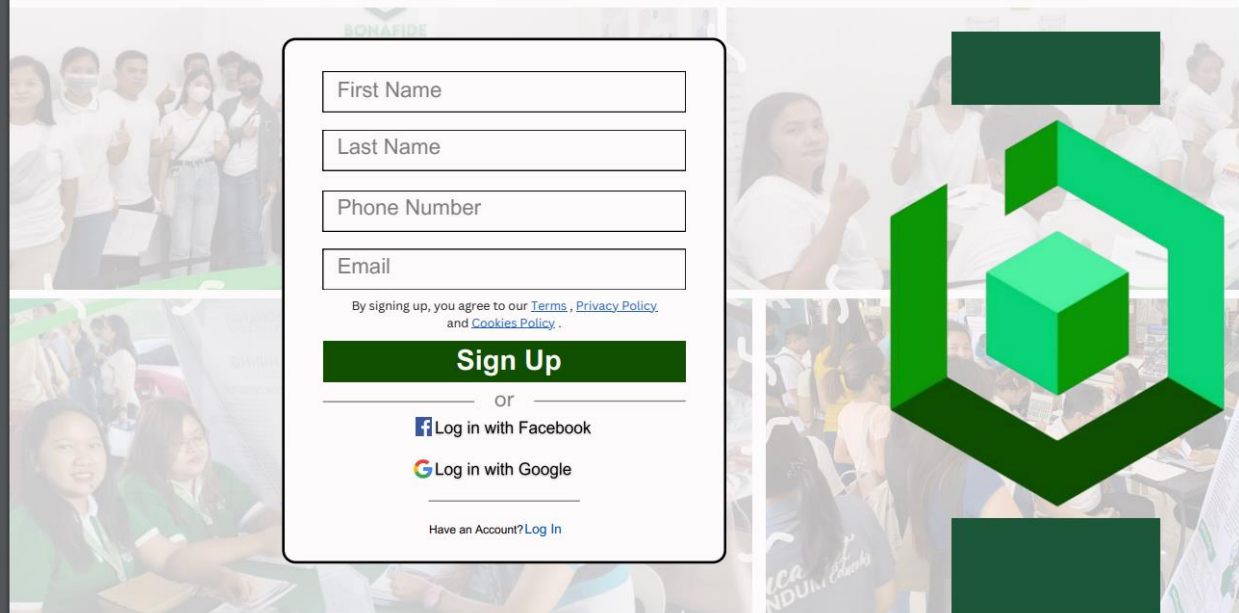
Home About Contact Services





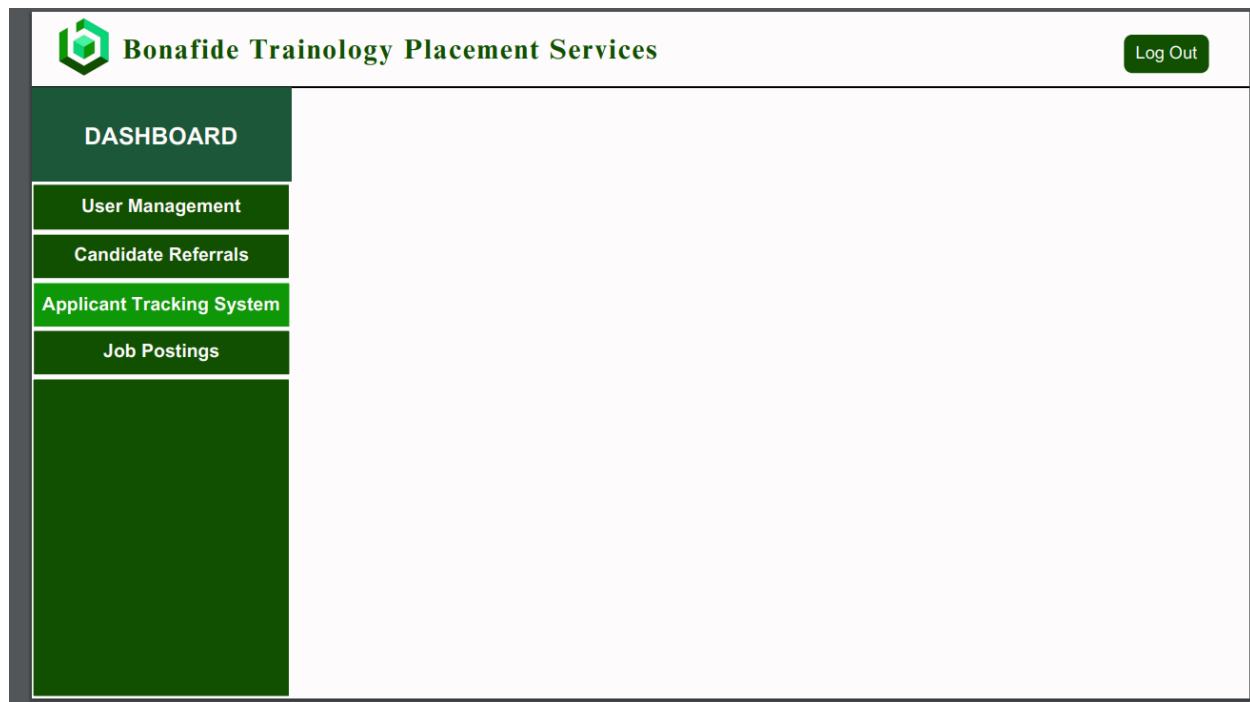
Log in
or
 Log in with Facebook
 Log in with Google
[Don't have an account? Sign Up](#)

**Bonafide Trainology Placement Services**

Home About Contact Services



By signing up, you agree to our [Terms](#), [Privacy Policy](#), and [Cookies Policy](#).
Sign Up
or
 Log in with Facebook
 Log in with Google
[Have an Account? Log In](#)



application.php

```
<?php
```

```
// Include database connection and start session
```

```
include '../db.php'; // Adjust this path based on your directory structure
```

```
// Check if the user is logged in as an applicant
```

```
if (!isset($_SESSION['user_id']) || $_SESSION['role'] != 'APPLICANT') {
```

```
    header('Location: login.php');
```

```
    exit();
```

```
}
```

```
include 'header.php';
```

```
$user_id = $_SESSION['user_id'];
```

```
// Fetch all active applications submitted by the logged-in applicant
```

```
$applications_sql = "
```

```
    SELECT a.application_id, a.application_status, a.rejection_reason, a.time_applied,  
           j.job_title, j.company, j.location,  
           i.interview_date, i.interview_type, i.meet_link,  
           a.job_id
```

```
FROM applications a
```

```
JOIN job_postings j ON a.job_id = j.job_id
```

```
LEFT JOIN tbl_interview i ON a.application_id = i.application_id
```

```
WHERE a.profile_id = (SELECT profile_id FROM profiles WHERE user_id = $user_id)
```

```
AND a.application_status != 'WITHDRAWN'
```

```
ORDER BY a.time_applied DESC
```

```
";
```

```
$applications_result = $conn->query($applications_sql);
```

```
if (!$applications_result) {
```

```
    die("Error fetching applications: " . $conn->error);
```

```
}
```

```
// Handle accept or withdraw offer actions
```

```
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
```

```
    $application_id = $_POST['application_id'];
```

```
    $job_id = $_POST['job_id'];
```

```
    if (isset($_POST['accept_offer'])) {
```

```
        // Accept the offer, update application status to 'ACCEPTED'
```

```
        $accept_sql = "UPDATE applications SET application_status = 'ACCEPTED' WHERE  
application_id = ?";
```

```

$stmt = $conn->prepare($accept_sql);
$stmt->bind_param('i', $application_id);
if ($stmt->execute()) {
    echo "Offer accepted!";
} else {
    echo "Error accepting offer: " . $conn->error;
}
}

```

```

if (isset($_POST['withdraw_offer'])) {
    // Withdraw the offer, update application status to 'WITHDRAWN'

    $withdraw_sql = "UPDATE applications SET application_status = 'WITHDRAWN', withdrawn_at =
NOW() WHERE application_id = ?";

    $stmt = $conn->prepare($withdraw_sql);
    $stmt->bind_param('i', $application_id);
    if ($stmt->execute()) {
        // Update tbl_pipeline_stage

        $pipeline_sql = "UPDATE tbl_pipeline_stage SET withdrawn_at = NOW() WHERE
application_id = ?";

        $pipeline_stmt = $conn->prepare($pipeline_sql);
        $pipeline_stmt->bind_param('i', $application_id);
        $pipeline_stmt->execute();

        // Update tbl_job_metrics

        $metrics_sql = "UPDATE tbl_job_metrics SET withdrawn_applicants = withdrawn_applicants +
1 WHERE job_id = ?";

        $metrics_stmt = $conn->prepare($metrics_sql);
        $metrics_stmt->bind_param('i', $job_id);
        $metrics_stmt->execute();
    }
}

```

```
        echo "Offer withdrawn!";
    } else {
        echo "Error withdrawing offer: " . $conn->error;
    }
}
}
?>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My Applications</title>
    <style>
        table {
            width: 100%;
            border-collapse: collapse;
        }

        table, th, td {
            border: 1px solid black;
        }

        th, td {
            padding: 10px;
            text-align: left;
        }
```

```
th {  
    background-color: #f2f2f2;  
}
```

```
form {  
    margin: 0;  
}
```

```
button {  
    padding: 5px 10px;  
    background-color: #007bff;  
    color: white;  
    border: none;  
    cursor: pointer;  
}
```

```
button:hover {  
    background-color: #0056b3;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>My Applications</h1>
```

```
<table>
```

```
<thead>
```

```
<tr>
```

```
<th>Job Title</th>
```

```
<th>Company</th>
```

```
<th>Location</th>
```

```

        <th>Status</th>

        <th>Date Applied</th>

        <th>Details</th>

        <th>Actions</th>

    </tr>
</thead>
<tbody>
    <?php if ($applications_result->num_rows > 0): ?>
        <?php while ($application = $applications_result->fetch_assoc()): ?>
            <tr>

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

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

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

                <td>

                    <?php
                        // Display status based on application_status field
                        switch ($application['application_status']) {
                            case 'APPLIED':
                                echo 'Application Submitted';
                                break;
                            case 'SCREENING':
                                echo 'Screening';
                                break;
                            case 'INTERVIEW':
                                echo 'Interview Scheduled';
                                break;
                            case 'OFFERED':
                                echo 'Offer Made';
                                break;
                        }
                    </?php
                </td>
            </tr>
        </?php while ($application = $applications_result->fetch_assoc()): ?>
    </?php if ($applications_result->num_rows > 0): ?>

```

```

        case 'REJECTED':
            echo 'Rejected';

            break;

        default:
            echo 'Unknown Status';
    }
    ?>
</td>

<td><?php echo date('F d, Y', strtotime($application['time_applied'])); ?></td>

<td>

    <?php if ($application['application_status'] == 'INTERVIEW'): ?>

        <!-- Display interview details if scheduled -->

        <p><strong>Interview Date:</strong> <?php echo date('F d, Y h:i A',
strtotime($application['interview_date'])); ?></p>

        <p><strong>Interview Type:</strong> <?php echo
htmlspecialchars($application['interview_type']); ?></p>

        <p><strong>Meeting Link:</strong> <a href="<?php echo
htmlspecialchars($application['meet_link']); ?>" target="_blank">Join Interview</a></p>

    <?php elseif ($application['application_status'] == 'REJECTED'): ?>

        <!-- Display rejection reason if rejected -->

        <p><strong>Rejection Reason:</strong> <?php echo
htmlspecialchars($application['rejection_reason']); ?></p>

    <?php elseif ($application['application_status'] == 'OFFERED'): ?>

        <!-- Fetch and display offer details -->

        <?php

            $offer_sql = "SELECT salary, start_date, benefits, remarks FROM tbl_offer_details
WHERE job_id = ?";

            $offer_stmt = $conn->prepare($offer_sql);

            $offer_stmt->bind_param('i', $application['job_id']);

            $offer_stmt->execute();

```



```

        $offer_result = $offer_stmt->get_result();

        if ($offer_result && $offer_result->num_rows > 0):
            $offer = $offer_result->fetch_assoc();
            ?>
            <p><strong>Salary:</strong> <?php echo htmlspecialchars($offer['salary']);
?></p>

            <p><strong>Start Date:</strong> <?php echo
htmlspecialchars($offer['start_date']); ?></p>

            <p><strong>Benefits:</strong> <?php echo htmlspecialchars($offer['benefits']);
?></p>

            <p><strong>Remarks:</strong> <?php echo htmlspecialchars($offer['remarks']);
?></p>

            <!-- Accept or Withdraw Offer -->

            <form method="POST" action="">

                <input type="hidden" name="application_id" value="<?php echo
$application['application_id']; ?>">

                <input type="hidden" name="job_id" value="<?php echo $application['job_id'];
?>">

                <button type="submit" name="accept_offer">Accept Offer</button>

                <button type="submit" name="withdraw_offer">Withdraw Offer</button>

            </form>

            <?php else: ?>

                <p>No offer details available.</p>

            <?php endif; ?>

        <?php else: ?>

            <p><strong>Status:</strong> In progress</p>

        <?php endif; ?>

    </td>

</tr>

<?php endwhile; ?>

```

```
<?php else: ?>

<tr>

    <td colspan="7">You have not submitted any applications yet.</td>

</tr>

<?php endif; ?>

</tbody>

</table>

</body>

</html>
```

```
<?php
// Close the database connection
$conn->close();
?>
```

apply_job.php:

```
<?php
include '../db.php';

// Check if the user is logged in as an applicant
if (!isset($_SESSION['user_id']) || $_SESSION['role'] != 'APPLICANT') {
    header('Location: login.php');
    exit();
}
```

```
$user_id = $_SESSION['user_id']; // Get the logged-in user's ID
```

```
// Ensure the job_id is provided
```

```
if (!isset($_GET['job_id'])) {  
    echo "Error: Job ID not found.";   
    exit();  
}  
  
$job_id = (int)$_GET['job_id']; // Sanitize job_id  
$_SESSION['job_id'] = $job_id; // Store job_id in session  
  
// Fetch job details to display  
$job_sql = "SELECT * FROM job_postings WHERE job_id = $job_id";  
$job_result = $conn->query($job_sql);  
if ($job_result->num_rows == 0) {  
    echo "Job not found.";   
    exit();  
}  
$job = $job_result->fetch_assoc();  
  
// Fetch the user's profile_id  
$profile_sql = "SELECT profile_id FROM profiles WHERE user_id = '$user_id'";  
$profile_result = $conn->query($profile_sql);  
  
if ($profile_result->num_rows > 0) {  
    $profile_row = $profile_result->fetch_assoc();  
    $profile_id = $profile_row['profile_id'];  
} else {  
    echo "Error: Profile not found.";   
    exit();  
}
```

```
// Check if the user has previously applied for this job (including withdrawn applications)

$existing_application_sql = "SELECT * FROM applications WHERE job_id = '$job_id' AND profile_id
= '$profile_id'";

$existing_application_result = $conn->query($existing_application_sql);

$existing_application = $existing_application_result->fetch_assoc();
```

```
$reapplying = false; // To check if the user is reapplying after withdrawal
```

```
if ($existing_application) {

    if ($existing_application['application_status'] == 'WITHDRAWN') {

        // If the user previously withdrew their application, allow them to re-apply by updating the
        existing application

        $reapplying = true;

    } else {

        // If the user already applied and didn't withdraw, show an error message

        echo "You have already applied for this job. You cannot apply again unless your application is
        withdrawn.";

        exit();

    }

}
```

```
// Fetch the questionnaire for the job (if any)

$question_sql = "SELECT * FROM questionnaire_template WHERE job_id = $job_id";

$question_result = $conn->query($question_sql);
```

```
// Handle form submission
```

```
if ($_SERVER['REQUEST_METHOD'] == 'POST') {

    // File upload handling

    if (isset($_FILES['resume']) && $_FILES['resume']['error'] == 0) {

        $file_tmp = $_FILES['resume']['tmp_name'];
```

```

$file_name = basename($_FILES['resume']['name']);
$upload_dir = 'uploads/';
$resume_path = $upload_dir . $file_name;

// Move the uploaded file to the correct directory
if (move_uploaded_file($file_tmp, $resume_path)) {
    $_SESSION['resume_file'] = $resume_path; // Store resume path in session
} else {
    echo "Failed to upload resume.";
    exit();
}
}

// Get the values of the new fields from the form
$referral_source = $conn->real_escape_string($_POST['referral_source']);
$qualifications = $conn->real_escape_string($_POST['qualifications']);
$skills = $conn->real_escape_string($_POST['skills']);
$work_experience = $conn->real_escape_string($_POST['work_experience']);

// Start a transaction to ensure atomicity of the insert/update operations
$conn->begin_transaction();

try{
    if ($reapplying) {
        // Update the withdrawn application with the new details
        $application_id = $existing_application['application_id'];
        $update_application_sql = "UPDATE applications
            SET application_status = 'APPLIED',
                resume = '{$_SESSION['resume_file']}',

```

```

        time_applied = NOW(),
        referral_source = '$referral_source',
        qualifications = '$qualifications',
        skills = '$skills',
        work_experience = '$work_experience'
    WHERE application_id = '$application_id'";

$conn->query($update_application_sql);

    // Delete any previous answers and insert new ones

    $delete_answers_sql = "DELETE FROM application_answers WHERE application_id =
'$application_id'";

    $conn->query($delete_answers_sql);

} else {

    // Insert into the applications table (new application)

    $insert_application_sql = "INSERT INTO applications (job_id, profile_id, resume,
application_status, time_applied, recruiter_id, referral_source, qualifications, skills,
work_experience)

        VALUES ('$job_id', '$profile_id', '{$_SESSION['resume_file']}', 'APPLIED', NOW(),
(SELECT created_by FROM job_postings WHERE job_id = '$job_id'), '$referral_source',
'$qualifications', '$skills', '$work_experience')";

    $conn->query($insert_application_sql);

    $application_id = $conn->insert_id;

}

// Insert answers into the application_answers table
if (isset($_POST['answers'])) {

    foreach ($_POST['answers'] as $question_id => $answer) {

        $answer = $conn->real_escape_string($answer);

        $insert_answer_sql = "INSERT INTO application_answers (application_id, question_id,
answer_text)

            VALUES ('$application_id', '$question_id', '$answer')";

```

```

        $conn->query($insert_answer_sql);
    }
}

// Update job metrics in tbl_job_metrics
if (!$reapplying) { // Only increase total_applicants for a new application

    $metrics_check_sql = "SELECT * FROM tbl_job_metrics WHERE job_id = '$job_id'";
    $metrics_check_result = $conn->query($metrics_check_sql);

    if ($metrics_check_result->num_rows > 0) {
        // If metrics entry exists, update the total_applicants count

        $update_metrics_sql = "UPDATE tbl_job_metrics SET total_applicants = total_applicants + 1
WHERE job_id = '$job_id'";

    } else {
        // Insert if no entry exists

        $update_metrics_sql = "INSERT INTO tbl_job_metrics (job_id, total_applicants) VALUES
('$job_id', 1)";

    }

    $conn->query($update_metrics_sql);
}

// Check if a pipeline entry already exists for this application

$pipeline_check_sql = "SELECT * FROM tbl_pipeline_stage WHERE application_id =
'$application_id'";

$pipeline_check_result = $conn->query($pipeline_check_sql);

if ($pipeline_check_result->num_rows > 0) {
    // If pipeline entry exists, update the applied_at timestamp

    $update_pipeline_sql = "UPDATE tbl_pipeline_stage SET applied_at = NOW() WHERE
application_id = '$application_id'";

```

```

        $conn->query($update_pipeline_sql);
    } else {
        // Insert into tbl_pipeline_stage (new pipeline entry)

        $insert_pipeline_sql = "INSERT INTO tbl_pipeline_stage (application_id, applied_at) VALUES
('$application_id', NOW())";

        $conn->query($insert_pipeline_sql);
    }

    // Commit the transaction

    $conn->commit();

    // Redirect to view_jobs.php after successful submission

    header('Location: view_job.php');

    exit();

} catch (Exception $e) {
    // Rollback the transaction on error

    $conn->rollback();

    echo "Error processing the application: " . $e->getMessage();
}

exit();
}
?>

<!-- HTML for the job application form -->
<!DOCTYPE html>
<html lang="en">
<head>

```



```
<meta charset="UTF-8">

<title>Apply for Job</title>

</head>

<body>

    <h2>Apply for Job: <?php echo htmlspecialchars($job['job_title']); ?></h2>

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

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

    <p><strong>Description:</strong> <?php echo nl2br(htmlspecialchars($job['description']));
    ?></p>

    <p><strong>Salary Range:</strong> <?php echo htmlspecialchars($job['min_salary']); ?> - <?php
    echo htmlspecialchars($job['max_salary']); ?></p>

    <form action="apply_job.php?job_id=<?php echo $job_id; ?>" method="POST"
    enctype="multipart/form-data">

        <label for="resume">Upload Resume:</label>

        <input type="file" name="resume" id="resume" required><br>

        <label for="referral_source">How did you hear about this job?</label>

        <select name="referral_source" id="referral_source" required>

            <option value="referral_applicants">Employee Referral</option>

            <option value="social_media_applicants">Social Media</option>

            <option value="career_site_applicants">Career Website</option>

        </select><br>

        <label for="qualifications">Qualifications:</label><br>

        <textarea name="qualifications" id="qualifications" rows="4" cols="50"
        required></textarea><br>

        <label for="skills">Skills:</label><br>

        <textarea name="skills" id="skills" rows="4" cols="50" required></textarea><br>
```

```

<label for="work_experience">Work Experience:</label><br>

<textarea name="work_experience" id="work_experience" rows="4" cols="50"
required></textarea><br>

<!-- Display job-specific questions (if any) -->

<?php if ($question_result->num_rows > 0): ?>

    <h3>Job Questionnaire</h3>

    <?php while ($question = $question_result->fetch_assoc()): ?>

        <label><?php echo htmlspecialchars($question['question_text']); ?></label><br>

        <!-- Display input based on question type -->

        <?php if ($question['question_type'] === 'TEXT'): ?>

            <input type="text" name="answers[<?php echo $question['question_id']; ?>]"
required><br>

        <?php elseif ($question['question_type'] === 'YES_NO'): ?>

            <select name="answers[<?php echo $question['question_id']; ?>]" required>

                <option value="YES">Yes</option>

                <option value="NO">No</option>

            </select><br>

        <?php elseif ($question['question_type'] === 'MULTIPLE_CHOICE'): ?>

            <!-- Radio buttons for multiple choice -->

            <?php if (!empty($question['choice_a'])): ?>

                <input type="radio" name="answers[<?php echo $question['question_id']; ?>]"
value="A" required> <?php echo htmlspecialchars($question['choice_a']); ?><br>

            <?php endif; ?>

            <?php if (!empty($question['choice_b'])): ?>

```

```
        <input type="radio" name="answers[<?php echo $question['question_id']; ?>]"
value="B" required> <?php echo htmlspecialchars($question['choice_b']); ?><br>
```

```
    <?php endif; ?>
```

```
    <?php if (!empty($question['choice_c'])): ?>
```

```
        <input type="radio" name="answers[<?php echo $question['question_id']; ?>]"
value="C" required> <?php echo htmlspecialchars($question['choice_c']); ?><br>
```

```
    <?php endif; ?>
```

```
    <?php if (!empty($question['choice_d'])): ?>
```

```
        <input type="radio" name="answers[<?php echo $question['question_id']; ?>]"
value="D" required> <?php echo htmlspecialchars($question['choice_d']); ?><br>
```

```
    <?php endif; ?>
```

```
    <?php endif; ?>
```

```
    <?php endwhile; ?>
```

```
    <?php endif; ?>
```

```
        <button type="submit">Submit Application</button>
```

```
    </form>
```

```
</body>
```

```
</html>
```

```
<?php
```

```
$conn->close();
```

```
?>
```

dashboard.php:

```
<?php
```

```
// dashboard.php
```

```
include '../db.php'; // Database connection
```

```

// Check if user is logged in and is an applicant
if (!isset($_SESSION['user_id']) || $_SESSION['role'] != 'APPLICANT') {
    header('Location: login.php'); // Redirect to login page
    exit();
}

include 'header.php';

// Check if the user has already filled out the profile
$user_id = $_SESSION['user_id'];

$stmt = prepare_and_execute($conn, "SELECT * FROM profiles WHERE user_id = ?", 'i', $user_id);
$result = $stmt->get_result();

if ($result->num_rows == 0) {
    // If no profile exists, redirect to setup profile
    header('Location: setup_profile.php');
    exit();
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Applicant Dashboard</title>
</head>
<body>
    <h2>Welcome, <?php echo $_SESSION['email']; ?>!</h2>
    <p>This is your applicant dashboard.</p>

```

```
<a href="logout.php">Logout</a>
</body>
</html>
```

footer.php:

```
<!-- Footer Section -->
<footer class="main-footer">
    <p>&copy; 2024 Bonafide Trainology Placement Services. All rights reserved.</p>
</footer>

</body>
</html>
```

header.php:

```
<!-- header.php -->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Applicant Dashboard</title>
    <link rel="stylesheet" href="styles.css"> <!-- Link to your CSS file -->
</head>
<body>
    <header>
        <nav>
            <ul>
                <li><a href="view_job.php">Job Postings</a></li>
```

```
<li><a href="referrals.php">Referrals</a></li>

<li><a href="application.php">Applications</a></li>

<li><a href="profile.php">Profile</a></li>

<li><a href="dashboard.php">Dashboard</a></li>

<li><a href="logout.php">Logout</a></li> <!-- Assume there's a logout page -->

</ul>

</nav>

</header>
```

login.php:

```
<?php
// login.php
include '../db.php'; // Database connection

// Check if login form is submitted
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $email = trim($_POST['email']);
    $password = $_POST['password'];

    // Fetch user by email
    $stmt = prepare_and_execute($conn, "SELECT * FROM users WHERE email = ?", 's', $email);
    $result = $stmt->get_result();
    $user = $result->fetch_assoc();

    if ($user && password_verify($password, $user['password'])) {
        // Set session and login the user
        $_SESSION['email'] = $user['email'];
        $_SESSION['user_id'] = $user['user_id'];
    }
}
```

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

// Update last login time

prepare_and_execute($conn, "UPDATE users SET last_login = NOW() WHERE user_id = ?", 'i',
$user['user_id']);

// Redirect based on role

if ($user['role'] == 'APPLICANT') {
    header('Location: dashboard.php');
} else {
    echo "Recruiters are not allowed to login from this page.";
}

exit();
} else {
    $error = "Invalid email or password!";
}
}
?>
```

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Applicant Login</title>

</head>

<body>

    <h2>Applicant Login</h2>

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

        <input type="email" name="email" placeholder="Email" required><br>
```

```
<input type="password" name="password" placeholder="Password" required><br>
<button type="submit">Login</button>

</form>
```

```
<!-- Add the registration link here -->
```

```
<p>Don't have an account? <a href="register.php">Register here</a></p>
```

```
<?php if (isset($error)) { echo "<p style='color:red;'>$error</p>"; } ?>
```

```
</body>
```

```
</html>
```

logout.php:

```
<?php
```

```
// logout.php
```

```
session_start();
```

```
session_unset(); // Unset all session variables
```

```
session_destroy(); // Destroy the session
```

```
header('Location: login.php'); // Redirect to login
```

```
exit();
```

process_application.php:

```
<?php
```

```
session_start();
```

```
include '../db.php';
```

```
if (!isset($_SESSION['email'])) {
```

```
    header('Location: login.php');
```

```
    exit();
```



```
}
```

```
$job_id = $_SESSION['job_id'];
```

```
$user_id = $_SESSION['user_id'];
```

```
// Fetch profile_id for the logged-in user
```

```
$profile_sql = "SELECT profile_id FROM profiles WHERE user_id = '$user_id'";
```

```
$profile_result = $conn->query($profile_sql);
```

```
if ($profile_result->num_rows > 0) {
```

```
    $profile_row = $profile_result->fetch_assoc();
```

```
    $profile_id = $profile_row['profile_id'];
```

```
// Handle file upload
```

```
if (isset($_FILES['resume']) && $_FILES['resume']['error'] === UPLOAD_ERR_OK) {
```

```
    $upload_dir = 'uploads/';
```

```
    $resume_name = basename($_FILES['resume']['name']);
```

```
    $resume_path = $upload_dir . $resume_name;
```

```
// Ensure the uploads directory exists
```

```
if (!file_exists($upload_dir)) {
```

```
    mkdir($upload_dir, 0777, true); // create directory if it doesn't exist
```

```
}
```

```
// Move the uploaded file
```

```
if (move_uploaded_file($_FILES['resume']['tmp_name'], $resume_path)) {
```

```
    $_SESSION['resume_file'] = $resume_path;
```

```
} else {
```

```
    echo "Error uploading the resume.";
```

```

        exit();
    }
} else {
    echo "Error: No resume uploaded or file upload error.";
    exit();
}

// Start the transaction
$conn->begin_transaction();

try {
    // Check for withdrawn applications

    $withdrawn_sql = "SELECT application_id FROM applications WHERE job_id = '$job_id' AND
profile_id = '$profile_id' AND application_status = 'WITHDRAWN'";

    $withdrawn_result = $conn->query($withdrawn_sql);

    if ($withdrawn_result->num_rows > 0) {
        // Re-apply by updating the withdrawn application

        $withdrawn_row = $withdrawn_result->fetch_assoc();

        $application_id = $withdrawn_row['application_id'];

        $update_application_sql = "UPDATE applications
                                SET application_status = 'SCREENING', time_applied = NOW(), resume =
{'$_SESSION['resume_file']}'
                                WHERE application_id = '$application_id'";

        $conn->query($update_application_sql);

        // Update the pipeline stage

        $update_pipeline_sql = "UPDATE tbl_pipeline_stage

```

```

        SET applied_at = NOW(), withdrawn_at = NULL

        WHERE application_id = '$application_id';

$conn->query($update_pipeline_sql);
    } else {

        // Insert a new application

        $insert_application_sql = "INSERT INTO applications (job_id, profile_id, resume,
application_status, time_applied)

            VALUES ('$job_id', '$profile_id', '{$_SESSION['resume_file']}', 'SCREENING',
NOW())";

        $conn->query($insert_application_sql);

        $application_id = $conn->insert_id;

        // Insert into the pipeline

        $insert_pipeline_sql = "INSERT INTO tbl_pipeline_stage (application_id, applied_at) VALUES
('$application_id', NOW())";

        $conn->query($insert_pipeline_sql);
    }

    // Insert questionnaire answers if provided
    if (isset($_POST['answers'])) {
        foreach ($_POST['answers'] as $question_id => $answer) {

            $answer = $conn->real_escape_string($answer); // escape special characters in answer

            $insert_answer_sql = "INSERT INTO application_answers (application_id, question_id,
answer_text)

                VALUES ('$application_id', '$question_id', '$answer')";

            $conn->query($insert_answer_sql);
        }
    }

    // Commit the transaction

```

```
$conn->commit();
```

```
// Clear session variables and redirect
```

```
unset($_SESSION['job_id'], $_SESSION['resume_file']);
```

```
header("Location: job_postings.php?success=1");
```

```
exit();
```

```
} catch (Exception $e) {
```

```
    // Rollback the transaction in case of error
```

```
    $conn->rollback();
```

```
    echo "Error processing the application: " . $e->getMessage();
```

```
}
```

```
} else {
```

```
    echo "Error: Profile not found.";
```

```
}
```

```
?>
```

referrals.php:

```
<?php
```

```
// Include database connection and start session
```

```
include '../db.php'; // Adjust this path based on your directory structure
```

```
include 'header.php'; // Include header
```

```
include 'footer.php'; // Include footer
```

```
// Check if the user is logged in
```

```
if (!isset($_SESSION['user_id'])) {
```

```
    header('Location: login.php');
```

```

    exit();
}

// Get the logged-in user's ID
$user_id = $_SESSION['user_id'];

// Fetch the referral code of the logged-in user from the profiles table
$profile_sql = "SELECT referral_code FROM profiles WHERE user_id = ?";
$stmt = $conn->prepare($profile_sql);
if (!$stmt) {
    die("Error preparing profile query: " . $conn->error);
}
$stmt->bind_param('i', $user_id);
$stmt->execute();
$profile_result = $stmt->get_result();
$user_profile = $profile_result->fetch_assoc();

// Calculate total points earned from referrals in the referrals table
$points_sql = "SELECT SUM(points) AS total_points FROM referrals WHERE referrer_user_id = ?";
$stmt = $conn->prepare($points_sql);
if (!$stmt) {
    die("Error preparing points query: " . $conn->error);
}
$stmt->bind_param('i', $user_id);
$stmt->execute();
$points_result = $stmt->get_result();
$points = $points_result->fetch_assoc()['total_points'] ?? 0; // Default to 0 if no points

// Fetch all referrals made by the logged-in user, excluding self-referrals

```

```
$referrals_sql = "
    SELECT r.referral_id, r.referral_code, r.points,
           p_referred.fname AS referred_fname, p_referred.lname AS referred_lname,
           p_referred.phone AS referred_phone, p_referred.linkedin_link, p_referred.facebook_link
    FROM referrals r
    JOIN profiles p_referred ON r.referred_user_id = p_referred.user_id
    WHERE r.referrer_user_id = ? AND r.referred_user_id != r.referrer_user_id
    ORDER BY r.referral_id DESC
";
```

```
// Prepare and execute the referral query
$stmt = $conn->prepare($referrals_sql);
if (!$stmt) {
    die("Error preparing referrals query: " . $conn->error);
}
$stmt->bind_param('i', $user_id);
$stmt->execute();
$referrals_result = $stmt->get_result();
```

```
// Handle errors fetching results
if (!$referrals_result) {
    die("Error fetching referrals: " . $conn->error);
}
```

```
?>
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

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

```
<title>My Referrals</title>
```

```
<style>
```

```
  table {  
    width: 100%;  
    border-collapse: collapse;  
  }
```

```
  table, th, td {  
    border: 1px solid black;  
  }
```

```
  th, td {  
    padding: 10px;  
    text-align: left;  
  }
```

```
  th {  
    background-color: #f2f2f2;  
  }
```

```
  .profile-details {  
    margin-bottom: 20px;  
  }
```

```
  .profile-details h3 {  
    margin-bottom: 5px;  
  }
```

```

</style>
</head>
<body>
    <h1>My Referrals</h1>

    <!-- Display referral code and points outside the table -->
    <div class="profile-details">
        <h3>Your Referral Code: <?php echo htmlspecialchars($user_profile['referral_code']); ?></h3>
        <h3>Points Earned: <?php echo htmlspecialchars($points); ?></h3>
    </div>

    <table>
        <thead>
            <tr>
                <th>Referred Person</th>
                <th>Phone</th>
                <th>LinkedIn</th>
                <th>Facebook</th>
            </tr>
        </thead>
        <tbody>
            <?php if ($referrals_result->num_rows > 0): ?>
                <?php while ($referral = $referrals_result->fetch_assoc()): ?>
                    <tr>
                        <td><?php echo htmlspecialchars($referral['referred_fname'] . ' ' .
$referral['referred_lname']); ?></td>
                        <td><?php echo htmlspecialchars($referral['referred_phone']); ?></td>
                        <td>
                            <?php if (!empty($referral['linkedin_link'])): ?>

```



```

        <a href="<?php echo htmlspecialchars($referral['linkedin_link']); ?>"
target="_blank">LinkedIn</a>

        <?php else: ?>

            No LinkedIn link

        <?php endif; ?>

    </td>

    <td>

        <?php if (!empty($referral['facebook_link'])): ?>

            <a href="<?php echo htmlspecialchars($referral['facebook_link']); ?>"
target="_blank">Facebook</a>

            <?php else: ?>

                No Facebook link

            <?php endif; ?>

        </td>

    </tr>

    <?php endwhile; ?>

    <?php else: ?>

        <tr>

            <td colspan="4">You have not made any referrals yet.</td>

        </tr>

    <?php endif; ?>

</tbody>

</table>

<?php include 'footer.php'; ?>

</body>

</html>

<?php

```

```
// Close the database connection
```

```
$conn->close();
```

```
?>
```

register.php:

```
<?php
```

```
include '../db.php'; // Database connection
```

```
// Check if form is submitted
```

```
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
```

```
    $email = trim($_POST['email']);
```

```
    $password = password_hash($_POST['password'], PASSWORD_BCRYPT); // Secure password hash
```

```
    $confirm_password = $_POST['confirm_password'];
```

```
    $referral_code = trim($_POST['referral_code']); // Referral code input
```

```
// Ensure password confirmation matches
```

```
if ($_POST['password'] !== $confirm_password) {
```

```
    $error = "Passwords do not match!";
```

```
} else {
```

```
    // Check if email already exists
```

```
    $stmt = prepare_and_execute($conn, "SELECT * FROM users WHERE email = ?", 's', $email);
```

```
    $result = $stmt->get_result();
```

```
    if ($result->num_rows > 0) {
```

```
        $error = "Email already registered.";
```

```
    } else {
```

```
        // Insert new applicant user into `users` table
```

```
        $role = 'APPLICANT';
```

```
        $sql = "INSERT INTO users (email, password, role, registration_date) VALUES (?, ?, ?, NOW())";
```

```

$stmt = prepare_and_execute($conn, $sql, 'sss', $email, $password, $role);

// Get the last inserted user_id
$user_id = $conn->insert_id;

// Automatically create a referral entry for the new user for future use
$insert_new_user_referral_sql = "INSERT INTO referrals (referred_user_id, referrer_user_id,
points) VALUES (?, ?, 0)";
prepare_and_execute($conn, $insert_new_user_referral_sql, 'ii', $user_id, $user_id);

// Insert referral information if a referral code is provided
if (!empty($referral_code)) {
    // Get the referrer user ID based on referral code
    $referrer_stmt = prepare_and_execute($conn, "SELECT user_id FROM profiles WHERE
referral_code = ?", 's', $referral_code);
    $referrer_result = $referrer_stmt->get_result();

    if ($referrer_result->num_rows > 0) {
        $referrer_row = $referrer_result->fetch_assoc();
        $referrer_user_id = $referrer_row['user_id'];

        // Check if there is already a record for the referrer in the referrals table
        $check_referral_sql = "SELECT * FROM referrals WHERE referrer_user_id = ? AND
referral_code = ?";
        $check_referral_stmt = prepare_and_execute($conn, $check_referral_sql, 'is',
$referrer_user_id, $referral_code);
        $check_referral_result = $check_referral_stmt->get_result();

        // If no record exists, create one for this specific referral
        if ($check_referral_result->num_rows === 0) {

```

```

        // Insert new referral record for the referrer

        $insert_referral_sql = "INSERT INTO referrals (referred_user_id, referrer_user_id,
referral_code, points) VALUES (?, ?, ?, 1)"; // Start with 1 point for the referrer

        prepare_and_execute($conn, $insert_referral_sql, 'iis', $user_id, $referrer_user_id,
$referral_code);

    } else {

        // Record exists; just increment the points

        $update_points_sql = "UPDATE referrals SET points = points + 1 WHERE referrer_user_id
= ? AND referral_code = ?";

        prepare_and_execute($conn, $update_points_sql, 'is', $referrer_user_id,
$referral_code);

    }

} else {

    $error = "Invalid referral code.";

}

}

// Store the session

$_SESSION['email'] = $email;

$_SESSION['user_id'] = $user_id; // Store user_id for profile setup

$_SESSION['role'] = 'APPLICANT';

// Redirect to the profile setup page

header('Location: setup_profile.php');

exit();

}

}

}

?>

```

```

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Applicant Registration</title>

</head>

<body>

    <h2>Applicant Registration</h2>

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

        <input type="email" name="email" placeholder="Email" required><br>

        <input type="password" name="password" placeholder="Password" required><br>

        <input type="password" name="confirm_password" placeholder="Confirm Password"
required><br>

        <input type="text" name="referral_code" placeholder="Referral Code (if any)"><br> <!-- Referral
code input -->

        <button type="submit">Register</button>

    </form>

    <?php if (isset($error)) { echo "<p style='color:red;'>$error</p>"; } ?>

</body>

</html>

```

setup_profile.php:

```

<?php

// setup_profile.php

include '../db.php'; // Include database connection


// Check if the user is logged in and is an applicant
if (!isset($_SESSION['user_id']) || $_SESSION['role'] != 'APPLICANT') {

    header('Location: login.php'); // Redirect to login if not logged in
}

```

```

    exit();
}

// Handle profile form submission
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $fname = trim($_POST['fname']);
    $lname = trim($_POST['lname']);
    $age = (int)$_POST['age'];
    $phone = trim($_POST['phone']);
    $address = trim($_POST['address']);
    $education_level = $_POST['education_level'];
    $school_graduated = trim($_POST['school_graduated']);
    $year_graduated = trim($_POST['year_graduated']);
    $degree = ($education_level == 'TERTIARY' || $education_level == 'POSTGRADUATE') ?
trim($_POST['degree']) : NULL;
    $linkedin_link = !empty($_POST['linkedin_link']) ? trim($_POST['linkedin_link']) : NULL;
    $facebook_link = !empty($_POST['facebook_link']) ? trim($_POST['facebook_link']) : NULL;
    $civil_status = $_POST['civil_status']; // Civil status selected by the user

    // Generate referral code
    function generateReferralCode($length = 8) {
        return strtoupper(substr(md5(time() . rand()), 0, $length));
    }

    // Auto-generate referral code
    $referral_code = generateReferralCode();

    $user_id = $_SESSION['user_id'];

```

```

// Insert profile details into the `profiles` table

$sql = "INSERT INTO profiles (user_id, fname, lname, age, phone, address, education_level,
school_graduated, year_graduated, degree, linkedin_link, facebook_link, referral_code, civil_status)

VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";

prepare_and_execute($conn, $sql, 'ississsssssss', $user_id, $fname, $lname, $age, $phone,
$address, $education_level, $school_graduated, $year_graduated, $degree, $linkedin_link,
$facebook_link, $referral_code, $civil_status);

// Redirect to dashboard after profile setup

header('Location: dashboard.php');

exit();
}
?>

```

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Setup Profile</title>
    <script>
        // Script to show/hide degree field based on education level
        function toggleDegreeField() {
            var educationLevel = document.getElementById('education_level').value;
            var degreeField = document.getElementById('degree_field');
            if (educationLevel === 'TERTIARY' || educationLevel === 'POSTGRADUATE') {
                degreeField.style.display = 'block'; // Show degree field
            } else {
                degreeField.style.display = 'none'; // Hide degree field
            }
        }
    }

```

```
</script>

</head>

<body>

  <h2>Setup Your Profile</h2>

  <!-- Added form opening tag with method="POST" -->

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

    <!-- Personal Information Section -->

    <fieldset>

      <legend>Personal Information</legend>

      <input type="text" name="fname" placeholder="First Name" required><br>

      <input type="text" name="lname" placeholder="Last Name" required><br>

      <input type="number" name="age" placeholder="Age" required><br>

      <!-- Civil Status Selection -->

      <label for="civil_status">Civil Status:</label>

      <select name="civil_status" id="civil_status" required>

        <option value="SINGLE">Single</option>

        <option value="MARRIED">Married</option>

        <option value="DIVORCED">Divorced</option>

        <option value="WIDOWED">Widowed</option>

      </select><br>

    </fieldset>

    <!-- Contact Information Section -->

    <fieldset>

      <legend>Contact Information</legend>

      <input type="text" name="phone" placeholder="Phone Number" required><br>
```



```
<input type="text" name="address" placeholder="Address" required><br>
<input type="text" name="linkedin_link" placeholder="LinkedIn Profile (Optional)"><br>
<input type="text" name="facebook_link" placeholder="Facebook Profile (Optional)"><br>
</fieldset>

<!-- Education Information Section -->

<fieldset>

  <legend>Education Information</legend>

  <label for="education_level">Education Level:</label>

  <select name="education_level" id="education_level" onchange="toggleDegreeField()"
required>

    <option value="PRIMARY">Primary</option>

    <option value="SECONDARY">Secondary</option>

    <option value="TERTIARY">Tertiary</option>

    <option value="POSTGRADUATE">Postgraduate</option>

  </select><br>

  <input type="text" name="school_graduated" placeholder="School Graduated"
required><br>

  <input type="text" name="year_graduated" placeholder="Year Graduated" required><br>

  <!-- Degree field, hidden initially, shown only for Tertiary or Postgraduate -->

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

    <input type="text" name="degree" placeholder="Degree Obtained"><br>

  </div>

</fieldset>

<!-- Submit Button -->

<button type="submit">Submit Profile</button>
```

```
</form>
</body>
</html>
```

view_job.php:

```
<?php
include '../db.php'; // Database connection

// Ensure the user is logged in as an applicant
if (!isset($_SESSION['user_id']) || $_SESSION['role'] != 'APPLICANT') {
    header('Location: login.php');
    exit();
}

$user_id = $_SESSION['user_id']; // Get the logged-in user's ID

// Fetch all active job postings from the database
$sql = "SELECT * FROM job_postings WHERE status = 'ACTIVE' ORDER BY created_at DESC";
$result = $conn->query($sql);
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Job Postings</title>
    <link rel="stylesheet" href="styles.css"> <!-- Include your CSS styles here -->
```

```
</head>
```

```
<body>
```

```
<?php include 'header.php'; ?> <!-- Include a common header if needed -->
```

```
<div class="container">
```

```
    <h2>Active Job Postings</h2>
```

```
    <?php if ($result->num_rows > 0): ?>
```

```
        <table border="1">
```

```
            <thead>
```

```
                <tr>
```

```
                    <th>Job Title</th>
```

```
                    <th>Company</th>
```

```
                    <th>Location</th>
```

```
                    <th>Salary Range</th>
```

```
                    <th>Description</th>
```

```
                    <th>Openings</th>
```

```
                    <th>Deadline</th>
```

```
                    <th>Action</th>
```

```
                </tr>
```

```
            </thead>
```

```
            <tbody>
```

```
                <?php while ($row = $result->fetch_assoc()):
```

```
                    $job_id = $row['job_id'];
```

```
                    // Check if the user has already applied for this job
```

```
                    $application_check_sql = "SELECT application_id, application_status FROM applications
```

```
                        WHERE job_id = '$job_id' AND profile_id =
```

```

        (SELECT profile_id FROM profiles WHERE user_id = '$user_id')
        AND application_status != 'WITHDRAWN'";

    $application_check_result = $conn->query($application_check_sql);

    $already_applied = $application_check_result->num_rows > 0;
    $application = $application_check_result->fetch_assoc();
?>
<tr>
    <td><?php echo htmlspecialchars($row['job_title']); ?></td>
    <td><?php echo htmlspecialchars($row['company']); ?></td>
    <td><?php echo htmlspecialchars($row['location']); ?></td>
    <td><?php echo htmlspecialchars($row['min_salary']) . ' - ' .
htmlspecialchars($row['max_salary']); ?></td>
    <td><?php echo htmlspecialchars($row['description']); ?></td>
    <td><?php echo htmlspecialchars($row['openings']); ?></td>
    <td><?php echo htmlspecialchars($row['deadline']); ?></td>
    <td>
        <?php if ($already_applied && $application['application_status'] != 'WITHDRAWN'): ?>
            <span>Already Applied</span>
            <a href="withdraw_application.php?application_id=<?php echo
$application['application_id']; ?>"
                onclick="return confirm('Are you sure you want to withdraw your
application?')">Withdraw Application</a>
        <?php else: ?>
            <a href="apply_job.php?job_id=<?php echo $row['job_id']; ?>">Apply Now</a>
        <?php endif; ?>
    </td>
</tr>
<?php endwhile; ?>
</tbody>

```

```

        </table>

<?php else: ?>

    <p>No active job postings at the moment.</p>

<?php endif; ?>

</div>

<?php include 'footer.php'; ?> <!-- Include a common footer if needed -->

</body>
</html>

<?php
// Close the database connection
$conn->close();
?>

withdraw_application.php

<?php
include '../db.php';
session_start();

// Check if the user is logged in as an applicant
if (!isset($_SESSION['user_id']) || $_SESSION['role'] != 'APPLICANT') {
    header('Location: login.php');
    exit();
}

```

```

// Check if the application_id is provided via GET
if (!isset($_GET['application_id'])) {
    echo "Error: Application ID not provided.";
    exit();
}

$application_id = (int)$_GET['application_id']; // Sanitize application_id

// Start a transaction to ensure atomicity
$conn->begin_transaction();

try {
    // Update the application status to 'WITHDRAWN' and set the withdrawn timestamp
    $withdraw_sql = "UPDATE applications SET application_status = 'WITHDRAWN', withdrawn_at =
NOW() WHERE application_id = '$application_id'";

    if ($conn->query($withdraw_sql) === TRUE) {
        // Update the tbl_pipeline_stage to record the withdrawal
        $withdraw_pipeline_sql = "UPDATE tbl_pipeline_stage SET withdrawn_at = NOW() WHERE
application_id = '$application_id'";

        if (!$conn->query($withdraw_pipeline_sql)) {
            throw new Exception("Error updating pipeline stage: " . $conn->error);
        }

        // Fetch the job_id related to the application
        $job_id_sql = "SELECT job_id FROM applications WHERE application_id = '$application_id'";
        $job_id_result = $conn->query($job_id_sql);

        if ($job_id_result->num_rows > 0) {
            $job_row = $job_id_result->fetch_assoc();

```

```

$job_id = $job_row['job_id'];

// Update tbl_job_metrics
// Decrement total_applicants and increment withdrawn_applicants
$update_metrics_sql = "UPDATE tbl_job_metrics
    SET withdrawn_applicants = withdrawn_applicants + 1
    WHERE job_id = '$job_id'";

if (!$conn->query($update_metrics_sql)) {
    throw new Exception("Error updating job metrics: " . $conn->error);
}
}

// Commit the transaction
$conn->commit();

// Redirect to the correct location
header('Location: view_job.php');
exit();
} else {
    throw new Exception("Error updating application status: " . $conn->error);
}
} catch (Exception $e) {
    // Rollback the transaction on error
    $conn->rollback();
    echo "Error withdrawing application: " . $e->getMessage();
}

$conn->close();

```

?>