

FLEXIHIRE

Signup.php

```
<?php
    $role=1;//user
    session_start();

    $name = $email = $password = $confirmpassword = $lastname=$dob="";
    $error = [];

    include '../database/connectdatabase.php';

    if ($_SERVER['REQUEST_METHOD'] == 'POST')
    {
        $name = $_POST['name'];
        $lastname=$_POST['lastname'];
        $email = $_POST['email'];
        $dob=$_POST['dob'];
        $password = $_POST['password'];

        $dbname = "project";
        mysqli_select_db($conn, $dbname);

        // Check if email is already linked with an account
        $check_duplicate = "
        SELECT 'email' AS type FROM tbl_login WHERE email = ?
        UNION
        SELECT 'user' AS type FROM tbl_user WHERE first_name = ? AND last_name = ?
        AND dob = ?";

        $stmt_check = mysqli_prepare($conn, $check_duplicate);
        mysqli_stmt_bind_param($stmt_check, "ssss", $email, $name, $lastname, $dob);
        mysqli_stmt_execute($stmt_check);
        mysqli_stmt_store_result($stmt_check);
```

```

if (mysqli_stmt_num_rows($stmt_check) > 0)
{
    mysqli_stmt_bind_result($stmt_check, $duplicateType);
    mysqli_stmt_fetch($stmt_check);
    if ($duplicateType == 'email')
    {
        $_SESSION['email_error'] = "This email is already registered!";
    } else
    {
        $_SESSION['email_error'] = "User with this name and date of birth already
exists!";
    }
    header("Location: signup.php");
    exit();
}
else
{
    $hashed_password = password_hash($password, PASSWORD_DEFAULT);
    // Insert into tbl_user
    $sql_user = "INSERT INTO tbl_user (first_name,last_name, dob) VALUES (?, ?, ?)";
    $stmt_user = mysqli_prepare($conn, $sql_user);
    mysqli_stmt_bind_param($stmt_user, "sss", $name, $lastname, $dob);

    if (mysqli_stmt_execute($stmt_user))
    {
        // fetches employer_id
        $user_id = mysqli_insert_id($conn);
        // Insert into tbl_login
        $sql_login = "INSERT INTO tbl_login (email, password, role,user_id) VALUES (?,
?, ?,?)";

```

```
$stmt_login = mysqli_prepare($conn, $sql_login);  
mysqli_stmt_bind_param($stmt_login, "ssii",$email, $hashed_password,  
$role,$user_id);
```

```
if (mysqli_stmt_execute($stmt_login))  
{  
    header("Location: ../Login/loginvalidation.php");  
    exit();  
}  
else  
{  
    echo "Error inserting into tbl_login: " . mysqli_error($conn);  
}  
}  
else  
{  
    echo "Error inserting into tbl_user: " . mysqli_error($conn);  
}  
}  
}  
?>
```

```
<!DOCTYPE html>
```

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

```
<head>
```

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

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

```
<title>Create an Account</title>
```

```
<link rel="stylesheet" href="signup.css">
```

```
<script>
```

```
function validatename()
```

```

{
    const name = document.getElementById("name").value;
    const error = document.getElementById("nameError");
    if (!name)
    {
        error.textContent = "Name is required."; //textContent is used to get content from
html element
        return false;
    }
    else if (!/^[a-zA-Z]*$/ .test(name))
    {
        error.textContent = "Should not contain digits or special characters or spaces";
        return false;
    }
    else if (name.length < 4)
    {
        error.textContent = "Name must be at least 5 characters.";
        return false;
    }

    error.textContent = ""; //used to clear the error message that was previously displayed
in the corresponding error <p> element

    return true;
}

```

```

function validatelastname()

```

```

{
    const name = document.getElementById("lastname").value;
    const error = document.getElementById("lastnameerror");
    if (!name)
    {

```

```
        error.textContent = "Last Name is required."; //textContent is used to get content  
from html element
```

```
        return false;
```

```
    }
```

```
    else if (!/^[a-zA-Z]*$/ .test(name))
```

```
    {
```

```
        error.textContent = "Should not contain digits or special characters or spaces";
```

```
        return false;
```

```
    }
```

```
    error.textContent = ""; //used to clear the error message that was previously displayed  
in the corresponding error <p> element
```

```
    return true;
```

```
}
```

```
function validateEmail()
```

```
{
```

```
    const email = document.getElementById("email").value;
```

```
    const error = document.getElementById("emailError");
```

```
    // Check if email is empty
```

```
    if (!email)
```

```
    {
```

```
        error.textContent = "E-mail is required.";
```

```
        return false;
```

```
    }
```

```
    // Check if email starts with a space or invalid character
```

```
    if (email[0] === " ")
```

```
    {
```

```
        error.textContent = "E-mail must start with a letter.";
```

```
        return false;
```

```

    }

    // Check if email has valid format and domain

    // const emailRegex = ;

    if (!/^[a-zA-Z0-9][^\s@]*@(gmail\.com|yahoo\.com|hotmail\.com|amaljyothi\.ac\.in|mca\.ajce\.in)$/.test(email))

    {
        // Check if it's the domain that's invalid

        if (email.includes('@'))

        {
            const domain = email.split('@')[1];

            if (domain !== 'gmail.com' && domain !== 'yahoo.com')

            {
                error.textContent = "Invalid domain";

                return false;
            }
        }

        error.textContent = "Invalid email address.";

        return false;
    }

    // Clear error message if all validations pass

    error.textContent = "";

    return true;
}

function validatedob()

{
    const dobField = document.getElementById("dob");//dobField stores the entire input
    element (<input type="date">).

    const error = document.getElementById("doberror");

```

```
const dob = dobField.value;//dob stores just the user's entered value (which is a string).
```

```
error.textContent = ""; // Clear previous errors
```

```
if (!dob)
{
    error.textContent = "Date of Birth is required.";
    return false;
}
```

```
const birthdate = new Date(dob);
```

```
const today = new Date();// stores the current date so we can compare the entered
DOB against today's date
```

```
let age = today.getFullYear() - birthdate.getFullYear();//calculates actual age
```

```
const monthdiff = today.getMonth() - birthdate.getMonth();
```

```
const daydiff = today.getDate() - birthdate.getDate();
```

```
// Adjust age if birthday hasn't occurred this year
```

```
if (monthdiff < 0 || (monthdiff === 0 && daydiff < 0))
```

```
{
    age--;
}
```

```
let isValid = true;
```

```
// Future Date Check
```

```
if (birthdate > today)
```

```
{
    error.textContent += "Date of Birth cannot be in the future.";
    isValid = false;
}
```

```

    }

    // Age Check (Must be 18+)
    if (age < 18)
    {
        error.textContent += "Must be at least 18 years old.";
        isValid = false;
    }

    return isValid;
}

function validatePassword()
{
    const password = document.getElementById("password").value;
    const error = document.getElementById("passwordError");
    if (!password)
    {
        error.textContent = "Password is required.";
        return false;
    }
    else if (password.length < 6)
    {
        error.textContent = "Password must be at least 6 characters.";
        return false;
    }
    else if (!/^[^a-zA-Z0-9]/.test(password))
    {
        error.textContent = "Password should contain atleast one special character.";
        return false;
    }

```



```

    }

    error.textContent = "";

    return true;
}

function validateConfirmPassword()
{
    const password = document.getElementById("password").value;
    const confirmpassword = document.getElementById("confirmpassword").value;
    const error = document.getElementById("confirmpasswordError");
    if (!confirmpassword)
    {
        error.textContent = "Confirm password is required.";
        return false;
    }
    else if (confirmpassword !== password)
    {
        error.textContent = "Passwords do not match.";
        return false;
    }
    error.textContent = "";
    return true;
}

```

function validateForm() // It ensures that all fields in the form meet their respective validation criteria before allowing the form to be submitted.

```

{
    const validname = validatename();
    const lastname=validatelastname();
    const validEmail = validateEmail();
    const dob=validatedob();

```

```

const validPassword = validatePassword();

const validConfirmPassword = validateConfirmPassword();

return validname && lastname && validEmail && dob && validPassword &&
validConfirmPassword;//The && operator combines the results of all the validations. If any
of the individual validations return false, the overall result will also be false.

}
</script>
</head>
<body>
<div class="container">
<div class="left-section">
<a href="../homepage/homepage.html" class="back-to-website"><span class="arrow-
circle">←</span> Back to website</a>
</div>
<div class="right-section">
<h2>Create an account</h2>
<button id="user1"><a href="../Signup/signup.php" id="user">User</a></button>
<button id="employer1"><a href="../Signup/employersignup.php"
id="employer">Employer</a></button>
<p>Already have an account? <a href="../Login/loginvalidation.php" style="color:
#665efc; text-decoration: none;">Log in</a></p>
<form method="post" onsubmit="return validateForm()">
<div style="display: flex; gap: 10px;">
<div>
<input type="text" id="name" name="name" placeholder="First Name"
onkeyup="validatename()">
<p class="error" id="nameError"></p>
</div>
<div>
<input type="text" id="lastname" name="lastname" placeholder="Last Name"
onkeyup="validatelastname()">
<p class="error" id="lastnameerror"></p>
</div>

```

```
</div>

<div>

    <input type="email" id="email" name="email" placeholder="E-
mail" onkeyup="validateEmail()">

    <p class="error" id="emailError">

        <?php
            if(isset($_SESSION['email_error']))
            {
                echo $_SESSION['email_error'];
                unset($_SESSION['email_error']);
            }
        ?>

    </p>
</div>

<div>

    <input type="date" id="dob" name="dob" placeholder="Date of Birth"
onkeyup="validatedob()">

    <p class="error" id="doberror"></p>
</div>

<div>

    <input type="password" id="password" name="password" placeholder="Enter
your password" onkeyup="validatePassword()">

    <p class="error" id="passwordError"></p>
</div>

<div>

    <input type="password" id="confirmpassword" name="confirmpassword"
placeholder="Confirm your password" onkeyup="validateConfirmPassword()">

    <p class="error" id="confirmpasswordError"></p>
</div>

<button type="submit">Create Account</button>

</form>

</div>
```

```
</div>
</body>
</html>
```

Signup.js

```
function validateForm() {
    // Get form elements

    const name = document.getElementById("name").value.trim();
    const email = document.getElementById("email").value.trim();
    const password = document.getElementById("password").value.trim();
    const confirmPassword = document.getElementById("confirmpassword").value.trim();

    // Error elements

    const nameError = document.getElementById("nameError");
    const emailError = document.getElementById("emailError");
    const passwordError = document.getElementById("passwordError");
    const confirmPasswordError = document.getElementById("confirmpasswordError");

    // Clear previous error messages
    nameError.textContent = "";
    emailError.textContent = "";
    passwordError.textContent = "";
    confirmPasswordError.textContent = "";

    // Validation flags
    let isValid = true;

    // Name validation
    if (name === "") {
        nameError.textContent = "Name is required.";
    }
```

```
        isValid = false;
    } else if (!/^[a-zA-Z ]*$/.test(name)) {
        nameError.textContent = "Name should not contain digits or special characters.";
        isValid = false;
    } else if (name.length < 5) {
        nameError.textContent = "Minimum length for the name should be 5.";
        isValid = false;
    }

    // Email validation
    if (email === "") {
        emailError.textContent = "E-mail is required.";
        isValid = false;
    } else if (!/^[^\\s@]+@[^\\s@]+\\.[^\\s@]+$/ .test(email)) {
        emailError.textContent = "Invalid email address.";
        isValid = false;
    }

    // Password validation
    if (password === "") {
        passwordError.textContent = "Password is required.";
        isValid = false;
    } else if (password.length < 6) {
        passwordError.textContent = "Password must be at least 6 characters long.";
        isValid = false;
    } else if (!/[!@#$%^&*()_+=-]/.test(password)) {
        passwordError.textContent = "Password must include at least one special character.";
        isValid = false;
    }
}
```

```
// Confirm password validation
if (confirmPassword === "") {
    confirmPasswordError.textContent = "Confirm password is required.";
    isValid = false;
} else if (confirmPassword !== password) {
    confirmPasswordError.textContent = "Passwords do not match.";
    isValid = false;
}

// Prevent form submission if validation fails
return isValid;
}
```

Loginvalidation.php

```
<?php
session_start();
session_regenerate_id(true); // Regenerate session ID for security

include '../database/connectdatabase.php';
$dbname = "project";
mysqli_select_db($conn, $dbname);

$errorMessage = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $email = trim($_POST['email']);
    $password = trim($_POST['password']);

    // Check login credentials and status

    $sql = "SELECT login_id, email, password, role, user_id, employer_id, status FROM
tbl_login WHERE email = ?";
```

```

$stmt = mysqli_prepare($conn, $sql);
mysqli_stmt_bind_param($stmt, "s", $email);
mysqli_stmt_execute($stmt);
$result = mysqli_stmt_get_result($stmt);

if ($row = mysqli_fetch_assoc($result)) {
    // First check if account is active
    if ($row['status'] === 'inactive') {
        $errorMessage = "Your account has been deactivated. Please contact support.";
    } else if (password_verify($password, $row['password'])) {
        // Clear any existing session data
        session_unset();

        // Set common session variables
        $_SESSION['logged_in'] = true;
        $_SESSION['email'] = $row['email'];
        $_SESSION['login_id'] = $row['login_id'];
        $_SESSION['role'] = $row['role'];

        switch ($row['role']) {
            case 1: // User
                $_SESSION['user_type'] = 'user';
                $_SESSION['user_id'] = $row['user_id'];
                header("Location: ../userdashboard/userdashboard.php");
                break;

            case 2: // Employer
                $_SESSION['user_type'] = 'employer';
                $_SESSION['employer_id'] = $row['employer_id'];
                header("Location: ../userdashboard/employerdashboard.php");
            }
        }
    }
}

```

```
        break;

    case 3: // Admin
        $_SESSION['user_type'] = 'admin';
        $_SESSION['admin_id'] = $row['login_id'];
        header("Location: ../userdashboard/admindashboard.php");
        break;

    default:
        $errorMessage = "Invalid user role";
        session_destroy();
        break;
}

if (empty($errorMessage)) {
    // Set session cookie parameters for better security
    $params = session_get_cookie_params();
    setcookie(session_name(), session_id(), [
        'expires' => time() + 3600, // 1 hour
        'path' => '/',
        'domain' => "",
        'secure' => true,
        'httponly' => true,
        'samesite' => 'Lax'
    ]);

    exit();
}
} else {
    $errorMessage = "Invalid password";
```



```
    }  
  } else {  
    $errorMessage = "Email not found";  
  }  
}  
  
// If not POST request or login failed, show the login form  
?>
```

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <title>Login</title>  
  <link rel="stylesheet" href="loginvalidation.css">  
  <script>  
    function validateEmail() {  
      const email = document.getElementById("email").value;  
      const emailError = document.getElementById("email-error");  
  
      if (!email) {  
        emailError.textContent = "E-mail is required.";  
        return false;  
      }  
  
      if (email[0] === " ") {  
        emailError.textContent = "E-mail must not start with a space.";  
        return false;  
      }  
    }  
  }  
</script>  
</head>  
<body>  
  <div class="login-validation">  
    <div class="form">  
      <div class="input">  
        <input type="text" value="" />  
      </div>  
      <div class="input">  
        <input type="password" value="" />  
      </div>  
      <div class="button">  
        <input type="button" value="Login" />  
      </div>  
    </div>  
    <div class="error">  
      <div class="error-message">  
        <div class="error-text"></div>  
      </div>  
  </div>  
</body>  
</html>
```

```
const emailRegex = /^[a-zA-Z0-9][^\s@]*@(gmail\.com|yahoo\.com|hotmail\.com|amaljyothi\.ac\.in|mca\.ajce\.in)$/;
```

```
if (!emailRegex.test(email)) {  
    emailError.textContent = "Invalid email format."  
    return false;  
}
```

```
emailError.textContent = "";  
return true;  
}
```

```
function validateForm() {  
    return validateEmail();  
}
```

```
</script>
```

```
</head>
```

```
<body>
```

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

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

```
<a href="../homepage/homepage.html" class="back-to-website">
```

```
<span class="arrow-circle">←</span> Return to Homepage
```

```
</a>
```

```
</div>
```

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

```
<h1><span id="wel">Welcome Back<br></span><span id="log">Sign  
In</span></h1>
```

```
<?php if(!empty($errorMessage)): ?>
```

```

        <div class="error-message"><?php echo $errorMessage; ?></div>

    <?php endif; ?>

    <form method="POST" id="form" onsubmit="return validateForm()">

        <input type="email" id="email" name="email" placeholder="Enter your email
address" onkeyup="validateEmail()" required>

        <p class="error" id="email-error"></p>

        <input type="password" name="password" placeholder="Enter your password"
required>

        <input type="submit" value="Sign In">

    </form>

    <p id="register">New to our platform? <a href="../Signup/signup.php">Create an
Account</a></p>

    <p class="forgot-password">Forgot Password? <a href="forgotpassword.php">Reset
Password</a></p>

</div>

</div>

</body>

</html>

```

Userdashboard.php

```

<?php
    session_start();

    // Message handling
    if (isset($_SESSION['message'])) {
        $message = $_SESSION['message'];
        $message_type = $_SESSION['message_type'] ?? 'info';
        unset($_SESSION['message']);
    }

```

```

        unset($_SESSION['message_type']);
    }

    // Check if user is logged in
    if (!isset($_SESSION['user_id']))
    {
        header("Location: ../login/loginvalidation.php");
        exit();
    }

    include '../database/connectdatabase.php';
    $dbname = "project";
    mysqli_select_db($conn, $dbname);

    // Initialize variables
    $username = "";
    $email = "";

    // Get user data with updated query
    $user_id = $_SESSION['user_id'];
    $sql = "SELECT l.email, u.first_name, u.last_name, u.username, u.profile_image,
u.user_id
        FROM tbl_login l
        INNER JOIN tbl_user u ON l.user_id = u.user_id
        WHERE l.user_id = ?";

    $stmt = mysqli_prepare($conn, $sql);
    mysqli_stmt_bind_param($stmt, "i", $user_id);
    mysqli_stmt_execute($stmt);
    $result = mysqli_stmt_get_result($stmt);

```

```

if ($result && mysqli_num_rows($result) > 0) {
    $user_data = mysqli_fetch_assoc($result);
    $email = $user_data['email'];

    // Set display name based on available data
    if (!empty($user_data['username']))
    {
        $display_name = $user_data['username'];
    }
    else
    {
        $display_name = $user_data['first_name'] . " " . $user_data['last_name'];
    }

    $profile_image = $user_data['profile_image'];
    $employer_id = $user_data['user_id'];

    // Store in session for consistency
    $_SESSION['display_name'] = $display_name;
}
else
{
    error_log("Database error or user not found for ID: $user_id");
    session_destroy();
    header("Location: ../login/loginvalidation.php");
    exit();
}

// Fetch active jobs based on selected date and deadline
$date_filter = isset($_POST['date']) ? $_POST['date'] : null;

```

```
$is_reset = isset($_POST['reset']) && $_POST['reset'] == 1;
```

```
$sql_fetch = "SELECT j.*, e.company_name, e.profile_image,  
                (SELECT COUNT(*) FROM tbl_applications WHERE job_id = j.job_id AND  
status = 'accepted') AS total_accepted,  
                (SELECT status FROM tbl_applications WHERE job_id = j.job_id AND user_id  
= ? LIMIT 1) AS application_status  
FROM tbl_jobs j  
JOIN tbl_employer e ON j.employer_id = e.employer_id  
WHERE j.status = 'approved'  
AND j.is_deleted = 0  
AND j.vacancy_date >= CURDATE() /* Only show jobs where deadline hasn't  
passed */  
ORDER BY j.created_at DESC";
```

```
$stmt = mysqli_prepare($conn, $sql_fetch);  
mysqli_stmt_bind_param($stmt, "i", $user_id);  
mysqli_stmt_execute($stmt);  
$result = mysqli_stmt_get_result($stmt);
```

```
// If it's an AJAX request, return only the job listings  
if ($_SERVER['REQUEST_METHOD'] === 'POST') {  
    if ($result && mysqli_num_rows($result) > 0) {  
        while ($row = mysqli_fetch_assoc($result)) {  
            $job_id = $row['job_id'];  
  
            // Check if user has already applied for this job  
            $check_sql = "SELECT status FROM tbl_applications WHERE user_id = ? AND  
job_id = ?";  
            $check_stmt = mysqli_prepare($conn, $check_sql);  
            mysqli_stmt_bind_param($check_stmt, "ii", $user_id, $job_id);  
            mysqli_stmt_execute($check_stmt);
```

```

$application_result = mysqli_stmt_get_result($check_stmt);

$has_applied = mysqli_num_rows($application_result) > 0;

$application_status = $has_applied ?
mysqli_fetch_assoc($application_result)['status'] : null;

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

echo '<div class="job-header">';

echo '<h3 class="job_title">' . htmlspecialchars($row['job_title']) . '</h3>';

echo '<span class="salary">₹' . htmlspecialchars($row['salary']) . '</span>';

echo '</div>';

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

echo '<p class="description">' . htmlspecialchars($row['job_description']) . '</p>';

echo '<p class="location"><i class="fas fa-map-marker-alt"></i> ' .
htmlspecialchars($row['location']) . ', ' . htmlspecialchars($row['town']) . '</p>';

echo '<p class="date"><i class="fas fa-calendar-plus"></i> Posted: ' . date('Y-m-d',
strtotime($row['created_at'])) . '</p>';

echo '<p class="date"><i class="fas fa-calendar-alt"></i> Vacancy Date: ' . date('Y-
m-d', strtotime($row['vacancy_date'])) . '</p>';

echo '</div>';

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

echo '<div class="button-group">';

echo '<a href="jobdetails.php?job_id=' . $job_id . '" class="details-btn"><i
class="fas fa-info-circle"></i> Details</a>';

if ($application_status) {

    echo '<button class="applied-btn" disabled><i class="fas fa-check-circle"></i> '
. htmlspecialchars($application_status) . '</button>';

} else {

    echo '<a href="applyjob.php?user_id=' . $user_id . '&job_id=' . $job_id . '"
class="apply-link">';

    echo '<button class="apply-btn"><i class="fas fa-paper-plane"></i> Apply
Now</button>';

    echo '</a>';

}

```

```

        echo '</div>';

        echo '</div>';

        echo '</div>';

    }
} else {
    echo '<div class="no-jobs">No jobs available.</div>';
}

exit();
}

// Only process error messages if they came from a redirect
if (isset($_GET['error']) && isset($_SERVER['HTTP_REFERER'])) {
    $error_message = "";
    switch ($_GET['error']) {
        case 'already_applied':
            $error_message = 'You have already applied for this job';
            break;
        case 'job_not_found':
            $error_message = 'Job listing not found';
            break;
        case 'application_failed':
            $error_message = 'Failed to submit application. Please try again';
            break;
        default:
            $error_message = 'An error occurred';
    }

    // Store the message in session and redirect to clean URL
    $_SESSION['error_message'] = $error_message;
    header('Location: ' . strtok($_SERVER['REQUEST_URI'], '?'));
    exit();
}

```



```
}
```

```
// Display message from session if exists and then clear it
```

```
if (isset($_SESSION['error_message'])) {
```

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

```
    unset($_SESSION['error_message']);
```

```
}
```

```
// Function to check if vacancy is filled
```

```
function isVacancyFilled($conn, $job_id) {
```

```
    $sql = "SELECT j.vacancy,
```

```
           (SELECT COUNT(*) FROM tbl_applications WHERE job_id = ? AND status =  
'accepted') AS total_accepted
```

```
           FROM tbl_jobs j
```

```
           WHERE j.job_id = ?";
```

```
    $stmt = mysqli_prepare($conn, $sql);
```

```
    mysqli_stmt_bind_param($stmt, "ii", $job_id, $job_id);
```

```
    mysqli_stmt_execute($stmt);
```

```
    $result = mysqli_stmt_get_result($stmt);
```

```
    $row = mysqli_fetch_assoc($result);
```

```
    return $row['total_accepted'] >= $row['vacancy'];
```

```
}
```

```
?>
```

```
<!DOCTYPE html>
```

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

```
<head>
```

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

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

```
    <title>User Dashboard</title>
```

```

<link rel="stylesheet" href="userdashboard.css">

<link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600&display=s
wap" rel="stylesheet">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css">

</head>

<body>

<!-- Navigation Bar -->

<nav class="navbar">

    <div class="nav-brand">

        

        <h1>FlexiHire</h1>

    </div>

    <div class="nav-right">

        <div class="profile-info">

            <span class="nav-username"><?php echo htmlspecialchars($display_name);
?></span>

            <div class="profile-container">

                <?php
                $profile_path = "/mini project/database/profile_picture/" . $profile_image;
                if (!empty($profile_image) && file_exists($_SERVER['DOCUMENT_ROOT'] .
                $profile_path)): ?>

                    

                <?php else: ?>

                    

                <?php endif; ?>

                <div class="dropdown-menu">

                    <a href="profiles/user/userprofile.php"><i class="fas fa-user"></i>
                Profile</a>

                    <a href="..../login/logout.php"><i class="fas fa-sign-out-alt"></i> Logout</a>

```

```
        </div>
    </div>
</div>
</div>
</nav>
```

```
<!-- Toast Notification -->
```

```
<div id="toast" class="toast">
    <div class="toast-content">
        <div class="toast-icon">
            <i class="fas fa-check-circle"></i>
        </div>
        <div class="toast-message-container">
            <div class="toast-title"></div>
            <span id="toast-message"></span>
        </div>
    </div>
</div>
```

```
<!-- Main Content -->
```

```
<div class="dashboard-container">
    <!-- Sidebar -->
    <aside class="sidebar">
        <div class="sidebar-menu">
            <a href="userdashboard.php" class="active"><i class="fas fa-home"></i>
Home</a>
            <a href="applied.php"><i class="fas fa-paper-plane"></i> Applied Job</a>
            <a href="bookmark.php"><i class="fas fa-bookmark"></i> Bookmarks</a>
            <a href="appointment.php"><i class="fas fa-calendar"></i> Appointments</a>
            <a href="reportedjobs.php"><i class="fas fa-flag"></i> Reported Jobs</a>
            <a href="reviews.php"><i class="fas fa-star"></i> Reviews</a>
```

```
<a href="profiles/user/userprofile.php"><i class="fas fa-user"></i> Profile</a>
</div>

<div class="logout-container">

    <div class="sidebar-divider"></div>

    <a href="../login/logout.php" class="logout-link">

        <i class="fas fa-sign-out-alt"></i> Logout

    </a>

</div>

</aside>

<!-- Main Content Area -->

<main class="main-content">

    <!-- Search Bar -->

    <div class="search-container">

        <div class="search-box">

            <i class="fas fa-search"></i>

            <input type="text" placeholder="Search jobs..." id="search" name="search"
oninput="filterjobs()">

        </div>

        <div class="filter-box">

            <input type="text" placeholder="Location" id="location" name="location"
oninput="filterlocation()">

            <div class="salary-range">

                <input type="number" placeholder="Min Salary" id="minsalary"
name="minsalary" oninput="filterminsalary()">

                <input type="number" placeholder="Max Salary" id="maxsalary"
name="maxsalary" oninput="filtermaxsalary()">

            </div>

            <input type="date" id="date" name="date" oninput="filterdate()">

            <button class="reset-button" onclick="resetFilters()">

                <i class="fas fa-undo"></i> Reset Filters

            </button>

        </div>

    </div>

</main>

</div>
```

```
</div>
```

```
</div>
```

```
<!-- Job Listings -->
```

```
<div class="job-listings">
```

```
<?php
```

```
if ($result && mysqli_num_rows($result) > 0) {
```

```
    while ($row = mysqli_fetch_assoc($result))
```

```
    {
```

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

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

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

```
        $check_sql = "SELECT status FROM tbl_applications WHERE user_id = ?  
AND job_id = ?";
```

```
        $check_stmt = mysqli_prepare($conn, $check_sql);
```

```
        mysqli_stmt_bind_param($check_stmt, "ii", $user_id, $job_id);
```

```
        mysqli_stmt_execute($check_stmt);
```

```
        $application_result = mysqli_stmt_get_result($check_stmt);
```

```
        $has_applied = mysqli_num_rows($application_result) > 0;
```

```
        $application_status = $has_applied ?  
mysqli_fetch_assoc($application_result)['status'] : null;
```

```
    ?>
```

```
<div class="job-card">
```

```
    <div class="job-header">
```

```
        <h3 class="job_title"><?php echo htmlspecialchars($row['job_title']);  
?></h3>
```

```
        <span class="salary">₹<?php echo htmlspecialchars($row['salary']);  
?></span>
```

```
    </div>
```

```
<div class="job-body">
```

```

        <p class="description"><?php echo
htmlspecialchars($row['job_description']); ?></p>

        <p class="location">

            <i class="fas fa-map-marker-alt"></i>

            <?php echo htmlspecialchars($row['location']) . ', ' .
htmlspecialchars($row['town']); ?>

        </p>

        <p class="date">

            <i class="fas fa-calendar-plus"></i>

            Posted: <?php echo date('Y-m-d', strtotime($row['created_at'])); ?>

        </p>

        <p class="date">

            <i class="fas fa-calendar-alt"></i>

            Vacancy Date: <?php echo date('Y-m-d',
strtotime($row['vacancy_date'])); ?>

        </p>
    </div>

    <div class="job-footer">

        <div class="button-group">

            <a href="jobdetails.php?job_id=<?php echo $job_id; ?>"
class="details-btn">

                <i class="fas fa-info-circle"></i> Details

            </a>

            <?php if (isVacancyFilled($conn, $job_id)): ?>

                <div class="job-closed">

                    <i class="fas fa-user-check"></i> Positions Filled

                </div>

            <?php else: ?>

                <?php if ($has_applied): ?>

                    <button class="applied-btn" disabled>

                        <i class="fas fa-check-circle"></i>

                        <?php echo htmlspecialchars(ucfirst($application_status)); ?>

```

```

        </button>

        <?php else: ?>

            <a href="applyjob.php?user_id=<?php echo $user_id;
?>&job_id=<?php echo $job_id; ?>" class="apply-link">

                <button class="apply-btn">

                    <i class="fas fa-paper-plane"></i> Apply Now

                </button>

            </a>

        <?php endif; ?>

    <?php endif; ?>

</div>

</div>

</div>

<?php
}
} else {
    echo '<div class="no-jobs">No active jobs available.</div>';
}
?>

</div>

</main>

</div>

<script src="shared-search.js"></script>
<script src="userdashboard.js"></script>

<style>
.job-footer {
    display: flex;
    justify-content: flex-end;
    align-items: center;

```

```
padding-top: 15px;  
border-top: 1px solid #eee;  
}
```

```
.button-group {  
  display: flex;  
  gap: 15px;  
  align-items: center;  
}
```

```
.details-btn {  
  padding: 10px 20px;  
  border-radius: 8px;  
  font-weight: 600;  
  font-size: 14px;  
  color: #9747FF;  
  text-decoration: none;  
  display: inline-flex;  
  align-items: center;  
  gap: 8px;  
  transition: all 0.3s ease;  
}
```

```
.details-btn:hover {  
  background: rgba(151, 71, 255, 0.1);  
}
```

```
.apply-btn {  
  padding: 10px 20px;  
  border-radius: 8px;
```



```
font-weight: 600;
font-size: 14px;
transition: all 0.3s ease;
background: #9747FF;
color: white;
border: none;
cursor: pointer;
display: inline-flex;
align-items: center;
gap: 8px;
}

.apply-btn:hover {
  transform: translateY(-2px);
  box-shadow: 0 5px 15px rgba(151, 71, 255, 0.2);
}

.apply-btn i {
  font-size: 16px;
}

.job-card {
  background: white;
  border-radius: 12px;
  padding: 20px;
  box-shadow: 0 2px 4px rgba(0,0,0,0.1);
  transition: transform 0.3s ease;
}

.job-card:hover {
```

```
    transform: translateY(-2px);
    box-shadow: 0 5px 15px rgba(0,0,0,0.1);
}

.job-footer a {
    text-decoration: none !important;
}

.toast {
    visibility: hidden;
    position: fixed;
    top: 20px;
    right: 20px;
    min-width: 300px;
    max-width: 500px;
    background-color: lightgreen;
    box-shadow: 0 4px 12px rgba(0, 0, 0, 0.15);
    border-radius: 8px;
    z-index: 1000;
    overflow: hidden;
}

.toast.show {
    visibility: visible;
    animation: slideInRight 0.3s, fadeOut 0.5s 2.5s;
}

.toast-content {
    display: flex;
    align-items: center;
```

```
padding: 16px 20px;  
gap: 12px;  
}
```

```
.toast-icon {  
  flex-shrink: 0;  
}
```

```
.toast-icon i {  
  font-size: 24px;  
}
```

```
.toast.success .toast-icon i {  
  color: #4CAF50;  
}
```

```
.toast.error .toast-icon i {  
  color: #f44336;  
}
```

```
.toast-message-container {  
  flex-grow: 1;  
}
```

```
.toast-title {  
  font-weight: 600;  
  margin-bottom: 4px;  
}
```

```
#toast-message {
```

```
    color: #666;  
}
```

```
@keyframes slideInRight {  
  from {  
    transform: translateX(100%);  
  }  
  to {  
    transform: translateX(0);  
  }  
}
```

```
@keyframes fadeOut {  
  from {  
    opacity: 1;  
  }  
  to {  
    opacity: 0;  
  }  
}
```

```
.applied-btn {  
  background-color: #4CAF50;  
  color: white;  
  border: none;  
  padding: 10px 20px;  
  border-radius: 8px;  
  font-size: 14px;  
  display: flex;  
  align-items: center;
```

```
    gap: 8px;
    cursor: default;
    transition: all 0.3s ease;
    opacity: 0.9;
}

.applied-btn i {
    color: #fff;
    font-size: 16px;
    animation: checkmark 0.5s ease-in-out;
}

@keyframes checkmark {
    0% {
        transform: scale(0);
    }
    50% {
        transform: scale(1.2);
    }
    100% {
        transform: scale(1);
    }
}

.applied-btn.disabled {
    opacity: 1;
    background-color: #4CAF50;
    cursor: default;
}
```

```
.applied-btn:hover {  
  box-shadow: 0 2px 8px rgba(76, 175, 80, 0.3);  
}
```

```
.nav-right {  
  display: flex;  
  align-items: center;  
}
```

```
.profile-info {  
  display: flex;  
  align-items: center;  
  gap: 12px;  
}
```

```
.nav-username {  
  color: #333;  
  font-weight: 500;  
  font-size: 0.95rem;  
}
```

```
.profile-container {  
  display: flex;  
  align-items: center;  
  position: relative;  
  cursor: pointer;  
}
```

```
.profile-pic {  
  width: 40px;
```

```
height: 40px;  
border-radius: 50%;  
object-fit: cover;  
}
```

```
.dropdown-menu {  
  display: none;  
  position: absolute;  
  top: 100%;  
  right: 0;  
  background: white;  
  box-shadow: 0 2px 10px rgba(0,0,0,0.1);  
  border-radius: 8px;  
  padding: 8px 0;  
  min-width: 180px;  
  z-index: 1000;  
}
```

```
.profile-container:hover .dropdown-menu {  
  display: block;  
}
```

```
.dropdown-menu a {  
  display: flex;  
  align-items: center;  
  gap: 8px;  
  padding: 8px 16px;  
  color: #333;  
  text-decoration: none;  
  transition: background-color 0.2s;
```

```
}
```

```
.dropdown-menu a:hover {  
    background-color: #f5f5f5;  
}
```

```
</style>
```

```
<script>
```

```
// Show toast message if exists
```

```
<?php if (isset($message)): ?>
```

```
    document.addEventListener('DOMContentLoaded', function() {
```

```
        showToast('<?php echo addslashes($message); ?>', '<?php echo $message_type; ?>');
```

```
    });
```

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

```
function showToast(message, type = 'info') {
```

```
    const toast = document.getElementById('toast');
```

```
    const toastMessage = document.getElementById('toast-message');
```

```
    const toastTitle = toast.querySelector('.toast-title');
```

```
    const icon = toast.querySelector('i');
```

```
// Reset classes
```

```
toast.className = 'toast';
```

```
icon.className = 'fas';
```

```
// Set type-specific properties
```

```
switch(type) {
```

```
    case 'success':
```

```
        toastTitle.textContent = 'Success';
```

```
        icon.classList.add('fa-check-circle');
```



```
        toast.classList.add('success');

        break;

    case 'error':

        toastTitle.textContent = 'Error';

        icon.classList.add('fa-exclamation-circle');

        toast.classList.add('error');

        break;

    default:

        toastTitle.textContent = 'Information';

        icon.classList.add('fa-info-circle');

        toast.classList.add('info');

    }
}
```

```
toastMessage.textContent = message;

toast.classList.add('show');
```

```
// Hide toast after 3 seconds

setTimeout(() => {

    toast.classList.remove('show');

}, 3000);

}
```

```
</script>
```

```
</body>
```

```
</html>
```

Apply.php

```
<?php

    session_start();

    include '../database/connectdatabase.php';

    $dbname="project";
```

```

mysqli_select_db($conn,$dbname);

if (!isset($_SESSION['user_id']) || !isset($_GET['job_id']))
{
    mysqli_close($conn);
    $_SESSION['message'] = "Please login to apply for jobs.";
    $_SESSION['message_type'] = "error";
    header('location: ../login/loginvalidation.php');
    exit();
}

$user_id = $_SESSION['user_id'];
$job_id = $_GET['job_id'];

// Function to check if vacancy is filled
function isVacancyFilled($conn, $job_id) {
    $sql = "SELECT j.vacancy,
                (SELECT COUNT(*) FROM tbl_applications WHERE job_id = ? AND status
= 'accepted') AS total_accepted
                FROM tbl_jobs j
                WHERE j.job_id = ?";

    $stmt = mysqli_prepare($conn, $sql);
    mysqli_stmt_bind_param($stmt, "ii", $job_id, $job_id);
    mysqli_stmt_execute($stmt);
    $result = mysqli_stmt_get_result($stmt);
    $row = mysqli_fetch_assoc($result);

    return $row['total_accepted'] >= $row['vacancy'];
}

```

```

// First check if vacancy is filled
if (isVacancyFilled($conn, $job_id)) {
    mysqli_close($conn);
    $_SESSION['message'] = "This job has already reached its hiring limit. You cannot
apply.";
    $_SESSION['message_type'] = "error";
    header('Location: userdashboard.php');
    exit();
}

// Checking if the user has already applied for the job
$check = "SELECT * FROM tbl_applications WHERE user_id = ? AND job_id = ?";
$check_stmt = mysqli_prepare($conn, $check);
mysqli_stmt_bind_param($check_stmt, "ii", $user_id, $job_id);
mysqli_stmt_execute($check_stmt);
$check_result = mysqli_stmt_get_result($check_stmt);

if (mysqli_num_rows($check_result) > 0)
{
    mysqli_close($conn);
    $_SESSION['message'] = "You have already applied for this job.";
    $_SESSION['message_type'] = "error";
    header('Location: userdashboard.php');
    exit();
}
else
{
    // Check if any certificates or documents are required
    $certificate_check = "SELECT license_required, badge_required FROM tbl_jobs
WHERE job_id = ?";
    $cert_stmt = mysqli_prepare($conn, $certificate_check);

```

```

mysqli_stmt_bind_param($cert_stmt, "i", $job_id);
mysqli_stmt_execute($cert_stmt);
$cert_result = mysqli_stmt_get_result($cert_stmt);
$job = mysqli_fetch_assoc($cert_result);

if ($job)
{
    $license_required = $job['license_required'];
    $badge_required = $job['badge_required'];

    // Check if license or badge is required
    if ($license_required !== NULL && $license_required !== 'not_required' &&
$license_required !== "")
    {
        $_SESSION['message'] = "This job requires a " . $license_required . " license.
Please upload it in your profile.";
        $_SESSION['message_type'] = "error";
        header('Location: uploadcertificate.php?user_id=' . $user_id . '&job_id=' . $job_id);
        exit();
    }

    if ($badge_required !== NULL && $badge_required === 'yes')
    {
        $_SESSION['message'] = "This job requires a badge. Please upload it in your
profile.";
        $_SESSION['message_type'] = "error";
        header('Location: uploadcertificate.php?user_id=' . $user_id . '&job_id=' . $job_id);
        exit();
    }
}

```

```

// If no certificates required or all certificates are valid, proceed with application

$sql = "INSERT INTO tbl_applications (user_id, job_id, status) VALUES (?, ?,
'Pending')";

$apply_stmt = mysqli_prepare($conn, $sql);

mysqli_stmt_bind_param($apply_stmt, "ii", $user_id, $job_id);

if (mysqli_stmt_execute($apply_stmt))
{
    $_SESSION['message'] = "You have successfully applied for this job!";
    $_SESSION['message_type'] = "success";
    header('Location: userdashboard.php');
    exit();
}
else
{
    $_SESSION['message'] = "Failed to apply for the job. Please try again.";
    $_SESSION['message_type'] = "error";
    mysqli_close($conn);
    header('Location: userdashboard.php');
    exit();
}
}
?>

```

Employerdashboard.php

```

<?php
session_start();

if (!isset($_SESSION['employer_id'])) {
    header("Location: ../login/loginvalidation.php");
    exit();
}

```

```
}
```

```
include '../database/connectdatabase.php';
```

```
$dbname = "project";
```

```
mysqli_select_db($conn, $dbname);
```

```
// Your existing code to fetch employer data...
```

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

```
$sql = "SELECT u.company_name, l.email, u.profile_image
```

```
FROM tbl_login AS l
```

```
JOIN tbl_employer AS u ON l.employer_id = u.employer_id
```

```
WHERE u.employer_id = ?";
```

```
$stmt = mysqli_prepare($conn, $sql);
```

```
mysqli_stmt_bind_param($stmt, "i", $employer_id);
```

```
mysqli_stmt_execute($stmt);
```

```
$result = mysqli_stmt_get_result($stmt);
```

```
if ($result && mysqli_num_rows($result) > 0) {
```

```
    $employer_data = mysqli_fetch_array($result);
```

```
    $email = $employer_data['email'];
```

```
    $username = $employer_data['company_name'];
```

```
    $profile_image = $employer_data['profile_image'];
```

```
} else {
```

```
    error_log("Database error or user not found for ID: $employer_id");
```

```
    session_destroy();
```

```
    header("Location: ../login/logout.php");
```

```
    exit();
```

```
}
```

```
$sql_fetch = "SELECT j.*,
```

```
    COUNT(CASE WHEN a.status = 'applied' THEN 1 END) AS applied_count,
```

```

COUNT(CASE WHEN a.status = 'accepted' THEN 1 END) AS accepted_count,
COUNT(CASE WHEN a.status = 'rejected' THEN 1 END) AS rejected_count,
COUNT(a.id) AS total_applicants
FROM tbl_jobs j
LEFT JOIN tbl_applications a ON j.job_id = a.job_id
WHERE j.employer_id = ? AND j.is_deleted = 0
GROUP BY j.job_id";

$stmt_fetch = mysqli_prepare($conn, $sql_fetch);
mysqli_stmt_bind_param($stmt_fetch, "i", $employer_id);
mysqli_stmt_execute($stmt_fetch);
$result_fetch = mysqli_stmt_get_result($stmt_fetch);
mysqli_stmt_close($stmt_fetch);

$sql_count = "SELECT COUNT(*) AS active_jobs FROM tbl_jobs WHERE employer_id =
? AND is_deleted = 0";

$stmt_count = mysqli_prepare($conn, $sql_count);
mysqli_stmt_bind_param($stmt_count, "i", $employer_id);
mysqli_stmt_execute($stmt_count);
$result_count = mysqli_stmt_get_result($stmt_count);
$row = mysqli_fetch_assoc($result_count);
$active_jobs = $row ? $row['active_jobs'] : 0;
mysqli_stmt_close($stmt_count);

$sql_profile = "SELECT profile_image FROM tbl_employer WHERE employer_id = ?";
$stmt_profile = mysqli_prepare($conn, $sql_profile);
mysqli_stmt_bind_param($stmt_profile, "i", $employer_id);
mysqli_stmt_execute($stmt_profile);
$result_profile = mysqli_stmt_get_result($stmt_profile);
$profile_data = mysqli_fetch_assoc($result_profile);
$profile_image_path = !empty($profile_data['profile_image']) &&
file_exists("../database/profile_picture/" . $profile_data['profile_image'])

```

```
? "/mini project/database/profile_picture/" .  
htmlspecialchars($profile_data['profile_image'])  
: "../assets/images/company-logo.png";
```

```
$sql_applicants = "SELECT COUNT(DISTINCT a.id) as total_applicants  
FROM tbl_applications a  
JOIN tbl_jobs j ON a.job_id = j.job_id  
WHERE j.employer_id = ? AND j.is_deleted = 0";  
$stmt_applicants = mysqli_prepare($conn, $sql_applicants);  
mysqli_stmt_bind_param($stmt_applicants, "i", $employer_id);  
mysqli_stmt_execute($stmt_applicants);  
$result_applicants = mysqli_stmt_get_result($stmt_applicants);  
$applicants_data = mysqli_fetch_assoc($result_applicants);  
$total_applicants = $applicants_data ? $applicants_data['total_applicants'] : 0;  
mysqli_stmt_close($stmt_applicants);
```

```
$sql_total_applications = "SELECT COUNT(*) as total_applications  
FROM tbl_applications a  
JOIN tbl_jobs j ON a.job_id = j.job_id  
WHERE j.employer_id = ? AND j.is_deleted = 0";  
$stmt_total_applications = mysqli_prepare($conn, $sql_total_applications);  
mysqli_stmt_bind_param($stmt_total_applications, "i", $employer_id);  
mysqli_stmt_execute($stmt_total_applications);  
$result_total_applications = mysqli_stmt_get_result($stmt_total_applications);  
$total_applications_data = mysqli_fetch_assoc($result_total_applications);  
$total_applications = $total_applications_data ? $total_applications_data['total_applications']  
: 0;  
mysqli_stmt_close($stmt_total_applications);
```

```
$sql_active_applications = "SELECT COUNT(*) as active_applications  
FROM tbl_applications a
```



```

        JOIN tbl_jobs j ON a.job_id = j.job_id
        WHERE j.employer_id = ? AND j.is_deleted = 0
        AND a.status = 'applied';

$stmt_active_applications = mysqli_prepare($conn, $sql_active_applications);
mysqli_stmt_bind_param($stmt_active_applications, "i", $employer_id);
mysqli_stmt_execute($stmt_active_applications);
$result_active_applications = mysqli_stmt_get_result($stmt_active_applications);
$active_applications_data = mysqli_fetch_assoc($result_active_applications);
$active_applications = $active_applications_data ?
$active_applications_data['active_applications'] : 0;
mysqli_stmt_close($stmt_active_applications);

```

```

$sql_hired_users = "SELECT COUNT(DISTINCT a.id) as hired_users
        FROM tbl_applications a
        JOIN tbl_jobs j ON a.job_id = j.job_id
        WHERE j.employer_id = ? AND j.is_deleted = 0
        AND a.status = 'accepted';

```

```

$stmt_hired_users = mysqli_prepare($conn, $sql_hired_users);
mysqli_stmt_bind_param($stmt_hired_users, "i", $employer_id);
mysqli_stmt_execute($stmt_hired_users);
$result_hired_users = mysqli_stmt_get_result($stmt_hired_users);
$hired_users_data = mysqli_fetch_assoc($result_hired_users);
$hired_users = $hired_users_data ? $hired_users_data['hired_users'] : 0;
mysqli_stmt_close($stmt_hired_users);

```

```

$sql_pending_interviews = "SELECT COUNT(*) as pending_interviews
        FROM tbl_appointments
        WHERE employer_id = ?
        AND status = 'Pending';

```

```

$stmt_pending_interviews = mysqli_prepare($conn, $sql_pending_interviews);
mysqli_stmt_bind_param($stmt_pending_interviews, "i", $employer_id);

```

```
mysqli_stmt_execute($stmt_pending_interviews);

$result_pending_interviews = mysqli_stmt_get_result($stmt_pending_interviews);

$pending_interviews_data = mysqli_fetch_assoc($result_pending_interviews);

$pending_interviews = $pending_interviews_data ?
$pending_interviews_data['pending_interviews'] : 0;

mysqli_stmt_close($stmt_pending_interviews);
```

```
$sql_recent_applicants = "SELECT a.id, a.status, a.applied_at,
                             u.first_name, u.last_name, u.profile_image,
                             j.job_title
                             FROM tbl_applications a
                             JOIN tbl_user u ON a.user_id = u.user_id
                             JOIN tbl_jobs j ON a.job_id = j.job_id
                             WHERE j.employer_id = ? AND j.is_deleted = 0
                             ORDER BY a.applied_at DESC
                             LIMIT 5";
```

```
$stmt_recent_applicants = mysqli_prepare($conn, $sql_recent_applicants);
mysqli_stmt_bind_param($stmt_recent_applicants, "i", $employer_id);
mysqli_stmt_execute($stmt_recent_applicants);
$result_recent_applicants = mysqli_stmt_get_result($stmt_recent_applicants);
mysqli_stmt_close($stmt_recent_applicants);
?>
```

```
<!DOCTYPE html>
```

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

```
<head>
```

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

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

```
    <title>Employer Dashboard</title>
```

```
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.0/css/all.min.css">
```

```
<link  
href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&displ  
ay=swap" rel="stylesheet">
```

```
<link rel="stylesheet" href="employerdashboard.css">
```

```
<style>
```

```
/* Main container adjustments */
```

```
.dashboard-container {  
    display: flex;  
    min-height: 100vh;  
}
```

```
/* Main content area */
```

```
.main-container {  
    flex: 1;  
    margin-left: 280px; /* Sidebar width */  
    margin-top: 60px; /* Navbar height */  
    padding: 20px;  
    background: #f8f9fa;  
}
```

```
/* Content sections */
```

```
.main-content {  
    width: 100%;  
    max-width: 1200px;  
    margin: 0 auto;  
}
```

```
.header {  
    margin-bottom: 30px;  
}
```

```
/* Stats grid */  
.stats-grid {  
  display: grid;  
  grid-template-columns: repeat(auto-fit, minmax(240px, 1fr));  
  gap: 20px;  
  margin-bottom: 30px;  
}
```

```
/* Jobs section */  
.jobs-section {  
  background: white;  
  border-radius: 10px;  
  padding: 20px;  
  box-shadow: 0 2px 4px rgba(0,0,0,0.1);  
}
```

```
/* Applicants sidebar adjustments */  
.applicants-sidebar {  
  position: fixed;  
  right: 0;  
  top: 0; /* Changed to full height */  
  height: 100vh;  
  width: 300px;  
  background: white;  
  box-shadow: -2px 0 5px rgba(0,0,0,0.1);  
  transform: translateX(100%);  
  transition: transform 0.3s ease;  
  z-index: 1001;  
  overflow-y: auto;  
}
```

```
.applicants-sidebar.active {  
    transform: translateX(0);  
}
```

```
/* Overlay for blur effect */
```

```
.sidebar-overlay {  
    position: fixed;  
    top: 0;  
    left: 0;  
    right: 0;  
    bottom: 0;  
    background: rgba(0, 0, 0, 0.5);  
    backdrop-filter: blur(4px);  
    -webkit-backdrop-filter: blur(4px);  
    opacity: 0;  
    visibility: hidden;  
    transition: all 0.3s ease;  
    z-index: 1000;  
}
```

```
.sidebar-overlay.active {  
    opacity: 1;  
    visibility: visible;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<!-- <div class="dashboard-container">
```

```
<div class="sidebar">
```

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

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

        "

            onerror="this.src='../assets/images/company-logo.png';">

    <?php else: ?>

        

    <?php endif; ?>

</div>

<div class="company-info">

    <span><?php echo htmlspecialchars($username); ?></span>

    <span><?php echo htmlspecialchars($email); ?></span>

</div>

<nav class="nav-menu">

    <div class="nav-item active"><i class="fas fa-th-large"></i><a

href="employerdashboard.php">Dashboard</a></div>

    <div class="nav-item"><i class="fas fa-plus-circle"></i><a

href="postjob.php">Post a Job</a></div>

    <div class="nav-item"><i class="fas fa-briefcase"></i><a

href="myjoblist.php">My Jobs</a></div>

    <div class="nav-item"><i class="fas fa-users"></i><a

href="applicants.php">Applicants</a></div>

    <div class="nav-item"><i class="fas fa-calendar-check"></i><a

href="interviews.php">Interviews</a></div>

</nav>

<div class="settings-section">

    <div class="nav-item"><i class="fas fa-user-cog"></i><a

href="employer_profile.php">My Profile</a></div>

    <div class="nav-item"><i class="fas fa-sign-out-alt"></i><a

href="../login/logout.php">Logout</a></div>

</div>

</div> -->

<?php include 'sidebar.php'; ?>
```

```

<div class="main-container">

  <div class="main-content">

    <div class="header">

      <h1>Welcome, <?php echo htmlspecialchars($username); ?></h1>

      <div class="header-actions">

        <button id="applicantsToggle" class="toggle-sidebar-btn"><i class="fas fa-
users"></i> Applicants</button>

        <!-- <button class="post-job-btn"><i class="fas fa-plus-circle"></i><a
href="postjob.php">Post a Job</a></button> -->

      </div>

    </div>

    <div class="stats-grid">

      <div class="stat-card"><i class="fas fa-briefcase"></i><div><span><?php echo
$active_jobs; ?></span><p>Active Jobs</p></div></div>

      <div class="stat-card"><i class="fas fa-file-alt"></i><div><span><?php echo
$total_applications; ?></span><p>Total Applications</p></div></div>

      <div class="stat-card"><i class="fas fa-user-check"></i><div><span><?php
echo $hired_users; ?></span><p>Hired Users</p></div></div>

      <div class="stat-card"><i class="fas fa-calendar-check"></i><div><span><?php
echo $pending_interviews; ?></span><p>Pending Interviews</p></div></div>

    </div>

    <div class="jobs-section">

      <div class="search-container">

        <div class="search-box">

          <i class="fas fa-search"></i>

          <input type="text" placeholder="Search jobs..." id="search" name="search"
oninput="filterJobs()">

          <button class="reset-button" onclick="resetFilters()" title="Reset Search">

            <i class="fas fa-times"></i>

          </button>

        </div>

      </div>

    </div>

  </div>

</div>

```

```

        </div>
    </div>

    <h2><i class="fas fa-list"></i> Recent Job Postings</h2>

    <div class="job-card-container">

        <?php if (mysqli_num_rows($result_fetch) > 0): ?>
            <?php while ($job_data = mysqli_fetch_array($result_fetch)): ?>
                <div class="job-card">
                    <div class="job-info">
                        <div class="company-logo"><i class="fas fa-briefcase"></i></div>
                        <div class="job-details">
                            <h3><?php echo htmlspecialchars($job_data['job_title']);
?></h3>

                            <div class="job-meta">
                                <span><i class="fas fa-map-marker-alt"></i> <?php echo
htmlspecialchars($job_data['location']); ?></span>

                                <span><i class="fas fa-calendar-alt"></i> <?php echo
htmlspecialchars($job_data['vacancy_date']); ?></span>

                                <span><i class="fas fa-rupee-sign"></i> <?php echo
htmlspecialchars($job_data['salary']); ?></span>

                            </div>

                            <p class="job-description"><?php echo
substr(htmlspecialchars($job_data['job_description']), 0, 150) . '...'; ?></p>

                            <div class="job-stats">
                                <span><i class="fas fa-users"></i> <?php echo
$job_data['total_applicants']; ?> Applicants</span>

                                <span><i class="fas fa-check-circle"></i> <?php echo
$job_data['accepted_count']; ?> Accepted</span>

                                <span><i class="fas fa-times-circle"></i> <?php echo
$job_data['rejected_count']; ?> Rejected</span>

                            </div>

                        </div>
                    </div>

                    <div class="job-actions">

```



```
        <a href="editjob.php?job_id=?php echo $job_data['job_id']; ?>"
class="action-btn edit-btn"><i class="fas fa-edit"></i> Edit</a>
```

```
        <!-- <a href="deletejob.php?id=?php echo $job_data['job_id'];
?>&action=deactivate" class="action-btn delete-btn" onclick="return confirm('Are you sure
you want to delete this job?')"><i class="fas fa-trash-alt"></i> Deactivate</a> -->
```

```
    </div>
```

```
    </div>
```

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

```
<?php else: ?>
```

```
<div class="no-jobs">
```

```
<i class="fas fa-briefcase"></i>
```

```
<h3>No jobs posted yet</h3>
```

```
<p>Start by posting your first job opening</p>
```

```
<!-- <button class="post-job-btn"><i class="fas fa-plus-circle"></i><a
href="postjob.php">Post a Job</a></button> -->
```

```
</div>
```

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

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<!-- Add overlay div before the sidebar -->
```

```
<div class="sidebar-overlay" id="sidebarOverlay"></div>
```

```
<div class="applicants-sidebar" id="applicantsSidebar">
```

```
<button id="closeSidebar"><i class="fas fa-times"></i></button>
```

```
<h2>Recent Applicants <span class="applicant-count"><?php echo
$total_applications; ?></span></h2>
```

```
<div class="applicant-list">
```

```
<?php if (mysqli_num_rows($result_recent_applicants) > 0): ?>
```

```
<?php while ($applicant = mysqli_fetch_assoc($result_recent_applicants)): ?>
```

```

<div class="applicant-group">

    <h3><?php echo htmlspecialchars($applicant['job_title']); ?></h3>

    <div class="applicant-card">

        <div class="applicant-avatar">

            <?php if (!empty($applicant['profile_image'])): ?>

                "

                    onerror="this.style.display='none';
this.parentElement.innerHTML='<?php echo substr($applicant['first_name'], 0, 1) .
substr($applicant['last_name'], 0, 1); ?>';">

                <?php else: ?>

                    <?php echo substr($applicant['first_name'], 0, 1) .
substr($applicant['last_name'], 0, 1); ?>

                <?php endif; ?>

            </div>

            <div>

                <p class="applicant-name"><?php echo
htmlspecialchars($applicant['first_name']) . ' ' . $applicant['last_name']; ?></p>

                <p class="applicant-position"><?php echo
htmlspecialchars($applicant['job_title']); ?></p>

                <p class="applicant-status">Status: <?php echo
htmlspecialchars($applicant['status']); ?></p>

                <p class="applicant-date">Applied: <?php echo date('M d, Y',
strtotime($applicant['applied_at'])); ?></p>

            </div>

        </div>

    </div>

<?php endwhile; ?>

<?php else: ?>

    <div class="no-applicants">

        <i class="fas fa-users"></i>

        <p>No recent applicants</p>

```

```

        </div>

        <?php endif; ?>

        <a href="applicants.php">View All Applicants <i class="fas fa-arrow-
right"></i></a>

    </div>

</div>

</div>

<script src="employerdashboard.js"></script>

<script>

    document.addEventListener('DOMContentLoaded', function() {

        const applicantsToggle = document.getElementById('applicantsToggle');
        const applicantsSidebar = document.getElementById('applicantsSidebar');
        const sidebarOverlay = document.getElementById('sidebarOverlay');
        const closeSidebar = document.getElementById('closeSidebar');

        function openSidebar() {

            applicantsSidebar.classList.add('active');
            sidebarOverlay.classList.add('active');
            document.body.style.overflow = 'hidden'; // Prevent scrolling
        }

        function closeSidebarFunc() {

            applicantsSidebar.classList.remove('active');
            sidebarOverlay.classList.remove('active');
            document.body.style.overflow = ""; // Restore scrolling
        }

        if (applicantsToggle) {

            applicantsToggle.addEventListener('click', openSidebar);
        }
    }

```

```
        if (closeSidebar) {
            closeSidebar.addEventListener('click', closeSidebarFunc);
        }

        // Close on overlay click
        if (sidebarOverlay) {
            sidebarOverlay.addEventListener('click', closeSidebarFunc);
        }

        // Close on escape key
        document.addEventListener('keydown', function(e) {
            if (e.key === 'Escape') {
                closeSidebarFunc();
            }
        });
    });
</script>
</body>
</html>
```

Postjob.php

```
<?php
    session_start();
    $employer_id = $_SESSION['employer_id'];
    include '../database/connectdatabase.php';
    $dbname = "project";
    mysqli_select_db($conn, $dbname);
    $error = "";

    if (!isset($_SESSION['employer_id']))
```

```
{  
    header("Location: ../login/loginvalidation.php");  
    exit();  
}
```

```
$sql = "SELECT u.company_name, l.email, u.profile_image  
FROM tbl_login AS l  
JOIN tbl_employer AS u ON l.employer_id = u.employer_id  
WHERE u.employer_id = ?";
```

```
$stmt = mysqli_prepare($conn, $sql);  
mysqli_stmt_bind_param($stmt, "i", $employer_id);  
mysqli_stmt_execute($stmt);  
$result = mysqli_stmt_get_result($stmt);
```

```
if ($result && mysqli_num_rows($result) > 0)  
{  
    $employer_data = mysqli_fetch_array($result);  
    $email = $employer_data['email'];  
    $username = $employer_data['company_name'];  
    $profile_image = $employer_data['profile_image'];  
}
```

```
if ($_SERVER['REQUEST_METHOD'] == "POST")  
{  
    $category = $_POST['category'];
```

```
// Handle license_required - set to NULL if empty
```

```
$license = !empty($_POST['license_required']) ? $_POST['license_required'] : NULL;
```

```

// Handle badge_required - set to NULL if empty
$badge = !empty($_POST['badge_required']) ? $_POST['badge_required'] : NULL;

$job_title = $_POST['job_title'];
$job_description = $_POST['job_description'];
$location = $_POST['location'];
$town = $_POST['town'];
$vacancy_date = $_POST['date'];
$start_time = $_POST['start_time'];
$end_time = $_POST['end_time'];
$working_days = $_POST['working_days'];
$vacancy = $_POST['vacancy'];
$salary = $_POST['salary'];
$application_deadline = $_POST['last_date'];
$contact = $_POST['phone'];
$interview = isset($_POST['interview']) ? $_POST['interview'] : NULL;
$status = 'pending'; // Set initial status as pending

// Update the SQL statement to handle NULL values
$stmt = $conn->prepare("INSERT INTO tbl_jobs (category, job_title, location,
job_description, vacancy_date, vacancy, salary, application_deadline, interview, employer_id,
town, start_time, end_time, working_days, contact_no, license_required, badge_required,
status, is_deleted) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, 0)");

// Fix the bind_param statement to match the number of parameters
$stmt->bind_param("ssssssssississ",
    $category,
    $job_title,
    $location,
    $job_description,
    $vacancy_date,

```

```

        $vacancy,
        $salary,
        $application_deadline,
        $interview,
        $employer_id,
        $town,
        $start_time,
        $end_time,
        $working_days,
        $contact,
        $license,
        $badge,
        $status
    );

    if ($stmt->execute()) {
        $_SESSION['message'] = "Job posted successfully and is pending admin approval.";
        $_SESSION['message_type'] = "success";
        header("Location: postjob.php");
        exit();
    } else {
        $_SESSION['message'] = "Error posting job: " . $conn->error;
        $_SESSION['message_type'] = "error";
        header("Location: postjob.php");
        exit();
    }

    if (isset($stmt) && $stmt !== false)
    {
        $stmt->close();
    }

```

```

    }

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Post Job</title>

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.0/css/all.min.css">

    <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&displ
ay=swap" rel="stylesheet">

    <link rel="stylesheet" href="employerdashboard.css">

</head>

<body>

    <div class="toast-container" id="toastContainer"></div>

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

    <!-- Main Content -->

    <div class="main-container">

        <div class="header">

            <h1>Post a New Job - <?php echo htmlspecialchars($username); ?></h1>

        </div>

        <!-- Form Container -->

        <div class="content-card">

            <form method="post" id="add_job" onsubmit="return validateForm()">

                <?php if (isset($_GET['message'])): ?>

                    <div class="alert <?php echo strpos($_GET['message'], 'successfully') !==
false ? 'alert-success' : 'alert-error'; ?>">

```



```

        <?php echo htmlspecialchars($_GET['message']); ?>
    </div>

    <?php endif; ?>

    <div class="form-container">
        <!-- Basic Job Information -->
        <div class="form-section">
            <h3 class="section-title">Basic Information</h3>
            <div class="form-grid">
                <div class="form-group">
                    <label>Job Category</label>
                    <select name="category" id="category"
onchange="validatecategory()">
                        <option value="">Select Category</option>
                        <option value="Delivery and logistics">Delivery &
Logistics</option>
                        <option value="Hospitality and catering">Hospitality &
Catering</option>
                        <option value="Housekeeping and cleaning">Housekeeping &
Cleaning</option>
                        <option value="Retail and store jobs">Retail & Store Jobs</option>
                        <option value="Warehouse and factory jobs">Warehouse & Factory
Jobs</option>
                        <option value="Maintenance">Maintenance</option>
                    </select>
                    <p id="categoryerror" class="error"></p>
                </div>

                <div class="form-group">
                    <label>Job Title</label>
                    <input type="text" name="job_title" id="job_title" placeholder="Enter
job title" onkeyup="validateJobTitle()">

```

```
<p id="titleerror" class="error"></p>
</div>
</div>
</div>

<!-- Delivery Documents Section -->
<div id="delivery-docs" class="form-section" style="display: none;">
  <h3 class="section-title">Required Documents</h3>
  <div class="form-grid">
    <div class="form-group">
      <label>Driving License Required</label>
      <select name="license_required" id="license_required">
        <option value="">Select License Type</option>
        <option value="two_wheeler">Two Wheeler License</option>
        <option value="four_wheeler">Four Wheeler License</option>
        <option value="both">Both Required</option>
        <option value="not_required">Not Required</option>
      </select>
      <p id="licenseerror" class="error"></p>
    </div>

    <div class="form-group">
      <label>Badge Required</label>
      <select name="badge_required" id="badge_required">
        <option value="">Select Badge Requirement</option>
        <option value="yes">Yes</option>
        <option value="no">No</option>
      </select>
      <p id="badgeerror" class="error"></p>
    </div>
  </div>
</div>
```

</div>

</div>

<!-- Job Description -->

<div class="form-section">

<h3 class="section-title">Job Description</h3>

<div class="form-group">

<textarea name="job_description" id="job_description"
placeholder="Enter detailed job description"
onkeyup="validateJobDescription()"></textarea>

<p id="descriptionerror" class="error"></p>

</div>

</div>

<!-- Location Information -->

<div class="form-section">

<h3 class="section-title">Location Details</h3>

<div class="form-grid">

<div class="form-group">

<label>District</label>

<select name="location" id="location" onchange="showTowns()">

<option value="">Select District</option>

<option

value="Thiruvananthapuram">Thiruvananthapuram</option>

<option value="Kollam">Kollam</option>

<option value="Pathanamthitta">Pathanamthitta</option>

<option value="Alappuzha">Alappuzha</option>

<option value="Kottayam">Kottayam</option>

<option value="Idukki">Idukki</option>

<option value="Ernakulam">Ernakulam</option>

<option value="Thrissur">Thrissur</option>

```
<option value="Palakkad">Palakkad</option>
<option value="Malappuram">Malappuram</option>
<option value="Kozhikode">Kozhikode</option>
<option value="Wayanad">Wayanad</option>
<option value="Kannur">Kannur</option>
<option value="Kasaragod">Kasaragod</option>
</select>
<p id="locationerror" class="error"></p>
</div>
```

```
<div class="form-group">
  <label id="townLabel" style="display:none;">Town</label>
  <select name="town" id="tvm_towns" style="display:none;">
    <option value="">Select Town</option>
  </select>
  <p id="townerror" class="error"></p>
</div>
</div>
</div>
```

```
<!-- Schedule Information -->
<div class="form-section">
  <h3 class="section-title">Work Schedule</h3>
  <div class="form-grid">
    <div class="form-group">
      <label>Vacancy Date</label>
      <input type="date" name="date" id="date"
onchange="validateDate()">
      <p id="dateerror" class="error"></p>
    </div>
```

```
<div class="form-group">
  <label>Working Hours</label>
  <div class="time-inputs">
    <select id="start-time" name="start_time">
      <option value="">Start Time</option>
    </select>
    <span class="time-separator">to</span>
    <select id="end-time" name="end_time">
      <option value="">End Time</option>
    </select>
  </div>
  <p id="timeerror" class="error"></p>
</div>
```

```
<div class="form-group">
  <label>Working Days</label>
  <select name="working_days" id="working_days"
onchange="validateWorkingDays()">
    <option value="">Select Working Days</option>
    <option value="part_time">Part Time</option>
    <option value="full_day">Full Day</option>
    <option value="shift">Shift-based</option>
  </select>
  <p id="workingdayerror" class="error"></p>
</div>
</div>
```

```
<!-- Job Details -->
<div class="form-section">
  <h3 class="section-title">Additional Details</h3>
```

```
<div class="form-grid">

  <div class="form-group">

    <label>Number of Vacancies</label>

    <input type="text" name="vacancy" id="vacancy" placeholder="Enter
number of vacancies" onkeyup="validateVacancy()">

    <p id="vacancyerror" class="error"></p>
  </div>

  <div class="form-group">

    <label>Salary</label>

    <input type="text" name="salary" id="salary" placeholder="Enter
salary" onkeyup="validateSalary()">

    <p id="salaryerror" class="error"></p>
  </div>

  <div class="form-group">

    <label>Application Deadline</label>

    <input type="date" name="last_date" id="last_date"
onchange="validateLastDate()">

    <p id="lastdateerror" class="error"></p>
  </div>

  <div class="form-group">

    <label>Contact Number</label>

    <input type="text" name="phone" id="phone" placeholder="Enter
contact number" onkeyup="validatePhone()">

    <p id="phoneerror" class="error"></p>
  </div>
</div>

<!-- Interview Option -->
```

```
<div class="form-section">

  <h3 class="section-title">Interview Preference</h3>

  <div class="radio-group">

    <label>Interview Required?</label>

    <div class="radio-options">

      <div class="radio-item">

        <input type="radio" id="yes" name="interview" value="yes">

        <label for="yes">Yes</label>

      </div>

      <div class="radio-item">

        <input type="radio" id="no" name="interview" value="no">

        <label for="no">No</label>

      </div>

    </div>

    <div>

      <p id="interviewerror" class="error"></p>

    </div>

  </div>

  <!-- Submit Button -->

  <div class="form-section submit-section">

    <button type="submit" class="submit-btn">Post Job</button>

  </div>

</div>

</form>

</div>

</div>

</div>
```

```
<style>
```

```
/* Main container adjustments */
```

```
.main-container {  
  margin-left: 280px; /* Sidebar width */  
  margin-top: 60px; /* Navbar height */  
  padding: 20px;  
  min-height: calc(100vh - 60px);  
  background: #f8f9fa;  
}
```

```
/* Content card adjustments */  
.content-card {  
  background: #fff;  
  border-radius: 12px;  
  box-shadow: 0 2px 4px rgba(0,0,0,0.1);  
  padding: 25px;  
  margin-bottom: 20px;  
}
```

```
/* Header adjustments */  
.header {  
  margin-bottom: 25px;  
}
```

```
.header h1 {  
  font-size: 24px;  
  color: #333;  
  margin: 0;  
}
```

```
/* Form container adjustments */  
.form-container {
```



```
    max-width: 1200px;
    margin: 0 auto;
}
```

```
/* Professional Form Styling */
```

```
.content-card {
    background: #f8fafc;
    padding: 30px;
    border-radius: 16px;
}
```

```
.form-container {
    max-width: 1000px;
    margin: 0 auto;
    padding: 20px;
}
```

```
.form-section {
    background: white;
    border-radius: 12px;
    box-shadow: 0 1px 3px rgba(0, 0, 0, 0.05), 0 1px 2px rgba(0, 0, 0, 0.1);
    padding: 28px;
    margin-bottom: 28px;
    border: 1px solid #e2e8f0;
    transition: transform 0.2s ease, box-shadow 0.2s ease;
}
```

```
.form-section:hover {
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.05), 0 2px 4px rgba(0, 0, 0, 0.1);
    transform: translateY(-2px);
}
```

```
}
```

```
.section-title {  
  color: #1e293b;  
  font-size: 1.25rem;  
  font-weight: 600;  
  margin-bottom: 24px;  
  padding-bottom: 12px;  
  border-bottom: 2px solid #f1f5f9;  
  display: flex;  
  align-items: center;  
  gap: 10px;  
}
```

```
.section-title::before {  
  content: " ";  
  width: 4px;  
  height: 24px;  
  background: #3b82f6;  
  border-radius: 2px;  
  display: inline-block;  
}
```

```
.form-grid {  
  display: grid;  
  grid-template-columns: repeat(auto-fit, minmax(280px, 1fr));  
  gap: 24px;  
}
```

```
.form-group {
```

```
    margin-bottom: 20px;
}
```

```
label {
    display: block;
    font-weight: 500;
    color: #475569;
    margin-bottom: 8px;
    font-size: 0.95rem;
}
```

```
input, select, textarea {
    width: 100%;
    padding: 12px 16px;
    border: 1.5px solid #e2e8f0;
    border-radius: 8px;
    font-size: 0.95rem;
    color: #1e293b;
    background-color: #fff;
    transition: all 0.2s ease;
}
```

```
input:hover, select:hover, textarea:hover {
    border-color: #cbd5e1;
}
```

```
input:focus, select:focus, textarea:focus {
    border-color: #3b82f6;
    box-shadow: 0 0 0 3px rgba(59, 130, 246, 0.1);
    outline: none;
}
```

```
}
```

```
input::placeholder, select::placeholder, textarea::placeholder {  
  color: #94a3b8;  
}
```

```
textarea {  
  min-height: 120px;  
  resize: vertical;  
  line-height: 1.5;  
}
```

```
.time-inputs {  
  display: flex;  
  gap: 16px;  
  align-items: center;  
}
```

```
.time-inputs select {  
  flex: 1;  
}
```

```
.time-separator {  
  color: #64748b;  
  font-weight: 500;  
  margin: 0 4px;  
}
```

```
.radio-group {  
  padding: 12px 0;
```

```
}
```

```
.radio-options {  
  display: flex;  
  gap: 32px;  
  margin-top: 12px;  
}
```

```
.radio-item {  
  display: flex;  
  align-items: center;  
  gap: 8px;  
  cursor: pointer;  
}
```

```
.radio-item input[type="radio"] {  
  width: 20px;  
  height: 20px;  
  margin: 0;  
  cursor: pointer;  
  border: 2px solid #cbd5e1;  
}
```

```
.radio-item input[type="radio"]:checked {  
  border-color: #3b82f6;  
  background-color: #3b82f6;  
}
```

```
.radio-item label {  
  margin: 0;
```

```
    cursor: pointer;
    user-select: none;
}
```

```
.submit-section {
    text-align: center;
    border: none;
    box-shadow: none;
    padding-top: 0;
}
```

```
.submit-btn {
    background-color: #3b82f6;
    color: white;
    padding: 14px 36px;
    border: none;
    border-radius: 8px;
    font-weight: 600;
    font-size: 1rem;
    cursor: pointer;
    transition: all 0.3s ease;
    box-shadow: 0 2px 4px rgba(59, 130, 246, 0.1);
}
```

```
.submit-btn:hover {
    background-color: #2563eb;
    transform: translateY(-1px);
    box-shadow: 0 4px 6px rgba(59, 130, 246, 0.2);
}
```

```
.submit-btn:active {  
  transform: translateY(0);  
}
```

```
.error {  
  color: #dc2626;  
  font-size: 0.85rem;  
  margin-top: 6px;  
  display: flex;  
  align-items: center;  
}
```

```
.alert {  
  padding: 16px 20px;  
  border-radius: 10px;  
  margin-bottom: 28px;  
  font-weight: 500;  
  display: flex;  
  align-items: center;  
  gap: 12px;  
}
```

```
.alert::before {  
  font-family: "Font Awesome 5 Free";  
  font-weight: 900;  
  font-size: 1.1rem;  
}
```

```
.alert-success {  
  background-color: #f0fdf4;
```

```
        color: #166534;
        border: 1px solid #bbf7d0;
    }

.alert-success::before {
    content: '\f00c';
    color: #22c55e;
}

.alert-error {
    background-color: #fef2f2;
    color: #991b1b;
    border: 1px solid #fecaca;
}

.alert-error::before {
    content: '\f071';
    color: #ef4444;
}

/* Responsive Design */
@media (max-width: 768px) {
    .content-card {
        padding: 20px;
    }

    .form-container {
        padding: 10px;
    }
}
```



```
.form-section {  
  padding: 20px;  
}  
  
.form-grid {  
  grid-template-columns: 1fr;  
  gap: 16px;  
}  
  
.time-inputs {  
  flex-direction: column;  
}  
  
.time-separator {  
  display: none;  
}  
  
.submit-btn {  
  width: 100%;  
  padding: 16px;  
}  
  
.radio-options {  
  flex-direction: column;  
  gap: 16px;  
}  
}  
  
/* Optional: Add smooth scrolling for better UX */  
html {
```

```
    scroll-behavior: smooth;
}

/* Optional: Custom scrollbar for modern browsers */
::-webkit-scrollbar {
    width: 12px;
}

::-webkit-scrollbar-track {
    background: #f1f5f9;
}

::-webkit-scrollbar-thumb {
    background: #cbd5e1;
    border-radius: 6px;
    border: 3px solid #f1f5f9;
}

::-webkit-scrollbar-thumb:hover {
    background: #94a3b8;
}

/* Add this to your existing style section */
.toast-container {
    position: fixed;
    top: 20px;
    right: 20px;
    z-index: 9999;
}
```

```
.toast {  
  background: white;  
  border-radius: 10px;  
  padding: 16px 20px;  
  margin-bottom: 10px;  
  box-shadow: 0 4px 12px rgba(0, 0, 0, 0.15);  
  display: flex;  
  align-items: center;  
  gap: 12px;  
  min-width: 300px;  
  max-width: 400px;  
  animation: slideIn 0.3s ease forwards;  
  border-left: 4px solid #4CAF50;  
}
```

```
.toast.error {  
  border-left: 4px solid #dc2626;  
}
```

```
.toast i {  
  font-size: 24px;  
  color: #4CAF50;  
}
```

```
.toast.error i {  
  color: #dc2626;  
}
```

```
.toast-message {  
  color: #1e293b;
```

```
font-size: 15px;  
font-weight: 500;  
}
```

```
@keyframes slideIn {  
  from {  
    transform: translateX(100%);  
    opacity: 0;  
  }  
  to {  
    transform: translateX(0);  
    opacity: 1;  
  }  
}
```

```
@keyframes fadeOut {  
  from {  
    transform: translateX(0);  
    opacity: 1;  
  }  
  to {  
    transform: translateX(100%);  
    opacity: 0;  
  }  
}
```

```
</style>
```

```
<!-- Keep your existing scripts -->
```

```
<script src="postjob.js"></script>
```

```
<script>
```

```
// Auto-hide alert messages after 3 seconds

document.addEventListener('DOMContentLoaded', function() {

  const alert = document.querySelector('.alert');

  if (alert) {

    setTimeout(function() {

      alert.style.transition = 'opacity 0.5s ease';

      alert.style.opacity = '0';

      setTimeout(function() {

        alert.remove();

      }, 500);

    }, 3000);

  }

});

function showToast(message, type = 'success') {

  const toastContainer = document.getElementById('toastContainer');

  // Create toast element

  const toast = document.createElement('div');

  toast.className = `toast ${type}`;

  // Add icon based on type

  const icon = type === 'success' ? 'check-circle' : 'exclamation-circle';

  toast.innerHTML = `

    <i class="fas fa-${icon}"></i>

    <span class="toast-message">${message}</span>

  `;

  // Add toast to container
```

```

toastContainer.appendChild(toast);

// Remove toast after 3 seconds
setTimeout(() => {
    toast.style.animation = 'fadeOut 0.3s ease forwards';
    setTimeout(() => {
        toast.remove();
    }, 300);
}, 3000);
}

// Show notification if there's a message in session
<?php if(isset($_SESSION['message'])): ?>
    showToast(
        '<?php echo addslashes($_SESSION['message']); ?>',
        '<?php echo $_SESSION['message_type'] ?? "success"; ?>'
    );
<?php
    // Clear the message after displaying
    unset($_SESSION['message']);
    unset($_SESSION['message_type']);
?>
<?php endif; ?>
</script>
</body>
</html>

```

Postjob.js

```

// Category validation

```

```
function validatecategory() {  
    const category = document.getElementById('category').value;  
    const error = document.getElementById('categoryerror');  
    const deliveryDocs = document.getElementById('delivery-docs');  
  
    if (!category) {  
        error.textContent = "Category is required";  
        return false;  
    }  
  
    if (category === "Delivery and logistics") {  
        deliveryDocs.style.display = "block";  
    } else {  
        deliveryDocs.style.display = "none";  
        document.getElementById('license_required').value = "";  
        document.getElementById('badge_required').value = "";  
    }  
  
    error.textContent = "";  
    return true;  
}  
  
// Job Title validation  
function validateJobTitle() {  
    const jobTitle = document.getElementById('job_title').value.trim();  
    const error = document.getElementById("titleerror");  
  
    if (!jobTitle) {  
        error.textContent = "Job title is required.";  
        return false;  
    }  
}
```

```
}  
if (jobTitle.length < 3 || jobTitle.length > 50) {  
    error.textContent = "Job title must be between 3 and 50 characters.";   
    return false;  
}  
if (!/^[a-zA-Z\s]+$/.test(jobTitle)) {  
    error.textContent = "Only letters and spaces are allowed";  
    return false;  
}  
  
error.textContent = "";  
return true;  
}  
  
// Job Description validation  
function validateJobDescription() {  
    const jobDescription = document.getElementById('job_description').value.trim();  
    const error = document.getElementById("descriptionerror");  
  
    if (!jobDescription) {  
        error.textContent = "Job description is required.";   
        return false;  
    }  
    if (jobDescription.length < 5 || jobDescription.length > 250) {  
        error.textContent = "Job description must be between 5 and 250 characters.";   
        return false;  
    }  
    if (!/^[a-zA-Z0-9\s.,!?-]+$/.test(jobDescription)) {  
        error.textContent = "Special characters are not allowed";  
        return false;  
    }  
}
```



```

    }

    error.textContent = "";

    return true;
}

// Location validation
function validateLocation() {
    const location = document.getElementById('location').value;
    const error = document.getElementById("locationerror");

    if (!location) {
        error.textContent = "Please select a location";
        return false;
    }

    error.textContent = "";
    return true;
}

// Town data and selection functionality
const districtTowns = {
    'Thiruvananthapuram': ['Thiruvananthapuram', 'Neyyattinkara', 'Attingal', 'Varkala',
        'Nedumangad', 'Kazhakkootam', 'Kallambalam', 'Kovalam', 'Balaramapuram', 'Pothencode'],
    'Kollam': ['Kollam', 'Paravur', 'Punalur', 'Karunagappally', 'Kottarakkara', 'Chavara', 'Kundara',
        'Anchal', 'Oachira', 'Sasthamkotta'],
    'Pathanamthitta': ['Pathanamthitta', 'Adoor', 'Thiruvalla', 'Ranni', 'Pandalam', 'Konni',
        'Mallappally', 'Kozhencherry', 'Mannarakulanji', 'Seethathode'],
    'Alappuzha': ['Alappuzha', 'Chengannur', 'Kayamkulam', 'Mavelikkara', 'Haripad', 'Cherthala',
        'Ambalappuzha', 'Thakazhi', 'Mannar', 'Edathua'],
    'Kottayam': ['Kottayam', 'Pala', 'Changanassery', 'Vaikom', 'Ettumanoor', 'Erattupetta',
        'Kuravilangad', 'Kanjirappally', 'Pampady', 'Kidangoor'],

```

'Idukki': ['Thodupuzha', 'Munnar', 'Adimali', 'Kumily', 'Kattappana', 'Nedumkandam',
'Vagamon', 'Devikulam', 'Peermade', 'Udumalpettai'],

'Ernakulam': ['Kochi', 'Aluva', 'Angamaly', 'Muvattupuzha', 'Perumbavoor', 'Kothamangalam',
'North Paravur', 'Kakkanad', 'Piravom', 'Kaloor'],

'Thrissur': ['Thrissur', 'Chalakudy', 'Kodungallur', 'Irinjalakuda', 'Guruvayur', 'Kunnamkulam',
'Wadakkanchery', 'Pavaratty', 'Kecheri', 'Mannuthy'],

'Palakkad': ['Palakkad', 'Ottapalam', 'Chittur', 'Pattambi', 'Shoranur', 'Mannarkkad', 'Alathur',
'Nemmara', 'Cherpulassery', 'Kongad'],

'Malappuram': ['Malappuram', 'Tirur', 'Ponnani', 'Manjeri', 'Perinthalmanna', 'Nilambur',
'Kottakkal', 'Parappanangadi', 'Edappal', 'Kondotty'],

'Kozhikode': ['Kozhikode', 'Vadakara', 'Koyilandy', 'Ramanattukara', 'Feroke', 'Koduvally',
'Balussery', 'Mavoor', 'Chelannur', 'Thamarassery'],

'Wayanad': ['Kalpetta', 'Sulthan Bathery', 'Mananthavady', 'Meenangadi', 'Panamaram',
'Vythiri', 'Pulpally', 'Ambalavayal', 'Muttill', 'Thariode'],

'Kannur': ['Kannur', 'Thalassery', 'Payyanur', 'Mattannur', 'Iritty', 'Koothuparamba',
'Taliparamba', 'Kuthuparamba', 'Panoor', 'Chirakkal'],

'Kasaragod': ['Kasaragod', 'Kanhangad', 'Nileshwar', 'Cheruvathur', 'Uppala', 'Manjeshwar',
'Periya', 'Hosdurg', 'Bekal', 'Mogral Puthur']
};

```
function showTowns() {  
    const selectedDistrict = document.getElementById('location').value;  
    const townSelect = document.getElementById('tvm_towns');  
    const townLabel = document.getElementById('townLabel');  
  
    // Clear existing options  
    townSelect.innerHTML = '<option value="">Select Town</option>';  
  
    if (selectedDistrict) {  
        // Show town dropdown and label  
        townSelect.style.display = 'block';  
        townLabel.style.display = 'block';  
  
        // Get towns for selected district
```

```
const towns = districtTowns[selectedDistrict] || [];

// Add town options
towns.forEach(town => {
    const option = document.createElement('option');
    option.value = town;
    option.textContent = town;
    townSelect.appendChild(option);
});
} else {
    // Hide town dropdown and label if no district is selected
    townSelect.style.display = 'none';
    townLabel.style.display = 'none';
}
}

// Vacancy Date validation
function validateDate() {
    const dateInput = document.getElementById('date').value;
    const error = document.getElementById('dateerror');

    if (!dateInput) {
        error.textContent = "Vacancy date is required.";
        return false;
    }

    const selectedDate = new Date(dateInput);
    const today = new Date();
    today.setHours(0, 0, 0, 0);
    selectedDate.setHours(0, 0, 0, 0);
```

```
if (selectedDate < today) {  
    error.textContent = "Vacancy date cannot be in the past.";   
    return false;  
}
```

```
const oneMonthFromToday = new Date(today);  
oneMonthFromToday.setMonth(today.getMonth() + 1);
```

```
if (selectedDate > oneMonthFromToday) {  
    error.textContent = "Vacancy date must be one month from today.";   
    return false;  
}
```

```
error.textContent = "";  
return true;  
}
```

// Time Selection validation

```
function validateTimeSelection() {  
    let startTime = document.getElementById("start-time").value;  
    let endTime = document.getElementById("end-time").value;  
    let error = document.getElementById("timeerror");  
  
    if (!startTime || !endTime) {  
        error.textContent = "Both start and end time are required.";   
        return false;  
    }  
  
    let startMinutes = convertTimeToMinutes(startTime);
```

```
let endMinutes = convertTimeToMinutes(endTime);
```

```
if (endMinutes <= startMinutes) {  
    error.textContent = "End time must be later than start time.";  
    return false;  
}
```

```
error.textContent = "";  
return true;  
}
```

```
function convertTimeToMinutes(time) {  
    let [hours, minutes] = time.split(':').map(Number);  
    return hours * 60 + minutes;  
}
```

```
// Vacancy Number validation
```

```
function validateVacancy() {  
    const vacancy = document.getElementById('vacancy').value.trim();  
    const error = document.getElementById('vacancyerror');
```

```
    if (!vacancy) {  
        error.textContent = "Number of vacancies is required.";  
        return false;  
    }
```

```
    if (!/^\d+$/i.test(vacancy)) {  
        error.textContent = "Only numbers are allowed";  
        return false;  
    }
```

```
    if (parseInt(vacancy) <= 0) {
```

```
    error.textContent = "Number of vacancies must be greater than 0.";
    return false;
}
```

```
error.textContent = "";
return true;
}
```

// Salary validation

```
function validateSalary() {
    const salary = document.getElementById('salary').value.trim();
    const error = document.getElementById('salaryerror');

    if (!salary) {
        error.textContent = "Salary is required.";
        return false;
    }
    if (!/^\d+$/.test(salary)) {
        error.textContent = "Only numbers are allowed";
        return false;
    }
    if (parseInt(salary) <= 0) {
        error.textContent = "Salary must be greater than 0.";
        return false;
    }

    error.textContent = "";
    return true;
}
```

```
// Last Date validation

function validateLastDate() {

    const lastDate = document.getElementById('last_date').value;
    const vacancyDate = document.getElementById('date').value;
    const error = document.getElementById('lastdateerror');

    if (!lastDate) {
        error.textContent = "Application deadline is required.";
        return false;
    }

    const deadline = new Date(lastDate);
    const today = new Date();
    const vacancy = new Date(vacancyDate);

    if (deadline < today)
    {
        error.textContent = "Application deadline cannot be in the past.";
        return false;
    }
    if (deadline > vacancy)
    {
        error.textContent = "Application deadline must be before vacancy date.";
        return false;
    }
    error.textContent = "";
    return true;
}

document.addEventListener('DOMContentLoaded', function() {
```

```
// Initialize time selectors

let startTimeSelect = document.getElementById("start-time");
let endTimeSelect = document.getElementById("end-time");


// Generate time options
function generateTimeOptions(selectElement) {
  selectElement.innerHTML = '<option value="">Select Time</option>';
  for (let hour = 0; hour < 24; hour++) {
    for (let min = 0; min < 60; min += 30) { // 30-minute intervals
      let formattedHour = hour % 12 || 12; // Convert to 12-hour format
      let amPm = hour < 12 ? "AM" : "PM";
      let formattedMinute = min.toString().padStart(2, "0");
      let timeValue = `${hour.toString().padStart(2, "0")}:${formattedMinute}`;
      let timeText = `${formattedHour}:${formattedMinute} ${amPm}`;

      let option = document.createElement('option');
      option.value = timeValue;
      option.textContent = timeText;
      selectElement.appendChild(option);
    }
  }
}


// Generate options for both selects
generateTimeOptions(startTimeSelect);
generateTimeOptions(endTimeSelect);


// Add change event listeners
startTimeSelect.addEventListener('change', validateTimeSelection);
endTimeSelect.addEventListener('change', validateTimeSelection);
```



```
});
```

```
function validateDeliveryDocs()
```

```
{
```

```
    const category = document.getElementById('category').value;
```

```
    if (category === "Delivery and logistics") {
```

```
        const license = document.getElementById('license_required').value;
```

```
        const badge = document.getElementById('badge_required').value;
```

```
        const licenseError = document.getElementById('licenseerror');
```

```
        const badgeError = document.getElementById('badgeerror');
```

```
        let isValid = true;
```

```
        if (!license) {
```

```
            licenseError.textContent = "Please select license requirement";
```

```
            isValid = false;
```

```
        } else {
```

```
            licenseError.textContent = "";
```

```
        }
```

```
        if (!badge) {
```

```
            badgeError.textContent = "Please select badge requirement";
```

```
            isValid = false;
```

```
        } else {
```

```
            badgeError.textContent = "";
```

```
        }
```

```
        return isValid;
```

```
    }
```

```
    return true;
```

```
}
```

```
// Working Hour validation
```

```
function validateWorkingHour() {  
    return validateTimeSelection(); // Since we already have time validation  
}
```

```
// Phone validation
```

```
function validatePhone() {  
    const phone = document.getElementById('phone').value.trim();  
    const error = document.getElementById('phoneerror');  
  
    if (!phone)  
    {  
        error.textContent = "Phone number is required.";   
        return false;  
    }  
    else if (! /^[6-9]/.test(phone))  
    {  
        error.textContent = "Phone number must start with 6, 7, 8, or 9.";   
        return false;  
    }  
    else if (!/^ \d{10}$/.test(phone))  
    {  
        error.textContent = "Phone number must be 10 digits.";   
        return false;  
    }  
    error.textContent = "";  
    return true;  
}
```

```
// Working Days validation
```

```
function validateWorkingDays() {  
    const workingDays = document.getElementById('working_days').value;  
    const error = document.getElementById('workingdayerror');  
  
    if (!workingDays) {  
        error.textContent = "Please select working days";  
        return false;  
    }  
  
    error.textContent = "";  
    return true;  
}
```

```
// Add event listeners for real-time validation
```

```
document.addEventListener('DOMContentLoaded', function()  
{  
    // Real-time validation for text inputs  
    document.getElementById('job_title').addEventListener('input', validateJobTitle);  
    document.getElementById('job_description').addEventListener('input',  
validateJobDescription);  
    document.getElementById('working_hour').addEventListener('input',  
validateWorkingHour);  
    document.getElementById('vacancy').addEventListener('input', validateVacancy);  
    document.getElementById('salary').addEventListener('input', validateSalary);  
  
    // Validation for date inputs  
    document.getElementById('date').addEventListener('change', function()  
    {  
        validateDate();  
        validateLastDate(); // Revalidate last date when vacancy date changes
```

```
});  
  
document.getElementById('last_date').addEventListener('change', validateLastDate);  
  
document.getElementById('category').addEventListener('change', validatecategory);  
  
document.getElementById('license_required').addEventListener('change',  
validateDeliveryDocs);  
  
document.getElementById('badge_required').addEventListener('change',  
validateDeliveryDocs);  
  
});  
  
// Main form validation  
function validateForm() {  
    let isValid = true;  
  
    // Run all validations  
    const validations = [  
        validatecategory(),  
        validateJobTitle(),  
        validateLocation(),  
        validateJobDescription(),  
        validateWorkingHour(),  
        validateDate(),  
        validateVacancy(),  
        validateSalary(),  
        validateLastDate(),  
        validateTimeSelection(),  
        validateDeliveryDocs(),  
        validatePhone(),  
        validateWorkingDays()  
    ];
```

```

// If any validation fails, form is invalid
if (validations.includes(false)) {
    isValid = false;
}

return isValid;
}

// Form submission handler
document.getElementById('add_job').addEventListener('submit', function(e) {
    if (!validateForm()) {
        e.preventDefault(); // Prevent form submission if validation fails
    }
});

```

Accept.php

```

<?php
    session_start();
    include '../database/connectdatabase.php';
    require '../vendor/autoload.php'; // For PHPMailer
    $dbname = "project";
    mysqli_select_db($conn, $dbname);

    if (!isset($_SESSION['employer_id'])) {
        header("Location: ../login/loginvalidation.php");
        exit();
    }

    if (isset($_GET['application_id']) && isset($_GET['status'])) {
        $application_id = $_GET['application_id'];
    }

```

```

$status = $_GET['status'];

// Validate status
if (!in_array($status, ['accepted', 'rejected'])) {
    header("Location: myjoblist.php");
    exit();
}

// Get job_id and check vacancy status before updating
$sql = "SELECT a.job_id, a.user_id, j.job_title, e.company_name, e.phone_number,
u.first_name, u.last_name, l.email, l2.email as employer_email
FROM tbl_applications a
JOIN tbl_jobs j ON a.job_id = j.job_id
JOIN tbl_employer e ON j.employer_id = e.employer_id
JOIN tbl_user u ON a.user_id = u.user_id
JOIN tbl_login l ON u.user_id = l.user_id
JOIN tbl_login l2 ON e.employer_id = l2.employer_id
WHERE a.id = ?";

$stmt = mysqli_prepare($conn, $sql);
mysqli_stmt_bind_param($stmt, "i", $application_id);
mysqli_stmt_execute($stmt);
$result = mysqli_stmt_get_result($stmt);

if ($row = mysqli_fetch_assoc($result)) {
    $job_id = $row['job_id'];
    $user_email = $row['email'];
    $user_name = $row['first_name'] . ' ' . $row['last_name'];
    $job_title = $row['job_title'];
    $company_name = $row['company_name'];
    $employer_phone = $row['phone_number'];
    $employer_email = $row['employer_email'];
}

```

```

// Check vacancy limit

$vacancy_sql = "SELECT
    (SELECT vacancy FROM tbl_jobs WHERE job_id = ?) as total_vacancy,
    (SELECT COUNT(*) FROM tbl_applications WHERE job_id = ? AND status =
'accepted') as filled_vacancy";

$stmt = mysqli_prepare($conn, $vacancy_sql);
mysqli_stmt_bind_param($stmt, "ii", $job_id, $job_id);
mysqli_stmt_execute($stmt);
$vacancy_result = mysqli_stmt_get_result($stmt);
$vacancy_data = mysqli_fetch_assoc($vacancy_result);

if ($status === 'accepted' &&
    $vacancy_data['filled_vacancy'] >= $vacancy_data['total_vacancy']) {
    header("Location: applicants.php?job_id=" . $job_id . "&error=vacancy_full");
    exit();
}

// Update the application status
$update_sql = "UPDATE tbl_applications SET status = ? WHERE id = ?";
$stmt = mysqli_prepare($conn, $update_sql);
mysqli_stmt_bind_param($stmt, "si", $status, $application_id);

if (mysqli_stmt_execute($stmt)) {
    // Send email for both accepted and rejected status
    $mail = new PHPMailer\PHPMailer\PHPMailer(true);

    try {
        // Email server settings
        $mail->isSMTP();
        $mail->Host = 'smtp.gmail.com';
    }
}

```

```

$mail->SMTPAuth = true;

$mail->Username = 'flexihire369@gmail.com';

$mail->Password = 'xtik ztxp zaji bszk';

$mail->SMTPSecure =
PHPMailer\PHPMailer\PHPMailer::ENCRYPTION_STARTTLS;

$mail->Port = 587;

// Recipients
$mail->setFrom($employer_email, $company_name);
$mail->addAddress($user_email);

// Content
$mail->isHTML(true);

if ($status === 'accepted') {
    // Insert into tbl_selected
    $insert_sql = "INSERT INTO tbl_selected (application_id, job_id, user_id)
        VALUES (?, ?, ?)";

    $stmt = mysqli_prepare($conn, $insert_sql);
    mysqli_stmt_bind_param($stmt, "iii", $application_id, $job_id,
$row['user_id']);
    mysqli_stmt_execute($stmt);

    $mail->Subject = "Congratulations! You've Been Selected for $job_title";
    $mail->Body = "
        <div style='font-family: Arial, sans-serif; max-width: 600px; margin: 0
auto;'>
            <div style='background-color: #f8f9fa; padding: 20px; text-align: center;
border-radius: 10px 0 0;'>
                <h2 style='color: #4a90e2; margin: 0;'>Congratulations!</h2>
            </div>

```



```
<div style='background-color: white; padding: 20px; border-radius: 0 0 10px; box-shadow: 0 2px 4px rgba(0,0,0,0.1);'>
```

```
<p>Dear <strong>$user_name</strong>,</p>
```

```
<p>We are pleased to inform you that you have been selected for the position of <strong>$job_title</strong> at <strong>$company_name</strong>.</p>
```

```
<div style='margin: 20px 0; padding: 15px; background-color: #f8f9fa; border-radius: 5px;'>
```

```
<p style='margin: 0; color: #666;'>
```

```
<strong>Next Steps:</strong><br>
```

```
Please contact us at: <strong>$employer_phone</strong> for further details.
```

```
</p>
```

```
</div>
```

```
<p style='margin-top: 20px;'>Best regards,<br>$company_name</p>
```

```
</div>
```

```
</div>";
```

```
} else {
```

```
// Rejection email
```

```
$mail->Subject = "Update Regarding Your Application for $job_title";
```

```
$mail->Body = "
```

```
<div style='font-family: Arial, sans-serif; max-width: 600px; margin: 0 auto;'>
```

```
<div style='background-color: #f8f9fa; padding: 20px; text-align: center; border-radius: 10px 0 0;'>
```

```
<h2 style='color: #4a90e2; margin: 0;'>Application Update</h2>
```

```
</div>
```

```
<div style='background-color: white; padding: 20px; border-radius: 0 0 10px; box-shadow: 0 2px 4px rgba(0,0,0,0.1);'>
```

```
<p>Dear <strong>$user_name</strong>,</p>
```

<p>Thank you for your interest in the \$job_title
position at \$company_name.</p>

<p>After careful consideration of your application, we regret to inform
you that we have decided to move forward with other candidates whose qualifications more
closely match our current needs.</p>

<p>We encourage you to apply for future positions that match your
skills and experience.</p>

<div style='margin: 20px 0; padding: 15px; background-color: #f8f9fa;
border-radius: 5px;'>

<p style='margin: 0; color: #666;'>

We wish you the best in your job search and future professional
endeavors.

</p>

</div>

<p style='margin-top: 20px;'>Best regards,
\$company_name</p>

</div>

</div>";

}

\$mail->send();

} catch (Exception \$e) {

error_log("Email sending failed: " . \$mail->ErrorInfo);

}

}

header("Location: applicants.php?job_id=" . \$job_id . "&success=status_updated");

} else {

```

        header("Location: myjoblist.php");
    }
    exit();
} else {
    header("Location: myjoblist.php");
    exit();
}
?>

```

Admindashboard.php

```

<?php
    session_start();
    include '../database/connectdatabase.php';
    $dbname = "project";
    mysqli_select_db($conn, $dbname);

    // Add the time_elapsed_string function here
    function time_elapsed_string($datetime) {
        $now = new DateTime;
        $ago = new DateTime($datetime);
        $diff = $now->diff($ago);

        if ($diff->y > 0) {
            return $diff->y . ' year' . ($diff->y > 1 ? 's' : '') . ' ago';
        }
        if ($diff->m > 0) {
            return $diff->m . ' month' . ($diff->m > 1 ? 's' : '') . ' ago';
        }
        if ($diff->d > 0) {
            if ($diff->d == 1) {
                return 'Yesterday';
            }
        }
    }

```

```

    }

    return $diff->d . ' days ago';
}

if ($diff->h > 0) {
    return $diff->h . ' hour' . ($diff->h > 1 ? 's' : '') . ' ago';
}

if ($diff->i > 0) {
    return $diff->i . ' minute' . ($diff->i > 1 ? 's' : '') . ' ago';
}

return 'Just now';
}

```

```

// Check if admin is logged in
if (!isset($_SESSION['admin_id'])) {
    header("Location: ../login/loginvalidation.php");
    exit();
}

```

```

//fetch stats div values

```

```

//fetching no of users are registered

```

```

$sql_total_users="SELECT COUNT(*) AS total_users FROM tbl_user";
$result_total_users=mysqli_query($conn,$sql_total_users);
$row_total_users=mysqli_fetch_assoc($result_total_users);
$total_users=$row_total_users['total_users'];

```

```

//fetching no of employers are registered

```

```

$sql_total_employers="SELECT COUNT(*) AS total_employers FROM tbl_employer";
$result_total_employers=mysqli_query($conn,$sql_total_employers);
$row_total_employers=mysqli_fetch_assoc($result_total_employers);

```

```
$total_employers=$row_total_employers['total_employers'];
```

```
//fetching no of jobs are posted
```

```
$sql_total_jobs="SELECT COUNT(*) AS total_jobs FROM tbl_jobs";
```

```
$result_total_jobs=mysqli_query($conn,$sql_total_jobs);
```

```
$row_total_jobs=mysqli_fetch_assoc($result_total_jobs);
```

```
$total_jobs=$row_total_jobs['total_jobs'];
```

```
//fetching no of reports
```

```
$sql_total_reports="SELECT COUNT(*) AS total_reports FROM tbl_reports";
```

```
$result_total_reports=mysqli_query($conn,$sql_total_reports);
```

```
$row_total_reports=mysqli_fetch_assoc($result_total_reports);
```

```
$total_reports=$row_total_reports['total_reports'];
```

```
// // Fetch dashboard statistics
```

```
// $stats = array();
```

```
// // Total Users
```

```
// $sql_users = "SELECT COUNT(*) as count FROM tbl_user";
```

```
// $result_users = mysqli_query($conn, $sql_users);
```

```
// $stats['users'] = mysqli_fetch_assoc($result_users)['count'];
```

```
// // Total Employers
```

```
// $sql_employers = "SELECT COUNT(*) as count FROM tbl_employer";
```

```
// $result_employers = mysqli_query($conn, $sql_employers);
```

```
// $stats['employers'] = mysqli_fetch_assoc($result_employers)['count'];
```

```
// // Total Jobs
```

```
// $sql_jobs = "SELECT COUNT(*) as count FROM tbl_jobs";
```

```
// $result_jobs = mysqli_query($conn, $sql_jobs);
```

```

// $stats['jobs'] = mysqli_fetch_assoc($result_jobs)['count'];

// // Total Applications
// $sql_applications = "SELECT COUNT(*) as count FROM tbl_applications";
// $result_applications = mysqli_query($conn, $sql_applications);
// $stats['applications'] = mysqli_fetch_assoc($result_applications)['count'];

// // Recent Activities (last 10)
// $sql_activities = "SELECT * FROM (
//     SELECT 'New User' as type, first_name, last_name, created_at
//     FROM tbl_user
//     UNION ALL
//     SELECT 'New Employer' as type, company_name as first_name, " as last_name,
//     created_at
//     FROM tbl_employer
//     UNION ALL
//     SELECT 'New Job' as type, job_title as first_name, " as last_name, posted_date as
//     created_at
//     FROM tbl_jobs
// ) as activities
// ORDER BY created_at DESC
// LIMIT 10";

// Add this before the HTML section, with your other queries
// Recent Activities Query
$sql_activities = "SELECT * FROM (
    SELECT
        'New User' as type,
        u.first_name,
        u.last_name,
        l.email,

```

```
    u.created_at,  
    l.status,  
    'user' as category  
FROM tbl_user u  
LEFT JOIN tbl_login l ON u.user_id = l.user_id
```

UNION ALL

```
SELECT  
    'New Employer' as type,  
    e.company_name as first_name,  
    " as last_name,  
    l.email,  
    e.created_at,  
    l.status,  
    'employer' as category  
FROM tbl_employer e  
LEFT JOIN tbl_login l ON e.employer_id = l.employer_id
```

UNION ALL

```
SELECT  
    'New Job' as type,  
    j.job_title as first_name,  
    e.company_name as last_name,  
    l.email,  
    j.created_at,  
    j.status,  
    'job' as category  
FROM tbl_jobs j
```

```
JOIN tbl_employer e ON j.employer_id = e.employer_id
LEFT JOIN tbl_login l ON e.employer_id = l.employer_id
) as activities
ORDER BY created_at DESC
LIMIT 10";
```

```
$result_activities = mysqli_query($conn, $sql_activities);
```

```
if (!$result_activities) {
    die("Query failed: " . mysqli_error($conn));
}
?>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Admin Dashboard</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.4.0/css/all.min.css">
    <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&displ
ay=swap" rel="stylesheet">
    <link rel="stylesheet" href="admindashboard.css">
<style>
    /* Activity list container */
    .activity-list {
        display: grid;
        grid-template-columns: repeat(auto-fill, minmax(350px, 1fr));
        gap: 20px;
```



```

        padding: 15px 0;
    }

    /* Responsive adjustments */
    @media (max-width: 1400px) {
        .activity-list {
            grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));
        }
    }

    @media (max-width: 768px) {
        .activity-list {
            grid-template-columns: 1fr;
        }
    }
</style>
</head>
<body>
    <div class="dashboard-container">
        <!-- Sidebar -->
        <div class="sidebar">
            <div class="logo-section">
                <h1>FlexiHire</h1>
                <div class="admin-badge">
                    <i class="fas fa-user-shield"></i>
                    <span>Admin Dashboard</span>
                </div>
            </div>
            <div>
                <nav class="nav-menu">
                    <a href="admindashboard.php" class="nav-item active">

```

```
        <i class="fas fa-home"></i>
        <span>Dashboard</span>
    </a>

    <a href="manage_users.php" class="nav-item">
        <i class="fas fa-users"></i>
        <span>Manage Users</span>
    </a>

    <a href="manage_employers.php" class="nav-item">
        <i class="fas fa-building"></i>
        <span>Manage Employers</span>
    </a>

    <a href="manage_jobs.php" class="nav-item">
        <i class="fas fa-briefcase"></i>
        <span>Manage Jobs</span>
    </a>

    <a href="reports.php" class="nav-item">
        <i class="fas fa-chart-bar"></i>
        <span>Reports</span>
    </a>

    <a href="../login/logout.php" class="nav-item">
        <i class="fas fa-sign-out-alt"></i>
        <span>Logout</span>
    </a>
</nav>
</div>
```

```
<!-- Main Content -->
<div class="main-content">
    <div class="stats-grid">
        <!-- Users Stats -->
```

```
<div class="stat-card">

  <div class="stat-header">

    <span class="stat-title">Total Users</span>

    <div class="stat-icon bg-blue">

      <i class="fas fa-users"></i>

    </div>

  </div>

  <div class="stat-value" data-target="<?php echo $total_users; ?>">0</div>

  <div class="stat-change">

    <i class="fas fa-arrow-up"></i>

    <a href="manage_users.php">Active Job Seekers</a>

  </div>

</div>
```

```
<!-- Employers Stats -->
```

```
<div class="stat-card">

  <div class="stat-header">

    <span class="stat-title">Total Employers</span>

    <div class="stat-icon bg-green">

      <i class="fas fa-building"></i>

    </div>

  </div>

  <div class="stat-value" data-target="<?php echo $total_employers; ?>">0</div>

  <div class="stat-change">

    <i class="fas fa-arrow-up"></i>

    <a href="manage_employers.php">Registered Companies</a>

  </div>

</div>
```

```
<!-- Jobs Stats -->
```

```
<div class="stat-card">

  <div class="stat-header">

    <span class="stat-title">Total Jobs</span>

    <div class="stat-icon bg-yellow">

      <i class="fas fa-briefcase"></i>

    </div>

  </div>

  <div class="stat-value" data-target="<?php echo $total_jobs; ?>">0</div>

  <div class="stat-change">

    <i class="fas fa-arrow-up"></i>

    <a href="manage_jobs.php">Active Listings</a>

  </div>

</div>
```

```
<!-- Reports Stats -->
```

```
<div class="stat-card">

  <div class="stat-header">

    <span class="stat-title">Total Reports</span>

    <div class="stat-icon bg-red">

      <i class="fas fa-file-alt"></i>

    </div>

  </div>

  <div class="stat-value" data-target="<?php echo $total_reports; ?>">0</div>

  <div class="stat-change">

    <i class="fas fa-arrow-up"></i>

    <a href="reports.php">View Reports</a>

  </div>

</div>

</div>
```

```

<!-- Recent Activity Section with essential information -->
<div class="recent-activity">
    <div class="section-header">
        <h2><i class="fas fa-history"></i> Recent Activities</h2>
    </div>

    <div class="activity-list">
        <?php if(mysqli_num_rows($result_activities) > 0): ?>
            <?php while($activity = mysqli_fetch_assoc($result_activities)): ?>
                <div class="activity-item">
                    <div class="activity-icon bg-<?php echo $activity['category']; ?>">
                        <i class="fas fa-<?php
                            echo $activity['category'] === 'user' ? 'user' :
                                ($activity['category'] === 'employer' ? 'building' :
                                    ($activity['category'] === 'job' ? 'briefcase' : 'file-alt'));
                            ?>"></i>
                    </div>

                    <div class="activity-content">
                        <div class="activity-header">
                            <div class="activity-info">
                                <span class="activity-name">
                                    <?php
                                        switch($activity['category']) {
                                            case 'user':
                                                echo htmlspecialchars($activity['first_name']) . ' ' .
$activity['last_name']);
                                            break;
                                            case 'employer':
                                                echo htmlspecialchars($activity['first_name']); // company
name

```

```

        break;
    case 'job':
        echo htmlspecialchars($activity['first_name']); // job title
        echo ' at ' . htmlspecialchars($activity['last_name']); //
company name

        break;
    default:
        echo htmlspecialchars($activity['first_name']);
    }
?>
</span>
<span class="activity-type"><?php echo $activity['type'];
?></span>

</div>
<span class="activity-time">
    <?php echo time_elapsed_string($activity['created_at']); ?>
</span>
</div>

<?php if($activity['email']): ?>
    <div class="activity-subtitle">
        <i class="fas fa-envelope"></i>
        <?php echo htmlspecialchars($activity['email']); ?>
    </div>
<?php endif; ?>

<div class="activity-status">
    <span class="status-badge <?php echo strtolower($activity['status']);
?>">

        <?php echo $activity['status']; ?>
    </span>

```

```

        </div>
    </div>
</div>
<?php endwhile; ?>
<?php else: ?>
    <div class="no-activity">
        <i class="fas fa-info-circle"></i>
        <p>No recent activities found</p>
    </div>
<?php endif; ?>
</div>
</div>
</div>
</div>
</div>

<script src="admindashboard.js"></script>
</body>
</html>

```

Activate user.php

```

<?php
    session_start();
    include '../database/connectdatabase.php';
    $dbname="project";
    mysqli_select_db($conn,$dbname);

    // Check if admin is logged in
    if (!isset($_SESSION['admin_id'])) {
        header('Location: ../login/loginvalidation.php');
        exit();
    }

```

```

$user_id = filter_input(INPUT_GET, 'user_id', FILTER_VALIDATE_INT);

if ($user_id) {
    $sql = "UPDATE tbl_login SET status = 'active' WHERE user_id = ?";
    $stmt = mysqli_prepare($conn, $sql);
    mysqli_stmt_bind_param($stmt, "i", $user_id);
    $result = mysqli_stmt_execute($stmt);

    if ($result) {
        $_SESSION['message'] = "User account has been activated successfully";
        $_SESSION['message_type'] = "success";
    } else {
        $_SESSION['message'] = "Error activating user account";
        $_SESSION['message_type'] = "error";
    }
}

header('Location: manage_users.php');
exit();
?>

```

Deactivate user.php

```

<?php
session_start();
require_once '../database/connectdatabase.php';
require '../vendor/autoload.php';

// Check if admin is logged in

```



```

if (!isset($_SESSION['admin_id'])) {
    header('Location: ../login/loginvalidation.php');
    exit();
}

// Get user_id from URL
$user_id = filter_input(INPUT_GET, 'user_id', FILTER_VALIDATE_INT);

try {
    // Update login status to inactive
    $login_sql = "UPDATE tbl_login SET status = 'inactive' WHERE user_id = ?";
    $login_stmt = mysqli_prepare($conn, $login_sql);
    mysqli_stmt_bind_param($login_stmt, "i", $user_id);
    mysqli_stmt_execute($login_stmt);

    // Get user email for notification
    $email_sql = "SELECT l.email, u.first_name, u.last_name
        FROM tbl_user u
        JOIN tbl_login l ON u.user_id = l.user_id
        WHERE u.user_id = ?";
    $email_stmt = mysqli_prepare($conn, $email_sql);
    mysqli_stmt_bind_param($email_stmt, "i", $user_id);
    mysqli_stmt_execute($email_stmt);
    $result = mysqli_stmt_get_result($email_stmt);
    $user = mysqli_fetch_assoc($result);

    if ($user) {
        $mail = new PHPMailer\PHPMailer\PHPMailer(true);

        // Email server settings

```

```
$mail->isSMTP();  
$mail->Host = 'smtp.gmail.com';  
$mail->SMTPAuth = true;  
$mail->Username = 'flexihire369@gmail.com';  
$mail->Password = 'xtik ztxp zaji bszk';  
$mail->SMTPSecure =  
PHPMailer\PHPMailer\PHPMailer::ENCRYPTION_STARTTLS;  
$mail->Port = 587;
```

```
// Recipients
```

```
$mail->setFrom('flexihire369@gmail.com', 'FlexiHire');  
$mail->addAddress($user['email']);
```

```
// Content
```

```
$mail->isHTML(true);  
$mail->Subject = 'Account Deactivation Notice - FlexiHire';  
$mail->Body = "  
    <div style='font-family: Arial, sans-serif; max-width: 600px; margin: 0 auto;*>  
        <div style='background-color: #f8f9fa; padding: 20px; text-align: center; border-  
radius: 10px 0 0;*>  
            <h2 style='color: #dc3545; margin: 0;*>Account Deactivation Notice</h2>  
        </div>  
  
        <div style='background-color: white; padding: 20px; border-radius: 0 0 10px; box-  
shadow: 0 2px 4px rgba(0,0,0,0.1);*>  
            <p>Dear <strong>{$user['first_name']} {$user['last_name']}</strong>,</p>  
  
            <p>We regret to inform you that your FlexiHire account has been deactivated  
due to violation of our terms of service.</p>  
  
            <p>If you believe this was done in error or would like to appeal this decision,  
please contact our support team at flexihire369@gmail.com.</p>
```

```
<p style='margin-top: 20px;'>Best regards,<br>FlexiHire Admin Team</p>
</div>
```

```
<div style='text-align: center; margin-top: 20px; color: #666; font-size: 12px;'>
    <p>This is an automated message from FlexiHire. Please do not reply to this
email.</p>
</div>
</div>";
```

```
$mail->send();
}
```

```
$_SESSION['message'] = "User account has been deactivated successfully";
$_SESSION['message_type'] = "success";
} catch (Exception $e) {
    $_SESSION['message'] = "Error deactivating user account";
    $_SESSION['message_type'] = "error";
    error_log($e->getMessage());
}
```

```
header('Location: manage_users.php');
exit();
?>
```