<?php

session\_start();

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

    header("Location: ../employee/employeelogin.php");

    exit();

}

// Include the database connection

include '../../db/db\_conn.php';

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

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

// Fetch user info from the employee\_register table

$employeeId = $\_SESSION['employee\_id'];

$sql = "SELECT employee\_id, first\_name, middle\_name, last\_name, birthdate, gender, email, role, position, department, phone\_number, address, pfp

        FROM employee\_register

        WHERE employee\_id = ?";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("s", $employeeId);

$stmt->execute();

$result = $stmt->get\_result();

$employeeInfo = $result->fetch\_assoc();

if (!$employeeInfo) {

    die("Error: Employee information not found.");

}

// Define the position

$position = 'employee';

// Fetch employee records where position is 'employee' and department matches the logged-in employee's department

$sql = "SELECT employee\_id, first\_name, last\_name, role, position

        FROM employee\_register

        WHERE position = ? AND department = ? AND role IN ('supervisor', 'staff', 'admin', 'fieldworker', 'contractual')";

$stmt = $conn->prepare($sql);

$stmt->bind\_param('ss', $position, $department);

$stmt->execute();

$result = $stmt->get\_result();

// Fetch evaluations for this employee from the evaluations table

$evaluatedEmployees = [];

$evalSql = "SELECT employee\_id FROM ptp\_evaluations WHERE evaluator\_id = ?";

$evalStmt = $conn->prepare($evalSql);

$evalStmt->bind\_param('s', $employeeId);

$evalStmt->execute();

$evalResult = $evalStmt->get\_result();

if ($evalResult->num\_rows > 0) {

    while ($row = $evalResult->fetch\_assoc()) {

        $evaluatedEmployees[] = $row['employee\_id'];

    }

}

// Fetch evaluation questions from the database for each category and role

$categories = ['Quality of Work', 'Communication Skills', 'Teamwork', 'Punctuality', 'Initiative'];

$questions = [];

foreach ($categories as $category) {

    $categorySql = "SELECT question FROM evaluation\_questions WHERE category = ? AND role = ?";

    $categoryStmt = $conn->prepare($categorySql);

    $categoryStmt->bind\_param('ss', $category, $role);

    $categoryStmt->execute();

    $categoryResult = $categoryStmt->get\_result();

    $questions[$category] = [];

    if ($categoryResult->num\_rows > 0) {

        while ($row = $categoryResult->fetch\_assoc()) {

            $questions[$category][] = $row['question'];

        }

    }

}

// Check if any records are found

$employees = [];

if ($result->num\_rows > 0) {

    while ($row = $result->fetch\_assoc()) {

        if ($row['employee\_id'] != $employeeId) {

            $employees[] = $row;

        }

    }

}

// Calculate statistics

$totalEmployees = count($employees);

$evaluatedCount = count($evaluatedEmployees);

$pendingCount = $totalEmployees - $evaluatedCount;

$completionPercentage = $totalEmployees > 0 ? round(($evaluatedCount / $totalEmployees) \* 100) : 0;

$pendingPercentage = 100 - $completionPercentage;

// Close the database connection

$conn->close();

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Employee Evaluation Dashboard</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">

    <link href="../../css/styles.css" rel="stylesheet" />

    <link href="../../css/star.css" rel="stylesheet" />

    <link href='https://cdn.jsdelivr.net/npm/fullcalendar@5.11.3/main.min.css' rel='stylesheet' />

    <link href="../../css/calendar.css" rel="stylesheet"/>

    <script src="https://use.fontawesome.com/releases/v6.3.0/js/all.js" crossorigin="anonymous"></script>

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css">

    <style>

        :root {

            --dark-bg: #121212;

            --card-bg: #1e1e1e;

            --darker-card-bg: #1a1a1a;

            --text-primary: #ffffff;

            --text-secondary: #b0b0b0;

            --border-color: #2d2d2d;

            --blue: #3a86ff;

            --green: #06d6a0;

            --red: #ef476f;

        }

        body {

            background-color: var(--dark-bg);

            color: var(--text-primary);

            font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

        }

        .navbar {

            background-color: var(--card-bg) !important;

            box-shadow: 0 2px 10px rgba(0, 0, 0, 0.3);

        }

        #layoutSidenav\_nav {

            background-color: var(--card-bg);

            border-right: 1px solid var(--border-color);

        }

        .sb-sidenav {

            background-color: var(--card-bg);

        }

        .stats-container {

            display: grid;

            grid-template-columns: repeat(3, 1fr);

            gap: 15px;

            margin-bottom: 20px;

        }

        .stat-card {

            background-color: var(--darker-card-bg);

            border-radius: 8px;

            padding: 20px;

            text-align: center;

            display: flex;

            flex-direction: column;

            align-items: center;

            justify-content: center;

        }

        .stat-icon {

            width: 40px;

            height: 40px;

            border-radius: 50%;

            display: flex;

            align-items: center;

            justify-content: center;

            margin-bottom: 10px;

            font-size: 1.2rem;

        }

        .stat-icon.blue {

            background-color: var(--blue);

            color: white;

        }

        .stat-icon.green {

            background-color: var(--green);

            color: white;

        }

        .stat-icon.red {

            background-color: var(--red);

            color: white;

        }

        .stat-value {

            font-size: 2.5rem;

            font-weight: 700;

            margin-bottom: 5px;

            line-height: 1;

        }

        .stat-label {

            color: var(--text-secondary);

            font-size: 0.9rem;

        }

        /\* Add this to your CSS styles section \*/

        .progress-container {

            position: relative;

            height: 30px;

            background-color: #2a2a2a;

            border-radius: 15px;

            overflow: hidden;

            margin-bottom: 10px;

        }

        .progress-bar {

            height: 100%;

            border-radius: 15px;

            position: relative;

            transition: width 0.5s ease, background-color 0.5s ease;

        }

        .progress-bar.complete {

            background-color: var(--green);

            z-index: 2; /\* Ensure the green bar is on top \*/

        }

        .progress-bar.pending {

            background-color: var(--red);

            z-index: 1; /\* Red bar stays behind \*/

        }

        .progress-label {

            position: absolute;

            top: 50%;

            left: 50%;

            transform: translate(-50%, -50%);

            color: white;

            font-weight: 600;

            font-size: 0.9rem;

            text-shadow: 0 0 3px rgba(0, 0, 0, 0.5);

            z-index: 3; /\* Ensure the label is always on top \*/

            width: 100%;

            text-align: center;

        }

        .progress-section {

            background-color: var(--darker-card-bg);

            border-radius: 8px;

            padding: 20px;

            margin-bottom: 20px;

        }

        .progress-section h2 {

            font-size: 1.2rem;

            margin-bottom: 15px;

            font-weight: 500;

        }

        .progress-stats {

            display: flex;

            justify-content: space-between;

            font-size: 0.9rem;

        }

        .progress-stats .complete {

            color: var(--green); /\* Green color for complete \*/

        }

        .progress-stats .pending {

            color: var(--red); /\* Red color for pending \*/

        }

        /\* Rest of your CSS styles remain the same \*/

        .employees-section {

            background-color: var(--darker-card-bg);

            border-radius: 8px;

            padding: 20px;

        }

        .section-header {

            display: flex;

            justify-content: space-between;

            align-items: center;

            margin-bottom: 20px;

        }

        .section-header h2 {

            font-size: 1.2rem;

            margin: 0;

            display: flex;

            align-items: center;

            gap: 10px;

            font-weight: 500;

        }

        .section-header i {

            color: var(--text-primary);

        }

        .search-container {

            position: relative;

        }

        .search-input {

            background-color: #2a2a2a;

            border: 1px solid var(--border-color);

            color: var(--text-primary);

            padding: 8px 15px 8px 35px;

            border-radius: 50px;

            width: 250px;

        }

        .search-input:focus {

            outline: none;

            border-color: var(--blue);

            box-shadow: 0 0 0 2px rgba(58, 134, 255, 0.2);

        }

        .search-icon {

            position: absolute;

            left: 12px;

            top: 50%;

            transform: translateY(-50%);

            color: var(--text-secondary);

        }

        .employees-table {

            width: 100%;

            border-collapse: separate;

            border-spacing: 0;

        }

        .employees-table th {

            text-align: left;

            padding: 12px 15px;

            color: var(--text-secondary);

            font-weight: 600;

            font-size: 0.9rem;

            border-bottom: 1px solid var(--border-color);

        }

        .employees-table td {

            padding: 12px 15px;

            border-bottom: 1px solid var(--border-color);

        }

        .employees-table tr:last-child td {

            border-bottom: none;

        }

        .employee-avatar {

            width: 40px;

            height: 40px;

            border-radius: 50%;

            display: flex;

            align-items: center;

            justify-content: center;

            font-weight: 600;

            color: white;

            margin-right: 10px;

            background-color: #4f46e5; /\* Indigo color \*/

        }

        .employee-info {

            display: flex;

            align-items: center;

        }

        .status-badge {

            padding: 5px 10px;

            border-radius: 50px;

            font-size: 0.8rem;

            font-weight: 600;

            display: inline-flex;

            align-items: center;

            gap: 5px;

        }

        .status-badge.pending {

            background-color: var(--red);

            color: white;

        }

        .status-badge.completed {

            background-color: var(--green);

            color: white;

        }

        .btn-evaluate {

            background-color: var(--blue);

            color: white;

            border: none;

            padding: 8px 15px;

            border-radius: 6px;

            font-weight: 500;

            display: flex;

            align-items: center;

            gap: 5px;

            transition: all 0.2s ease;

        }

        .btn-evaluate:hover {

            background-color: #2a75e6;

            transform: translateY(-2px);

        }

        .btn-evaluate:disabled {

            background-color: #2a2a2a;

            color: var(--text-secondary);

            cursor: not-allowed;

            transform: none;

        }

        .modal-content {

            background-color: var(--card-bg);

            border-radius: 12px;

            border: 1px solid var(--border-color);

        }

        .modal-header {

            border-bottom: 1px solid var(--border-color);

            padding: 20px;

        }

        .modal-footer {

            border-top: 1px solid var(--border-color);

            padding: 20px;

        }

        .modal-title {

            font-weight: 600;

            color: var(--blue);

        }

        .btn-close-custom {

            background: transparent;

            border: none;

            color: var(--text-primary);

            font-size: 1.5rem;

            line-height: 1;

            padding: 0;

        }

        .evaluation-table {

            border-radius: 8px;

            overflow: hidden;

        }

        .evaluation-table th {

            background-color: rgba(58, 134, 255, 0.15);

            padding: 15px;

            font-weight: 600;

        }

        .evaluation-table td {

            padding: 15px;

            border: 1px solid var(--border-color);

        }

        .category-cell {

            background-color: rgba(131, 56, 236, 0.1);

            font-weight: 600;

            color: white;

        }

        /\* Enhanced star rating \*/

        .star-rating {

            display: flex;

            flex-direction: row-reverse;

            justify-content: flex-end;

        }

        .star-rating input {

            display: none;

        }

        .star-rating label {

            cursor: pointer;

            font-size: 1.5rem;

            color: #555;

            transition: all 0.2s ease;

            padding: 0 2px;

        }

        .star-rating label:hover,

        .star-rating label:hover ~ label,

        .star-rating input:checked ~ label {

            color: #ffbe0b; /\* Warning color for stars \*/

            transform: scale(1.1);

        }

        /\* Responsive adjustments \*/

        @media (max-width: 768px) {

            .stats-container {

                grid-template-columns: 1fr;

            }

            .search-input {

                width: 100%;

            }

            .section-header {

                flex-direction: column;

                align-items: flex-start;

                gap: 15px;

            }

            .search-container {

                width: 100%;

            }

        }

    </style>

</head>

<body class="sb-nav-fixed">

    <?php include 'navbar.php'; ?>

    <div id="layoutSidenav">

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

        <div id="layoutSidenav\_content">

            <main>

                <div class="container-fluid px-4 py-4">

                    <!-- Statistics Cards -->

                    <div class="stats-container">

                        <div class="stat-card">

                            <div class="stat-icon blue">

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

                            </div>

                            <div class="stat-value"><?php echo $totalEmployees; ?></div>

                            <div class="stat-label">Total Employees</div>

                        </div>

                        <div class="stat-card">

                            <div class="stat-icon green">

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

                            </div>

                            <div class="stat-value"><?php echo $evaluatedCount; ?></div>

                            <div class="stat-label">Evaluated</div>

                        </div>

                        <div class="stat-card">

                            <div class="stat-icon red">

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

                            </div>

                            <div class="stat-value"><?php echo $pendingCount; ?></div>

                            <div class="stat-label">Pending</div>

                        </div>

                    </div>

                    <!-- Progress Section -->

                    <div class="progress-section">

                        <h2>Evaluation Progress</h2>

                        <div class="progress-container">

                            <!-- Red background progress bar (always 100% width) -->

                            <div class="progress-bar pending" style="width: 100%;">

                                <!-- Green overlay progress bar that grows as employees are evaluated -->

                                <div class="progress-bar complete" style="width: <?php echo $completionPercentage; ?>%; position: absolute; top: 0; left: 0; height: 100%;">

                                    <!--<div class="progress-label"><?php echo $completionPercentage; ?>% Complete</div>

                                </div> -->

                                <?php if ($completionPercentage == 0): ?>

                                    <div class="progress-label">No Evaluations</div>

                                <?php endif; ?>

                            </div>

                        </div>

                        <div class="progress-stats">

                            <div class="complete"><?php echo $completionPercentage; ?>% Complete</div>

                            <div class="pending"><?php echo $pendingPercentage; ?>% Pending</div>

                        </div>

                    </div>

                    <!-- Employees Section -->

                    <div class="employees-section">

                        <div class="section-header">

                            <h2>

                                <i class="fas fa-building"></i> Administration Department Employees

                            </h2>

                            <div class="search-container">

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

                                <input type="text" class="search-input" placeholder="Search employees...">

                            </div>

                        </div>

                        <div class="table-responsive">

                            <table class="employees-table">

                                <thead>

                                    <tr>

                                        <th>Employee</th>

                                        <th>Position</th>

                                        <th>Role</th>

                                        <th>Status</th>

                                        <th>Action</th>

                                    </tr>

                                </thead>

                                <tbody>

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

                                        <?php foreach ($employees as $employee): ?>

                                            <tr>

                                                <td>

                                                    <div class="employee-info">

                                                        <div class="employee-avatar">

                                                            <?php

                                                                $initials = substr($employee['first\_name'], 0, 1) . substr($employee['last\_name'], 0, 1);

                                                                echo strtoupper($initials);

                                                            ?>

                                                        </div>

                                                        <?php echo htmlspecialchars($employee['first\_name'] . ' ' . $employee['last\_name']); ?>

                                                    </div>

                                                </td>

                                                <td><?php echo htmlspecialchars($employee['position']); ?></td>

                                                <td><?php echo htmlspecialchars($employee['role']); ?></td>

                                                <td>

                                                    <?php if (in\_array($employee['employee\_id'], $evaluatedEmployees)): ?>

                                                        <span class="status-badge completed">

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

                                                        </span>

                                                    <?php else: ?>

                                                        <span class="status-badge pending">

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

                                                        </span>

                                                    <?php endif; ?>

                                                </td>

                                                <td>

                                                    <button class="btn-evaluate"

                                                        onclick="evaluateEmployee(<?php echo $employee['employee\_id']; ?>, '<?php echo htmlspecialchars($employee['first\_name'] . ' ' . $employee['last\_name']); ?>', '<?php echo htmlspecialchars($employee['role']); ?>')"

                                                        <?php echo in\_array($employee['employee\_id'], $evaluatedEmployees) ? 'disabled' : ''; ?>>

                                                        <i class="fas fa-star"></i> Evaluate

                                                    </button>

                                                </td>

                                            </tr>

                                        <?php endforeach; ?>

                                    <?php else: ?>

                                        <tr>

                                            <td colspan="5" class="text-center py-4">

                                                <i class="fas fa-user-slash fa-3x mb-3 d-block opacity-50"></i>

                                                <p>No employees found to evaluate.</p>

                                            </td>

                                        </tr>

                                    <?php endif; ?>

                                </tbody>

                            </table>

                        </div>

                    </div>

                </div>

            </main>

            <!-- Evaluation Modal -->

            <div class="modal fade" id="evaluationModal" tabindex="-1" role="dialog" aria-labelledby="evaluationModalLabel" aria-hidden="true">

                <div class="modal-dialog modal-lg" role="document">

                    <div class="modal-content">

                        <div class="modal-header">

                            <h5 class="modal-title" id="evaluationModalLabel">Employee Evaluation</h5>

                            <button type="button" class="btn-close-custom" data-bs-dismiss="modal" aria-label="Close">

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

                            </button>

                        </div>

                        <div class="modal-body">

                            <div class="employee-info mb-4 p-3 rounded" style="background-color: rgba(58, 134, 255, 0.1);">

                                <div id="employeeDetails"></div>

                            </div>

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

                            <div id="questions"></div>

                            <div class="alert alert-info mt-3">

                                <i class="fas fa-info-circle me-2"></i>

                                Please rate all questions before submitting. 1 star is the lowest rating, and 6 stars is the highest.

                            </div>

                        </div>

                        <div class="modal-footer">

                            <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">

                                <i class="fas fa-times me-2"></i>Cancel

                            </button>

                            <button type="button" class="btn btn-primary" onclick="submitEvaluation()">

                                <i class="fas fa-paper-plane me-2"></i>Submit Evaluation

                            </button>

                        </div>

                    </div>

                </div>

            </div>

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

        </div>

    </div>

    <!-- JavaScript Code -->

    <script src='https://cdn.jsdelivr.net/npm/fullcalendar@5.11.3/main.min.js'></script>

    <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"></script>

    <script>

        // Declare evaluatedEmployees and initialize it with PHP data

        let evaluatedEmployees = <?php echo json\_encode($evaluatedEmployees); ?>;

        let currentEmployeeId, currentEmployeeName, currentEmployeeRole;

        // Function to fetch questions based on the evaluated employee's role

        async function fetchQuestions(role) {

            try {

                const response = await fetch(`../../employee\_db/supervisor/fetchQuestions.php?role=${role}`);

                if (!response.ok) {

                    throw new Error('Network response was not ok');

                }

                return await response.json();

            } catch (error) {

                console.error('Error fetching questions:', error);

                return {};

            }

        }

        async function evaluateEmployee(employee\_id, employeeName, employeeRole) {

            currentEmployeeId = employee\_id;

            currentEmployeeName = employeeName;

            currentEmployeeRole = employeeRole;

            // Show loading state

            document.getElementById('questions').innerHTML = `

                <div class="text-center p-5">

                    <div class="spinner-border text-primary" role="status">

                        <span class="visually-hidden">Loading...</span>

                    </div>

                    <p class="mt-3">Loading evaluation questions...</p>

                </div>

            `;

            // Show the modal while loading

            $('#evaluationModal').modal('show');

            try {

                // Fetch questions based on the evaluated employee's role

                const questions = await fetchQuestions(employeeRole);

                // Display employee details in the modal

                const employeeDetails = `

                    <div class="d-flex align-items-center">

                        <div class="avatar-placeholder bg-primary rounded-circle me-3 d-flex align-items-center justify-content-center" style="width: 50px; height: 50px;">

                            <span class="fs-4">${employeeName.split(' ')[0][0]}${employeeName.split(' ')[1][0]}</span>

                        </div>

                        <div>

                            <h5 class="mb-1">${employeeName}</h5>

                            <span class="badge bg-info text-dark">${employeeRole}</span>

                        </div>

                    </div>

                `;

                document.getElementById('employeeDetails').innerHTML = employeeDetails;

                // Clear previous questions

                const questionsDiv = document.getElementById('questions');

                questionsDiv.innerHTML = '';

                // Start the table structure with a more modern design

                let tableHtml = `

                <table class="table evaluation-table">

                    <thead>

                        <tr>

                            <th width="20%">Category</th>

                            <th width="50%">Question</th>

                            <th width="30%">Rating</th>

                        </tr>

                    </thead>

                    <tbody>`;

                // Loop through categories and questions to add them into the table

                for (const [category, categoryQuestions] of Object.entries(questions)) {

                    categoryQuestions.forEach((question, index) => {

                        const questionName = `${category.replace(/\s/g, '')}q${index}`; // Unique name per question

                        tableHtml += `

                        <tr>

                            <td class="${index === 0 ? 'category-cell' : ''}">${index === 0 ? category : ''}</td>

                            <td>${question}</td>

                            <td>

                                <div class="star-rating">

                                    ${[6, 5, 4, 3, 2, 1].map(value => `

                                        <input type="radio" name="${questionName}" value="${value}" id="${questionName}star${value}" required>

                                        <label for="${questionName}star${value}">&#9733;</label>

                                    `).join('')}

                                </div>

                            </td>

                        </tr>`;

                    });

                }

                // Close the table structure

                tableHtml += `

                    </tbody>

                </table>`;

                questionsDiv.innerHTML = tableHtml;

            } catch (error) {

                console.error('Error in evaluateEmployee:', error);

                document.getElementById('questions').innerHTML = `

                    <div class="alert alert-danger">

                        <i class="fas fa-exclamation-triangle me-2"></i>

                        Failed to load evaluation questions. Please try again.

                    </div>

                `;

            }

        }

        // Define calculateAverage function

        function calculateAverage(category, evaluations) {

            // Filter evaluations for the current category

            const categoryEvaluations = evaluations.filter(evaluation => {

                // Ensure the question name matches the category

                return evaluation.question.toLowerCase().includes(category.toLowerCase());

            });

            if (categoryEvaluations.length === 0) {

                console.warn(`No evaluations found for category: ${category}`);

                return 0; // No evaluations for this category

            }

            // Calculate the average rating

            const total = categoryEvaluations.reduce((sum, evaluation) => sum + parseFloat(evaluation.rating), 0);

            const average = total / categoryEvaluations.length;

            console.log(`Category: ${category}, Average: ${average}`); // Debugging output

            return average;

        }

        // Function to update progress bar based on completion percentage

        function updateProgressBar(completionPercentage) {

            const progressContainer = document.querySelector('.progress-container');

            const greenProgressBar = document.querySelector('.progress-bar.complete');

            const progressLabel = greenProgressBar.querySelector('.progress-label');

            // Update the width of the green progress bar

            greenProgressBar.style.width = completionPercentage + '%';

            // Update the progress stats text

            document.querySelector('.progress-stats div:first-child').textContent = completionPercentage + '% Complete';

            document.querySelector('.progress-stats div:last-child').textContent = (100 - completionPercentage) + '% Pending';

        }

        // Your existing submitEvaluation function with improved UX

        function submitEvaluation() {

            const evaluations = [];

            const questionsDiv = document.getElementById('questions');

            // Check if all questions are answered

            const totalQuestions = questionsDiv.querySelectorAll('.star-rating').length;

            const answeredQuestions = questionsDiv.querySelectorAll('input[type="radio"]:checked').length;

            if (answeredQuestions < totalQuestions) {

                // Show validation message with animation

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

                alertDiv.className = 'alert alert-danger alert-dismissible fade show';

                alertDiv.innerHTML = `

                    <i class="fas fa-exclamation-circle me-2"></i>

                    Please complete all ${totalQuestions} questions before submitting. You have answered ${answeredQuestions} questions.

                    <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>

                `;

                // Insert at the top of the modal body

                const modalBody = document.querySelector('.modal-body');

                modalBody.insertBefore(alertDiv, modalBody.firstChild);

                // Scroll to the top of the modal

                modalBody.scrollTop = 0;

                return;

            }

            // Show loading state

            const submitBtn = document.querySelector('.btn-primary');

            const originalBtnHtml = submitBtn.innerHTML;

            submitBtn.innerHTML = '<span class="spinner-border spinner-border-sm me-2" role="status" aria-hidden="true"></span>Submitting...';

            submitBtn.disabled = true;

            // Collect all ratings

            questionsDiv.querySelectorAll('input[type="radio"]:checked').forEach(input => {

                evaluations.push({

                    question: input.name, // Question identifier

                    rating: input.value  // Rating value

                });

            });

            // Calculate category averages

            const categoryAverages = {

                QualityOfWork: calculateAverage('QualityOfWork', evaluations),

                CommunicationSkills: calculateAverage('CommunicationSkills', evaluations),

                Teamwork: calculateAverage('Teamwork', evaluations),

                Punctuality: calculateAverage('Punctuality', evaluations),

                Initiative: calculateAverage('Initiative', evaluations)

            };

            console.log('Category Averages:', categoryAverages);

            // Get the logged-in employee ID and department

            const employeeId = document.getElementById('employee\_id').value;

            const department = '<?php echo $department; ?>'; // Use the department from PHP

            // Submit the evaluation via AJAX

            $.ajax({

                type: 'POST',

                url: '../../employee\_db/supervisor/submit\_evaluation.php',

                data: {

                    employee\_id: currentEmployeeId,

                    employeeName: currentEmployeeName,

                    employeeRole: currentEmployeeRole,

                    categoryAverages: categoryAverages,

                    employeeId: employeeId,

                    department: department

                },

                success: function (response) {

                    console.log(response);

                    // Reset button state

                    submitBtn.innerHTML = originalBtnHtml;

                    submitBtn.disabled = false;

                    if (response === 'You have already evaluated this employee.') {

                        // Show error message

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

                        alertDiv.className = 'alert alert-warning alert-dismissible fade show';

                        alertDiv.innerHTML = `

                            <i class="fas fa-exclamation-triangle me-2"></i>

                            ${response}

                            <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>

                        `;

                        // Insert at the top of the modal body

                        const modalBody = document.querySelector('.modal-body');

                        modalBody.insertBefore(alertDiv, modalBody.firstChild);

                    } else {

                        // Add the evaluated employee's ID to the evaluatedEmployees array

                        evaluatedEmployees.push(currentEmployeeId);

                        // Disable the button for this employee on the page

                        const evaluateButton = document.querySelector(`button[onclick\*="${currentEmployeeId}"]`);

                        if (evaluateButton) {

                            evaluateButton.disabled = true;

                            evaluateButton.innerHTML = '<i class="fas fa-check-circle"></i> Evaluated';

                        }

                        // Update the status badge

                        const statusCell = evaluateButton.closest('tr').querySelector('td:nth-child(4)');

                        statusCell.innerHTML = `

                            <span class="status-badge completed">

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

                            </span>

                        `;

                        // Update the stats

                        const evaluatedCount = evaluatedEmployees.length;

                        const totalEmployees = <?php echo $totalEmployees; ?>;

                        const pendingCount = totalEmployees - evaluatedCount;

                        const completionPercentage = Math.round((evaluatedCount / totalEmployees) \* 100);

                        const pendingPercentage = 100 - completionPercentage;

                        // Update the stats cards

                        document.querySelector('.stat-card:nth-child(2) .stat-value').textContent = evaluatedCount;

                        document.querySelector('.stat-card:nth-child(3) .stat-value').textContent = pendingCount;

                        // Update the progress bar with the new percentage

                        updateProgressBar(completionPercentage);

                        // Show success message with toast

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

                        toastContainer.className = 'position-fixed bottom-0 end-0 p-3';

                        toastContainer.style.zIndex = '1050';

                        toastContainer.innerHTML = `

                            <div class="toast show" role="alert" aria-live="assertive" aria-atomic="true">

                                <div class="toast-header bg-success text-white">

                                    <i class="fas fa-check-circle me-2"></i>

                                    <strong class="me-auto">Success</strong>

                                    <button type="button" class="btn-close btn-close-white" data-bs-dismiss="toast" aria-label="Close"></button>

                                </div>

                                <div class="toast-body">

                                    Evaluation for ${currentEmployeeName} has been submitted successfully.

                                </div>

                            </div>

                        `;

                        document.body.appendChild(toastContainer);

                        // Auto-hide toast after 5 seconds

                        setTimeout(() => {

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

                            if (toast) {

                                toast.classList.remove('show');

                                setTimeout(() => {

                                    toastContainer.remove();

                                }, 500);

                            }

                        }, 5000);

                        // Hide the modal after submission

                        $('#evaluationModal').modal('hide');

                    }

                },

                error: function (err) {

                    console.error(err);

                    // Reset button state

                    submitBtn.innerHTML = originalBtnHtml;

                    submitBtn.disabled = false;

                    // Show error message

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

                    alertDiv.className = 'alert alert-danger alert-dismissible fade show';

                    alertDiv.innerHTML = `

                        <i class="fas fa-times-circle me-2"></i>

                        An error occurred while submitting the evaluation. Please try again.

                        <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>

                    `;

                    // Insert at the top of the modal body

                    const modalBody = document.querySelector('.modal-body');

                    modalBody.insertBefore(alertDiv, modalBody.firstChild);

                }

            });

        }

        // Search functionality

        document.querySelector('.search-input').addEventListener('input', function() {

            const searchTerm = this.value.toLowerCase();

            const rows = document.querySelectorAll('.employees-table tbody tr');

            rows.forEach(row => {

                const name = row.querySelector('td:first-child').textContent.toLowerCase();

                const position = row.querySelector('td:nth-child(2)').textContent.toLowerCase();

                const role = row.querySelector('td:nth-child(3)').textContent.toLowerCase();

                if (name.includes(searchTerm) || position.includes(searchTerm) || role.includes(searchTerm)) {

                    row.style.display = '';

                } else {

                    row.style.display = 'none';

                }

            });

        });

        // Initialize any tooltips and check progress bar on page load

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

            // Initialize tooltips if Bootstrap 5 is used

            var tooltipTriggerList = [].slice.call(document.querySelectorAll('[data-bs-toggle="tooltip"]'));

            var tooltipList = tooltipTriggerList.map(function (tooltipTriggerEl) {

                return new bootstrap.Tooltip(tooltipTriggerEl);

            });

            // Check if we need to update the progress bar color on page load

            const completionPercentage = <?php echo $completionPercentage; ?>;

            updateProgressBar(completionPercentage);

        });

    </script>

</body>

</html>