<?php

session\_start();

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

    header("Location: ../admin/adminlogin.php");

    exit();

}

// Include the database connection

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

// Define the values for role and department

$role = 'employee';

$department = 'IT Department';

// Check if it is the first week of the month

$currentDay = date('j'); // Current day of the month (1-31)

$isFirstWeek = ($currentDay <= 7); // First week is days 1-7

// Set the evaluation period to the previous month if it is the first week

if ($isFirstWeek) {

    $evaluationMonth = date('m', strtotime('last month')); // Previous month

    $evaluationYear = date('Y', strtotime('last month'));  // Year of the previous month

    $evaluationPeriod = date('F Y', strtotime('last month')); // Format: February 2024

    // Calculate the end date of the evaluation period (7th day of the current month)

    $evaluationEndDate = date('F j, Y', strtotime(date('Y-m-07'))); // Format: March 7, 2024

} else {

    // If it is not the first week, evaluations are closed

    $evaluationMonth = null;

    $evaluationYear = null;

    $evaluationPeriod = null;

    $evaluationEndDate = null;

}

// Fetch employee records where role is 'employee' and department is 'Administration Department'

$sql = "SELECT e\_id, firstname, lastname, role, position FROM employee\_register WHERE role = ? AND department = ?";

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

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

$stmt->execute();

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

// Fetch evaluations for this admin

$adminId = $\_SESSION['a\_id'];

$evaluatedEmployees = [];

$evalSql = "SELECT e\_id FROM admin\_evaluations WHERE a\_id = ?";

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

$evalStmt->bind\_param('i', $adminId);

$evalStmt->execute();

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

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

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

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

    }

}

// Fetch evaluation questions from the database for each category

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

$questions = [];

foreach ($categories as $category) {

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

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

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

    $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()) {

        $employees[] = $row;

    }

}

// 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>Evaluation</title>

    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" rel="stylesheet">

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

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

</head>

<body class="bg-dark text-light">

    <div class="container mt-5">

        <h2 class="text-center text-light mb-4">IT Department</h2>

        <!-- Display Evaluation Period -->

        <?php if ($isFirstWeek): ?>

            <p class="text-center text-warning">

                Evaluation is open for <?php echo $evaluationPeriod; ?> until <?php echo $evaluationEndDate; ?>.

            </p>

        <?php else: ?>

            <p class="text-center text-danger">

                Evaluations are closed. They will open in the first week of the next month.

            </p>

        <?php endif; ?>

        <!-- Employee Evaluation Table -->

        <div class="table-responsive">

            <table class="table table-striped table-hover text-dark">

                <thead class="thead-dark">

                    <tr>

                        <th>Name</th>

                        <th>Position</th>

                        <th>Role</th>

                        <th>Evaluation</th>

                    </tr>

                </thead>

                <tbody>

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

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

                            <tr>

                                <td class="text-light"><?php echo htmlspecialchars($employee['firstname'] . ' ' . $employee['lastname']); ?></td>

                                <td class="text-light"><?php echo htmlspecialchars($employee['position']); ?></td>

                                <td class="text-light"><?php echo htmlspecialchars($employee['role']); ?></td>

                                <td>

                                    <button class="btn btn-success"

                                        onclick="evaluateEmployee(<?php echo $employee['e\_id']; ?>, '<?php echo htmlspecialchars($employee['firstname'] . ' ' . $employee['lastname']); ?>', '<?php echo htmlspecialchars($employee['position']); ?>')"

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

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

                                    </button>

                                </td>

                            </tr>

                        <?php endforeach; ?>

                    <?php else: ?>

                        <tr><td class="text-light text-center" colspan="4">No employees found for evaluation in IT Department.</td></tr>

                    <?php endif; ?>

                </tbody>

            </table>

        </div>

    </div>

    <!-- 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 bg-primary text-white">

                    <h5 class="modal-title" id="employeeDetails"></h5>

                    <button type="button" class="close text-white" data-dismiss="modal" aria-label="Close">

                        <span aria-hidden="true">&times;</span>

                    </button>

                </div>

                <div class="modal-body">

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

                    <div class="text-dark" id="questions"></div>

                </div>

                <div class="modal-footer">

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

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

                </div>

            </div>

        </div>

    </div>

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

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.2/dist/js/bootstrap.bundle.min.js"></script>

    <script>

        let currentEmployeeId;

        let currentEmployeeName;

        let currentEmployeePosition;

        // The categories and questions fetched from the PHP script

        const questions = <?php echo json\_encode($questions); ?>;

        function evaluateEmployee(e\_id, employeeName, employeePosition) {

            currentEmployeeId = e\_id;

            currentEmployeeName = employeeName;

            currentEmployeePosition = employeePosition;

            const employeeDetails = `<strong>Name: ${employeeName} <br> Position: ${employeePosition}</strong>`;

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

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

            questionsDiv.innerHTML = '';

            // Start the table structure

            let tableHtml = `

            <table class="table table-bordered">

                <thead>

                    <tr>

                        <th>Category</th>

                        <th>Question</th>

                        <th>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>${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}">

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

                                `).join('')}

                            </div>

                        </td>

                    </tr>`;

                });

            }

            // Close the table structure

            tableHtml += `

                </tbody>

            </table>`;

            questionsDiv.innerHTML = tableHtml;

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

        }

        function submitEvaluation() {

            const evaluations = [];

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

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

                evaluations.push({

                    question: input.name,

                    rating: input.value

                });

            });

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

            if (evaluations.length !== totalQuestions) {

                alert('Please complete the evaluation before submitting.');

                return;

            }

            const categoryAverages = {

                QualityOfWork: calculateAverage('Quality of Work', evaluations),

                CommunicationSkills: calculateAverage('Communication Skills', evaluations),

                Teamwork: calculateAverage('Teamwork', evaluations),

                Punctuality: calculateAverage('Punctuality', evaluations),

                Initiative: calculateAverage('Initiative', evaluations)

            };

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

            const adminId = document.getElementById('a\_id').value;

            const department = 'IT Department';

            $.ajax({

                type: 'POST',

                url: '../db/submit\_evaluation.php',

                data: {

                    e\_id: currentEmployeeId,

                    employeeName: currentEmployeeName,

                    employeePosition: currentEmployeePosition,

                    categoryAverages: categoryAverages,

                    adminId: adminId,

                    department: department

                },

                success: function (response) {

                    console.log(response);

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

                        alert(response);

                    } else {

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

                        alert('Evaluation submitted successfully!');

                    }

                },

                error: function (err) {

                    console.error(err);

                    alert('An error occurred while submitting the evaluation.');

                }

            });

        }

        function calculateAverage(category, evaluations) {

            const categoryEvaluations = evaluations.filter(evaluation => evaluation.question.startsWith(category.replace(/\s/g, '')));

            if (categoryEvaluations.length === 0) {

                return 0;

            }

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

            return total / categoryEvaluations.length;

        }

    </script>

</body>

</html>