<?php

session\_start();

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

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

exit();

}

// Include database connection

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

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

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

// Fetch notifications for the employee

$notificationQuery = "SELECT \* FROM notifications WHERE employee\_id = ? ORDER BY created\_at DESC";

$notificationStmt = $conn->prepare($notificationQuery);

if ($notificationStmt === false) {

die("Error preparing notification query: " . $conn->error);

}

$notificationStmt->bind\_param("i", $employeeId);

$notificationStmt->execute();

$notifications = $notificationStmt->get\_result();

// Fetch employee info

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

$sql = "SELECT employee\_id, first\_name, middle\_name, last\_name, birthdate, email, role, position, department, phone\_number, address, pfp FROM employee\_register WHERE employee\_id = ?";

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

if ($stmt === false) {

die("Error preparing employee info query: " . $conn->error);

}

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

$stmt->execute();

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

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

function calculateProgressCircle($averageScore) {

return ($averageScore / 10) \* 100;

}

function getTopEmployeesByCriterion($conn, $criterion, $criterionLabel, $index) {

// Whitelist allowed criteria to avoid SQL injection

$allowedCriteria = ['quality', 'communication\_skills', 'teamwork', 'punctuality', 'initiative'];

if (!in\_array($criterion, $allowedCriteria)) {

echo "<p>Invalid criterion selected</p>";

return;

}

// SQL query to fetch the highest average score for each employee

$sql = "SELECT e.employee\_id, e.first\_name, e.last\_name, e.department, e.pfp, e.email,

AVG(ae.$criterion) AS avg\_score,

AVG(ae.quality) AS avg\_quality,

AVG(ae.communication\_skills) AS avg\_communication,

AVG(ae.teamwork) AS avg\_teamwork,

AVG(ae.punctuality) AS avg\_punctuality,

AVG(ae.initiative) AS avg\_initiative

FROM employee\_register e

JOIN evaluations ae ON e.employee\_id = ae.employee\_id

GROUP BY e.employee\_id

ORDER BY avg\_score DESC

LIMIT 5"; // Fetch top 5 employees

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

if ($stmt === false) {

die("Error preparing top employees query: " . $conn->error);

}

$stmt->execute();

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

// Path to the default profile picture

$defaultPfpPath = '../../img/defaultpfp.jpg';

$defaultPfp = base64\_encode(file\_get\_contents($defaultPfpPath));

// Output the awardee's information for each criterion

?>

<div class='category' id='category-<?php echo $index; ?>'>

<h3 class='text-center mt-4'><?php echo $criterionLabel; ?></h3>

<?php

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

$employeeIndex = 0;

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

$employeeIndex++;

?>

<div class='employee-card' id='employee-<?php echo $index; ?>-<?php echo $employeeIndex; ?>' style='display: none;'>

<?php

// Check if profile picture exists, else use the default picture

if (!empty($row['pfp'])) {

$pfp = base64\_encode($row['pfp']); // Assuming pfp is a BLOB

} else {

$pfp = $defaultPfp;

}

// Calculate percentage for the progress circle

$scorePercentage = calculateProgressCircle($row['avg\_score']);

?>

<div class='metrics-container'>

<!-- Left metrics -->

<div class='metrics-column'>

<div class='metric-box fade-in'>

<span class='metric-label'>Quality of Work</span>

<span class='metric-value'><?php echo round($row['avg\_quality'], 2); ?></span>

<div class='star-rating' style='margin-top: 5px;'>

<?php

// Get the exact quality score

$qualityScore = $row['avg\_quality'];

for ($i = 1; $i <= 6; $i++) {

if ($i <= floor($qualityScore)) {

echo "<i class='bi bi-star-fill' style='color: gold;'></i>";

} elseif ($i == ceil($qualityScore)) {

$decimalPart = $qualityScore - floor($qualityScore);

if ($decimalPart >= 0.2 && $decimalPart <= 0.9) {

echo "<i class='bi bi-star-half' style='color: gold;'></i>";

} else {

echo "<i class='bi bi-star' style='color: lightgray;'></i>";

}

} else {

echo "<i class='bi bi-star' style='color: lightgray;'></i>";

}

}

?>

</div>

</div>

<div class='metric-box fade-in' style='animation-delay: 0.2s;'>

<span class='metric-label'>Communication Skills</span>

<span class='metric-value'><?php echo round($row['avg\_communication'], 2); ?></span>

<div class='star-rating' style='margin-top: 5px;'>

<?php

// Get the exact communication skills score

$communicationScore = $row['avg\_communication'];

for ($i = 1; $i <= 6; $i++) {

if ($i <= floor($communicationScore)) {

echo "<i class='bi bi-star-fill' style='color: gold;'></i>";

} elseif ($i == ceil($communicationScore)) {

$decimalPart = $communicationScore - floor($communicationScore);

if ($decimalPart >= 0.2 && $decimalPart <= 0.9) {

echo "<i class='bi bi-star-half' style='color: gold;'></i>";

} else {

echo "<i class='bi bi-star' style='color: lightgray;'></i>";

}

} else {

echo "<i class='bi bi-star' style='color: lightgray;'></i>";

}

}

?>

</div>

</div>

</div>

<!-- Center profile section -->

<div class='profile-section'>

<div class='progress-circle-container'>

<div class='progress-circle' data-progress='<?php echo $scorePercentage; ?>'>

<div class='profile-image-container'>

<?php if (!empty($pfp)) { ?>

<img src='data:image/jpeg;base64,<?php echo $pfp; ?>' alt='Profile Picture' class='profile-image'>

<?php } ?>

</div>

</div>

</div>

<div class='profile-info'>

<h2 class='employee-name'><?php echo htmlspecialchars($row['first\_name'] . ' ' . $row['last\_name']); ?></h2>

<p class='department-name'><?php echo htmlspecialchars($row['department']); ?></p>

</div>

<div class='employee-id fade-in' style='animation-delay: 0.8s;'>

Employee ID: <?php echo htmlspecialchars($row['employee\_id']); ?>

</div>

<!-- New metric box below employee ID -->

<div class='metric-box fade-in' style='animation-delay: 0.8s;'>

<span class='metric-label'>Initiative</span>

<span class='metric-value'><?php echo round($row['avg\_initiative'], 2); ?></span>

<div class='star-rating' style='margin-top: 5px;'>

<?php

// Get the exact initiative score

$initiativeScore = $row['avg\_initiative'];

for ($i = 1; $i <= 6; $i++) {

if ($i <= floor($initiativeScore)) {

echo "<i class='bi bi-star-fill' style='color: gold;'></i>";

} elseif ($i == ceil($initiativeScore)) {

$decimalPart = $initiativeScore - floor($initiativeScore);

if ($decimalPart >= 0.2 && $decimalPart <= 0.9) {

echo "<i class='bi bi-star-half' style='color: gold;'></i>";

} else {

echo "<i class='bi bi-star' style='color: lightgray;'></i>";

}

} else {

echo "<i class='bi bi-star' style='color: lightgray;'></i>";

}

}

?>

</div>

</div>

<!-- Add buttons for comments and reactions -->

<div class='comment-reaction-buttons'>

<button id="comment-button-<?php echo $row['employee\_id']; ?>" class='btn btn-primary comment-btn' onclick='openCommentModal(<?php echo $row['employee\_id']; ?>)'>

<i class="bi bi-chat-dots-fill me-1"></i> Comments (<span id="comment-count-<?php echo $row['employee\_id']; ?>">0</span>)

</button>

<div class='comment-input-container'>

<button class="btn btn-primary react-btn" id="react-button-<?php echo $row['employee\_id']; ?>">

<i class="bi bi-emoji-smile-fill me-1"></i> React (<span id="reaction-count-<?php echo $row['employee\_id']; ?>">0</span>)

</button>

<div id='reaction-menu-<?php echo $row['employee\_id']; ?>' class='reaction-dropdown'>

<button onclick='selectReaction("👍 Like", <?php echo $row['employee\_id']; ?>)'>👍</button>

<button onclick='selectReaction("❤️ Heart", <?php echo $row['employee\_id']; ?>)'>❤️</button>

<button onclick='selectReaction("😎 Awesome", <?php echo $row['employee\_id']; ?>)'>😎</button>

<button onclick='selectReaction("😮 Wow", <?php echo $row['employee\_id']; ?>)'>😮</button>

</div>

</div>

</div>

</div>

<!-- Right metrics -->

<div class='metrics-column'>

<div class='metric-box fade-in' style='animation-delay: 0.4s;'>

<span class='metric-label'>Teamwork</span>

<span class='metric-value'><?php echo round($row['avg\_teamwork'], 2); ?></span>

<div class='star-rating' style='margin-top: 5px;'>

<?php

// Get the exact teamwork score

$teamworkScore = $row['avg\_teamwork'];

for ($i = 1; $i <= 6; $i++) {

if ($i <= floor($teamworkScore)) {

echo "<i class='bi bi-star-fill' style='color: gold;'></i>";

} elseif ($i == ceil($teamworkScore)) {

$decimalPart = $teamworkScore - floor($teamworkScore);

if ($decimalPart >= 0.2 && $decimalPart <= 0.9) {

echo "<i class='bi bi-star-half' style='color: gold;'></i>";

} else {

echo "<i class='bi bi-star' style='color: lightgray;'></i>";

}

} else {

echo "<i class='bi bi-star' style='color: lightgray;'></i>";

}

}

?>

</div>

</div>

<div class='metric-box fade-in' style='animation-delay: 0.6s;'>

<span class='metric-label'>Punctuality</span>

<span class='metric-value'><?php echo round($row['avg\_punctuality'], 2); ?></span>

<div class='star-rating' style='margin-top: 5px;'>

<?php

// Get the exact punctuality score

$punctualityScore = $row['avg\_punctuality'];

for ($i = 1; $i <= 6; $i++) {

if ($i <= floor($punctualityScore)) {

echo "<i class='bi bi-star-fill' style='color: gold;'></i>";

} elseif ($i == ceil($punctualityScore)) {

$decimalPart = $punctualityScore - floor($punctualityScore);

if ($decimalPart >= 0.2 && $decimalPart <= 0.9) {

echo "<i class='bi bi-star-half' style='color: gold;'></i>";

} else {

echo "<i class='bi bi-star' style='color: lightgray;'></i>";

}

} else {

echo "<i class='bi bi-star' style='color: lightgray;'></i>";

}

}

?>

</div>

</div>

</div>

</div>

</div>

<?php

}

} else {

?>

<p class='text-center'>No outstanding employees found for <?php echo $criterionLabel; ?>.</p>

<?php

}

?>

</div>

<?php

$stmt->close();

}

function getAllEmployees($conn) {

$sql = "SELECT e.employee\_id, e.first\_name, e.last\_name, e.department, e.pfp, e.email,

AVG(ae.quality) AS avg\_quality,

AVG(ae.communication\_skills) AS avg\_communication,

AVG(ae.teamwork) AS avg\_teamwork,

AVG(ae.punctuality) AS avg\_punctuality,

AVG(ae.initiative) AS avg\_initiative

FROM employee\_register e

JOIN evaluations ae ON e.employee\_id = ae.employee\_id

GROUP BY e.employee\_id

ORDER BY e.employee\_id ASC"; // Fetch all employees

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

if ($stmt === false) {

die("Error preparing all employees query: " . $conn->error);

}

$stmt->execute();

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

// Path to the default profile picture

$defaultPfpPath = '../../img/defaultpfp.jpg';

$defaultPfp = base64\_encode(file\_get\_contents($defaultPfpPath));

// Output the employee's information

?>

<div class='category' id='category-all' style='display: none;'>

<h3 class='text-center mt-4'>All Employees</h3>

<?php

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

$employeeIndex = 0;

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

$employeeIndex++;

?>

<div class='employee-card' id='employee-all-<?php echo $employeeIndex; ?>' style='display: none;'>

<?php

// Check if profile picture exists, else use the default picture

if (!empty($row['pfp'])) {

$pfp = base64\_encode($row['pfp']); // Assuming pfp is a BLOB

} else {

$pfp = $defaultPfp;

}

// Calculate percentage for the progress circle

$scorePercentage = calculateProgressCircle(($row['avg\_quality'] + $row['avg\_communication'] + $row['avg\_teamwork'] + $row['avg\_punctuality'] + $row['avg\_initiative']) / 5);

?>

<div class='metrics-container'>

<!-- Left metrics -->

<div class='metrics-column'>

<div class='metric-box fade-in'>

<span class='metric-label'>Quality of Work</span>

<span class='metric-value'><?php echo round($row['avg\_quality'], 2); ?></span>

<div class='star-rating' style='margin-top: 5px;'>

<?php

// Get the exact quality score

$qualityScore = $row['avg\_quality'];

for ($i = 1; $i <= 6; $i++) {

if ($i <= floor($qualityScore)) {

echo "<i class='bi bi-star-fill' style='color: gold;'></i>";

} elseif ($i == ceil($qualityScore)) {

$decimalPart = $qualityScore - floor($qualityScore);

if ($decimalPart >= 0.2 && $decimalPart <= 0.9) {

echo "<i class='bi bi-star-half' style='color: gold;'></i>";

} else {

echo "<i class='bi bi-star' style='color: lightgray;'></i>";

}

} else {

echo "<i class='bi bi-star' style='color: lightgray;'></i>";

}

}

?>

</div>

</div>

<div class='metric-box fade-in' style='animation-delay: 0.2s;'>

<span class='metric-label'>Communication Skills</span>

<span class='metric-value'><?php echo round($row['avg\_communication'], 2); ?></span>

<div class='star-rating' style='margin-top: 5px;'>

<?php

// Get the exact communication skills score

$communicationScore = $row['avg\_communication'];

for ($i = 1; $i <= 6; $i++) {

if ($i <= floor($communicationScore)) {

echo "<i class='bi bi-star-fill' style='color: gold;'></i>";

} elseif ($i == ceil($communicationScore)) {

$decimalPart = $communicationScore - floor($communicationScore);

if ($decimalPart >= 0.2 && $decimalPart <= 0.9) {

echo "<i class='bi bi-star-half' style='color: gold;'></i>";

} else {

echo "<i class='bi bi-star' style='color: lightgray;'></i>";

}

} else {

echo "<i class='bi bi-star' style='color: lightgray;'></i>";

}

}

?>

</div>

</div>

</div>

<!-- Center profile section -->

<div class='profile-section'>

<div class='progress-circle-container'>

<div class='progress-circle' data-progress='<?php echo $scorePercentage; ?>'>

<div class='profile-image-container'>

<?php if (!empty($pfp)) { ?>

<img src='data:image/jpeg;base64,<?php echo $pfp; ?>' alt='Profile Picture' class='profile-image'>

<?php } ?>

</div>

</div>

</div>

<div class='profile-info'>

<h2 class='employee-name'><?php echo htmlspecialchars($row['first\_name'] . ' ' . $row['last\_name']); ?></h2>

<p class='department-name'><?php echo htmlspecialchars($row['department']); ?></p>

</div>

<div class='employee-id fade-in' style='animation-delay: 0.8s;'>

Employee ID: <?php echo htmlspecialchars($row['employee\_id']); ?>

</div>

<!-- New metric box below employee ID -->

<div class='metric-box fade-in' style='animation-delay: 0.8s;'>

<span class='metric-label'>Initiative</span>

<span class='metric-value'><?php echo round($row['avg\_initiative'], 2); ?></span>

<div class='star-rating' style='margin-top: 5px;'>

<?php

// Get the exact initiative score

$initiativeScore = $row['avg\_initiative'];

for ($i = 1; $i <= 6; $i++) {

if ($i <= floor($initiativeScore)) {

echo "<i class='bi bi-star-fill' style='color: gold;'></i>";

} elseif ($i == ceil($initiativeScore)) {

$decimalPart = $initiativeScore - floor($initiativeScore);

if ($decimalPart >= 0.2 && $decimalPart <= 0.9) {

echo "<i class='bi bi-star-half' style='color: gold;'></i>";

} else {

echo "<i class='bi bi-star' style='color: lightgray;'></i>";

}

} else {

echo "<i class='bi bi-star' style='color: lightgray;'></i>";

}

}

?>

</div>

</div>

</div>

<!-- Right metrics -->

<div class='metrics-column'>

<div class='metric-box fade-in' style='animation-delay: 0.4s;'>

<span class='metric-label'>Teamwork</span>

<span class='metric-value'><?php echo round($row['avg\_teamwork'], 2); ?></span>

<div class='star-rating' style='margin-top: 5px;'>

<?php

// Get the exact teamwork score

$teamworkScore = $row['avg\_teamwork'];

for ($i = 1; $i <= 6; $i++) {

if ($i <= floor($teamworkScore)) {

echo "<i class='bi bi-star-fill' style='color: gold;'></i>";

} elseif ($i == ceil($teamworkScore)) {

$decimalPart = $teamworkScore - floor($teamworkScore);

if ($decimalPart >= 0.2 && $decimalPart <= 0.9) {

echo "<i class='bi bi-star-half' style='color: gold;'></i>";

} else {

echo "<i class='bi bi-star' style='color: lightgray;'></i>";

}

} else {

echo "<i class='bi bi-star' style='color: lightgray;'></i>";

}

}

?>

</div>

</div>

<div class='metric-box fade-in' style='animation-delay: 0.6s;'>

<span class='metric-label'>Punctuality</span>

<span class='metric-value'><?php echo round($row['avg\_punctuality'], 2); ?></span>

<div class='star-rating' style='margin-top: 5px;'>

<?php

// Get the exact punctuality score

$punctualityScore = $row['avg\_punctuality'];

for ($i = 1; $i <= 6; $i++) {

if ($i <= floor($punctualityScore)) {

echo "<i class='bi bi-star-fill' style='color: gold;'></i>";

} elseif ($i == ceil($punctualityScore)) {

$decimalPart = $punctualityScore - floor($punctualityScore);

if ($decimalPart >= 0.2 && $decimalPart <= 0.9) {

echo "<i class='bi bi-star-half' style='color: gold;'></i>";

} else {

echo "<i class='bi bi-star' style='color: lightgray;'></i>";

}

} else {

echo "<i class='bi bi-star' style='color: lightgray;'></i>";

}

}

?>

</div>

</div>

</div>

</div>

</div>

<?php

}

} else {

?>

<p class='text-center'>No employees found.</p>

<?php

}

?>

</div>

<?php

$stmt->close();

}

function getComments($conn, $employeeId) {

$sql = "SELECT comment, comment\_time FROM comments WHERE employee\_id = ? ORDER BY comment\_time DESC";

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

if ($stmt === false) {

die("Error preparing comments query: " . $conn->error);

}

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

$stmt->execute();

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

$comments = [];

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

$comments[] = $row;

}

$stmt->close();

return $comments;

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Outstanding Employees</title>

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

<link href="https://cdn.jsdelivr.net/npm/bootstrap-icons/font/bootstrap-icons.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>

<style>

.card {

border: 2px solid #ddd;

border-radius: 10px;

box-shadow: 0 2px 4px rgba(233, 233, 233, 0.1);

padding: 10px;

background-color: #f9f9f9;

}

.card-img {

border-radius: 10px;

}

.card-body {

padding-left: 10px;

}

.card-title {

font-size: 20px;

font-weight: bold;

color: #333;

}

.card-text {

font-size: 14px;

}

.category {

display: none;

}

.btn {

transition: transform 0.3s, background-color 0.3s;

border-radius: 20px;

padding: 5px 10px;

font-size: 14px;

}

.btn:hover {

transform: translateY(-2px);

}

.emoji-container {

display: none;

gap: 15px;

cursor: pointer;

margin-top: 15px;

flex-wrap: wrap;

}

.emoji {

font-size: 30px;

transition: transform 0.2s ease;

padding: 10px;

}

.emoji:hover {

transform: scale(1.2);

}

.reaction {

margin-top: 15px;

font-size: 18px;

color: #333;

}

.saved-reaction {

margin-top: 10px;

color: #007bff;

}

.open-btn {

font-size: 16px;

padding: 8px 16px;

background-color: #007bff;

color: white;

border: none;

cursor: pointer;

border-radius: 5px;

transition: background-color 0.3s;

}

.open-btn:hover {

background-color: #0056b3;

}

.reaction-count {

font-size: 18px;

color: #333;

margin-top: 10px;

display: flex;

gap: 15px;

}

.reaction-item {

display: flex;

align-items: center;

gap: 5px;

}

.selected-emoji {

font-size: 50px;

margin-top: 20px;

}

.employee-card {

background: rgb(33, 37, 41);

border-radius: 20px;

padding: 2rem;

max-width: 1200px;

margin: 2rem auto;

color: rgba(248, 249, 250);

}

.dashboard-title {

text-align: center;

font-size: 1.5rem;

font-weight: bold;

margin-bottom: 2rem;

color: white;

}

.metrics-container {

display: grid;

grid-template-columns: 1fr 1.5fr 1fr;

gap: 2rem;

margin-bottom: 2rem;

}

.metrics-column {

display: flex;

flex-direction: column;

gap: 1rem;

}

.metric-box {

background: rgb(16, 17, 18);

border-radius: 15px;

padding: 1rem;

display: flex;

flex-direction: column;

margin-top: 2rem;

margin-bottom: 2rem;

width: 100%;

}

.metric-label {

color: rgba(248, 249, 250);

font-size: 0.875rem;

margin-bottom: 0.5rem;

}

.metric-value {

font-size: 1.875rem;

font-weight: bold;

}

.profile-section {

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

}

.progress-circle-container {

position: relative;

width: 200px;

height: 200px;

}

.progress-circle {

width: 100%;

height: 100%;

border-radius: 50%;

position: relative;

background: #1e3a8a;

}

.profile-image-container {

position: absolute;

top: 50%;

left: 50%;

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

width: 100px;

height: 100px;

border-radius: 50%;

border: 4px solid #22d3ee;

overflow: hidden;

}

.profile-image {

width: 100%;

height: 100%;

object-fit: cover;

}

.profile-info {

text-align: center;

margin-top: 1rem;

}

.employee-name {

font-size: 1.25rem;

font-weight: bold;

margin-bottom: 0.25rem;

}

.department-name {

color: #22d3ee;

font-size: 0.875rem;

}

.employee-id {

text-align: center;

color: rgba(255, 255, 255, 0.6);

margin-top: 1rem;

}

@keyframes fadeIn {

from {

opacity: 0;

transform: translateY(20px);

}

to {

opacity: 1;

transform: translateY(0);

}

}

.fade-in {

animation: fadeIn 0.5s ease-out forwards;

opacity: 0;

}

@keyframes progressCircle {

from {

stroke-dashoffset: 628;

}

to {

stroke-dashoffset: var(--progress);

}

}

@media (max-width: 768px) {

.metrics-container {

grid-template-columns: 1fr;

}

.employee-card {

margin: 1rem;

padding: 1rem;

}

}

.progress-ring {

position: absolute;

top: 0;

left: 0;

transform: rotate(-90deg);

}

.progress-ring\_\_circle {

transition: stroke-dashoffset 0.5s ease-out;

}

.comment-reaction-buttons {

display: flex;

gap: 10px;

justify-content: center;

margin-top: 20px;

}

.comment-reaction-buttons .comment-btn,

.comment-reaction-buttons .react-btn {

display: flex;

align-items: center;

padding: 1px;

font-size: 1rem;

border-radius: 20px;

height: 40px; /\* Adjust height to match \*/

}

.comment-reaction-buttons {

position: relative;

}

.comment-reaction-buttons .reaction-dropdown {

display: block;

}

.react-btn {

position: relative;

transition: background-color 0.3s;

}

.react-btn:hover + .reaction-dropdown,

.reaction-dropdown:hover {

display: flex;

opacity: 1;

}

.reaction-dropdown {

display: none;

position: absolute;

background-color: #343a40;

border-radius: 30px;

padding: 8px;

box-shadow: 0 4px 15px rgba(0, 0, 0, 0.2);

z-index: 10;

opacity: 0;

transition: opacity 0.3s ease-in-out;

transform: translateY(-100%);

left: 0;

top: -10px;

}

.reaction-dropdown button {

background: none;

border: none;

cursor: pointer;

font-size: 20px;

margin: 0 5px;

transition: transform 0.2s;

}

.reaction-dropdown button:hover {

transform: scale(1.2);

}

/\* Modern Comment Modal Styling \*/

.modal-right {

position: fixed;

top: 0;

right: 0;

width: 400px;

height: 100vh;

margin-top: 5rem;

background: #212529;

box-shadow: -5px 0 15px rgba(0, 0, 0, 0.2);

padding: 0;

display: none;

z-index: 1000;

overflow: hidden;

transition: transform 0.3s ease-in-out;

transform: translateX(100%);

}

.modal-right.show {

transform: translateX(0);

display: flex;

flex-direction: column;

}

.modal-header {

display: flex;

justify-content: space-between;

align-items: center;

padding: 10px;

border-bottom: 1px solid #343a40;

background-color: #343a40;

}

.modal-header h5 {

color: white;

font-weight: 600;

margin: 0;

font-size: 1.25rem;

}

.modal-body {

flex: 1;

overflow-y: auto;

padding: 20px;

display: flex;

flex-direction: column;

}

.close-btn {

background: none;

border: none;

color: rgba(255, 255, 255, 0.7);

font-size: 24px;

cursor: pointer;

transition: color 0.2s;

}

.close-btn:hover {

color: white;

}

.comment-input-container {

position: relative;

margin-bottom: 20px;

}

.comment-input {

width: 100%;

padding: 12px 15px;

border: 1px solid #343a40;

border-radius: 30px;

background-color: #343a40;

color: white;

font-size: 14px;

transition: all 0.3s;

}

.comment-input:focus {

outline: none;

border-color: #0d6efd;

box-shadow: 0 0 0 2px rgba(13, 110, 253, 0.25);

}

.comment-input::placeholder {

color: rgba(255, 255, 255, 0.5);

}

.comment-input-container .comment-post-btn {

position: absolute;

right: 10px;

top: 50%;

transform: translateY(-50%);

background: #0d6efd;

border: none;

color: white;

border-radius: 50%;

width: 32px;

height: 32px;

display: flex;

align-items: center;

justify-content: center;

cursor: pointer;

transition: background-color 0.2s;

}

.comment-input-container .comment-post-btn:hover {

background-color: #0b5ed7;

}

.comment-input-container .comment-post-btn i {

font-size: 16px;

}

.comment-input-container .comment-post-btn {

position: absolute;

right: 10px;

top: 50%;

transform: translateY(-50%);

background: #0d6efd;

border: none;

color: white;

border-radius: 50%;

width: 32px;

height: 32px;

display: flex;

align-items: center;

justify-content: center;

cursor: pointer;

transition: background-color 0.2s;

}

.comment-input-container .comment-post-btn:hover {

background-color: #0b5ed7;

}

.comment-input-container .comment-post-btn i {

font-size: 16px;

}

.comment-list {

display: flex;

flex-direction: column;

gap: 15px;

margin-top: 10px;

overflow-y: auto;

padding-right: 5px;

}

.comment {

background-color: #343a40;

border-radius: 15px;

padding: 15px;

position: relative;

animation: fadeIn 0.3s ease-out forwards;

}

.comment-header {

display: flex;

justify-content: space-between;

margin-bottom: 8px;

}

.comment-author {

font-weight: 600;

color: #0d6efd;

}

.comment-time {

font-size: 12px;

color: rgba(255, 255, 255, 0.5);

}

.comment-content {

color: white;

word-break: break-word;

}

.comment-actions {

display: flex;

gap: 15px;

margin-top: 10px;

font-size: 12px;

}

.comment-action {

color: rgba(255, 255, 255, 0.6);

cursor: pointer;

transition: color 0.2s;

}

.comment-action:hover {

color: white;

}

.navigation-buttons {

display: flex;

justify-content: space-between;

margin-top: 20px;

gap: 10px;

}

.navigation-buttons .btn {

width: 5%;

padding: 10px;

margin-left: 350px;

margin-right: 350px;

}

/\* Scrollbar styling \*/

.comment-list::-webkit-scrollbar {

width: 6px;

}

.comment-list::-webkit-scrollbar-track {

background: #212529;

}

.comment-list::-webkit-scrollbar-thumb {

background-color: #495057;

border-radius: 3px;

}

.comment-list::-webkit-scrollbar-thumb:hover {

background-color: #6c757d;

}

/\* Empty state for comments \*/

.empty-comments {

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

padding: 40px 0;

color: rgba(255, 255, 255, 0.5);

text-align: center;

}

.empty-comments i {

font-size: 48px;

margin-bottom: 15px;

opacity: 0.5;

}

.empty-comments p {

font-size: 16px;

}

</style>

</head>

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

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

<div id="layoutSidenav">

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

<div id="layoutSidenav\_content">

<main class="container-fluid position-relative bg-black">

<div class="container text-light">

<!-- Top Employees for Different Criteria -->

<?php getTopEmployeesByCriterion($conn, 'quality', 'Quality of Work', 1); ?>

<?php getTopEmployeesByCriterion($conn, 'communication\_skills', 'Communication Skills', 2); ?>

<?php getTopEmployeesByCriterion($conn, 'teamwork', 'Teamwork', 3); ?>

<?php getTopEmployeesByCriterion($conn, 'punctuality', 'Punctuality', 4); ?>

<?php getTopEmployeesByCriterion($conn, 'initiative', 'Initiative', 5); ?>

<?php getAllEmployees($conn); ?>

</main>

<div class="modal fade" id="logoutModal" tabindex="-1" aria-labelledby="logoutModalLabel" aria-hidden="true">

<div class="modal-dialog modal-dialog-centered">

<div class="modal-content bg-dark text-light">

<div class="modal-header border-bottom border-secondary">

<h5 class="modal-title" id="logoutModalLabel">Confirm Logout</h5>

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

</div>

<div class="modal-body">

Are you sure you want to log out?

</div>

<div class="modal-footer border-top border-secondary">

<button type="button" class="btn border-secondary text-light" data-bs-dismiss="modal">Cancel</button>

<form action="../../employee/logout.php" method="POST">

<button type="submit" class="btn btn-danger">Logout</button>

</form>

</div>

</div>

</div>

</div>

<div class='navigation-buttons'>

<button class='btn btn-primary' onclick='showPreviousEmployee(1)'>Previous</button>

<button class='btn btn-primary' onclick='showNextEmployee(1)'>Next</button>

</div>

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

</div>

<div class="modal fade" id="loadingModal" tabindex="-1" aria-labelledby="loadingModalLabel" aria-hidden="true">

<div class="modal-dialog modal-dialog-centered">

<div class="modal-content bg-transparent border-0">

<div class="modal-body d-flex flex-column align-items-center justify-content-center">

<!-- Bouncing coin spinner -->

<div class="coin-spinner"></div>

<div class="mt-3 text-light fw-bold">Please wait...</div>

</div>

</div>

</div>

</div>

<!-- Modal for displaying comments -->

<div id="commentModal" class="modal-right">

<div class="modal-header">

<h5><i class="bi bi-chat-dots-fill me-2"></i>Comments</h5>

<button class="close-btn" onclick="closeModal('commentModal')">&times;</button>

</div>

<div class="modal-body">

<div class="comment-input-container">

<input type="text" class="comment-input" placeholder="Write a comment..." onkeypress="postComment(event, this)">

<button class="comment-post-btn" onclick="postComment({key:'Enter'}, document.querySelector('.comment-input'))">

<i class="bi bi-send-fill"></i>

</button>

</div>

<div class="comment-list">

<!-- Comments will be loaded here -->

</div>

</div>

</div>

<!-- Reaction Modal -->

<div id="reactionModal" class="modal">

<div class="modal-content">

<span class="close" onclick="closeModal('reactionModal')">&times;</span>

<div class="modal-body"></div>

</div>

</div>

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

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

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

<script>

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

const buttons = document.querySelectorAll('.loading');

const loadingModal = new bootstrap.Modal(document.getElementById('loadingModal'));

// Loop through each button and add a click event listener

buttons.forEach(button => {

button.addEventListener('click', function (event) {

// Show the loading modal

loadingModal.show();

// Disable the button to prevent multiple clicks

this.classList.add('disabled');

// Handle form submission buttons

if (this.closest('form')) {

event.preventDefault(); // Prevent the default form submit

// Submit the form after a short delay

setTimeout(() => {

this.closest('form').submit();

}, 1500);

}

// Handle links

else if (this.tagName.toLowerCase() === 'a') {

event.preventDefault(); // Prevent the default link behavior

// Redirect after a short delay

setTimeout(() => {

window.location.href = this.href;

}, 1500);

}

});

});

// Hide the loading modal when navigating back and enable buttons again

window.addEventListener('pageshow', function (event) {

if (event.persisted) { // Check if the page was loaded from cache (back button)

loadingModal.hide();

// Re-enable all buttons when coming back

buttons.forEach(button => {

button.classList.remove('disabled');

});

}

});

});

let currentCategoryIndex = 1;

const totalCategories = 6; // Assuming there are 6 categories (5 criteria + all employees)

let currentEmployeeIndex = {};

function showNextCategory() {

// Hide the current category

document.getElementById(`category-${currentCategoryIndex}`).style.display = 'none';

// Update to the next category index, loop back to 1 if at the end

currentCategoryIndex = (currentCategoryIndex % totalCategories) + 1;

// Show the next category

document.getElementById(`category-${currentCategoryIndex}`).style.display = 'block';

}

function showPreviousCategory() {

// Hide the current category

document.getElementById(`category-${currentCategoryIndex}`).style.display = 'none';

// Update to the previous category index, loop back to totalCategories if at the start

currentCategoryIndex = (currentCategoryIndex - 1) || totalCategories;

// Show the previous category

document.getElementById(`category-${currentCategoryIndex}`).style.display = 'block';

}

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

// Initialize progress circles

const circles = document.querySelectorAll('.progress-circle');

circles.forEach(circle => {

const progress = circle.getAttribute('data-progress');

const circumference = 2 \* Math.PI \* 90; // for r=90

const strokeDashoffset = circumference - (progress / 100) \* circumference;

// Create SVG element

const svg = document.createElementNS('http://www.w3.org/2000/svg', 'svg');

svg.setAttribute('class', 'progress-ring');

svg.setAttribute('width', '200');

svg.setAttribute('height', '200');

// Create a circle for the background with the color rgb(16, 17, 18)

const backgroundCircle = document.createElementNS('http://www.w3.org/2000/svg', 'circle');

backgroundCircle.setAttribute('cx', '100'); // Center X coordinate

backgroundCircle.setAttribute('cy', '100'); // Center Y coordinate

backgroundCircle.setAttribute('r', '90'); // Radius

backgroundCircle.setAttribute('fill', 'rgb(16, 17, 18)'); // Background color

// Append the background circle to the SVG

svg.appendChild(backgroundCircle);

// Now you can add other elements (e.g., a progress circle) on top of the background

// Create a progress circle (example)

const progressCircle = document.createElementNS('http://www.w3.org/2000/svg', 'circle');

progressCircle.setAttribute('cx', '100');

progressCircle.setAttribute('cy', '100');

progressCircle.setAttribute('r', '90');

progressCircle.setAttribute('fill', 'none');

progressCircle.setAttribute('stroke', 'rgb(16, 17, 18)');

progressCircle.setAttribute('stroke-width', '20');

// Append the progress circle

svg.appendChild(progressCircle);

const circleElement = document.createElementNS('http://www.w3.org/2000/svg', 'circle');

circleElement.setAttribute('class', 'progress-ring\_\_circle');

circleElement.setAttribute('stroke', '#22d3ee');

circleElement.setAttribute('stroke-width', '4');

circleElement.setAttribute('fill', 'transparent');

circleElement.setAttribute('r', '90');

circleElement.setAttribute('cx', '100');

circleElement.setAttribute('cy', '100');

circleElement.style.strokeDasharray = `${circumference} ${circumference}`;

circleElement.style.strokeDashoffset = strokeDashoffset;

svg.appendChild(circleElement);

circle.insertBefore(svg, circle.firstChild);

});

});

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

const reactButtons = document.querySelectorAll('.react-btn');

reactButtons.forEach(button => {

const menu = button.nextElementSibling;

let reactionAlertShown = false; // Flag to track if the alert has been shown

button.addEventListener('mouseenter', function() {

menu.style.display = 'flex';

menu.style.opacity = 1;

});

button.addEventListener('mouseleave', function() {

setTimeout(() => {

if (!menu.matches(':hover')) {

menu.style.opacity = 0;

setTimeout(() => {

menu.style.display = 'none';

}, 300);

}

}, 200);

});

menu.addEventListener('mouseleave', function() {

menu.style.opacity = 0;

setTimeout(() => {

menu.style.display = 'none';

}, 300);

});

menu.querySelectorAll('button').forEach(emojiButton => {

emojiButton.addEventListener('click', function() {

const reaction = emojiButton.textContent.trim();

const employeeId = button.closest('.employee-card').querySelector('.comment-btn').id.split('-')[2];

if (!reactionAlertShown) { // Check if the alert has already been shown

selectReaction(reaction, employeeId);

reactionAlertShown = true; // Set the flag to true after showing the alert

}

menu.style.opacity = 0;

setTimeout(() => {

menu.style.display = 'none';

}, 300);

});

});

});

});

function selectReaction(reaction, employeeId) {

fetch('../supervisor/save\_reaction.php', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({

employee\_id: employeeId,

reaction: reaction

})

})

.then(response => response.json())

.then(data => {

if (data.status === 'success') {

alert('Reaction saved: ' + reaction);

updateReactionCount(employeeId);

// Send a notification after saving the reaction

sendNotification(employeeId, reaction);

}

})

.catch(error => console.error('Error:', error));

}

function sendNotification(employeeId, reaction) {

fetch('../../db/fetchnotif.php', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({

employee\_id: employeeId,

reaction: reaction

})

})

.then(response => response.json())

.then(data => {

if (data.success) {

console.log('Notification sent successfully');

// Add the new notification to the notification dropdown

addNotificationToDropdown(data.notification);

} else {

console.error('Failed to send notification:', data.error);

}

})

.catch(error => console.error('Error:', error));

}

function addNotificationToDropdown(notification) {

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

// Create a new notification item

const listItem = document.createElement('li');

listItem.innerHTML = `

<a class="dropdown-item" href="#">

<div class="d-flex justify-content-between">

<span>${notification.message}</span>

<small class="text-muted">${new Date(notification.created\_at).toLocaleString()}</small>

</div>

<div class="d-flex justify-content-end mt-2">

<button class="btn btn-sm btn-danger delete-notification" data-id="${notification.notification\_id}">

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

</button>

</div>

</a>`;

// Add the new notification at the top of the list

notificationList.insertBefore(listItem, notificationList.firstChild);

// Update the notification count

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

const currentCount = parseInt(notificationCount.textContent, 10);

notificationCount.textContent = currentCount + 1;

}

// Function to open the comment modal with animation

function openCommentModal(employeeId) {

const modal = document.getElementById('commentModal');

modal.style.display = 'flex';

modal.classList.add('show');

// Add employee ID to the modal for reference

modal.setAttribute('data-employee-id', employeeId);

// Fetch and display existing comments and update the comment count

fetchComments(employeeId);

}

// Function to close the comment modal

function closeModal(modalId) {

const modal = document.getElementById(modalId);

modal.classList.remove('show');

setTimeout(() => {

modal.style.display = 'none';

}, 300);

}

// Function to post a comment

function postComment(event, input) {

if ((event.key === 'Enter' || event.type === 'click') && input.value.trim() !== '') {

const employeeId = document.getElementById('commentModal').getAttribute('data-employee-id');

const comment = input.value.trim();

fetch('../supervisor/save\_comments.php', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({

employee\_id: employeeId,

comment: comment

})

})

.then(response => response.json())

.then(data => {

if (data.status === 'success') {

// Add the comment to the UI

const commentList = document.querySelector('.modal-body .comment-list');

// Remove empty state if it exists

const emptyState = commentList.querySelector('.empty-comments');

if (emptyState) {

emptyState.remove();

}

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

commentElement.classList.add('comment');

commentElement.innerHTML = `

<div class="comment-header">

<span class="comment-author">${data.username || 'You'}</span>

<span class="comment-time" data-created-at="${data.created\_at}">just now</span>

</div>

<div class="comment-content">${comment}</div>

<div class="comment-actions">

<span class="comment-action"><i class="bi bi-hand-thumbs-up"></i> Like</span>

<span class="comment-action"><i class="bi bi-reply"></i> Reply</span>

</div>

`;

// Add new comment at the top

commentList.insertBefore(commentElement, commentList.firstChild);

input.value = ''; // Clear the input field

// Refresh the comment count

fetchComments(employeeId);

} else {

alert('Failed to save comment');

}

})

.catch(error => {

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

alert('Failed to save comment');

});

}

}

// Function to calculate time ago

function getTimeAgo(dateString) {

const now = new Date();

const past = new Date(dateString); // Parse the ISO 8601 date string

const seconds = Math.floor((now - past) / 1000);

let interval = Math.floor(seconds / 31536000);

if (interval >= 1) return interval === 1 ? '1 year ago' : interval + ' years ago';

interval = Math.floor(seconds / 2592000);

if (interval >= 1) return interval === 1 ? '1 month ago' : interval + ' months ago';

interval = Math.floor(seconds / 86400);

if (interval >= 1) return interval === 1 ? '1 day ago' : interval + ' days ago';

interval = Math.floor(seconds / 3600);

if (interval >= 1) return interval === 1 ? '1 hour ago' : interval + ' hours ago';

interval = Math.floor(seconds / 60);

if (interval >= 1) return interval === 1 ? '1 minute ago' : interval + ' minutes ago';

return seconds < 10 ? 'just now' : seconds + ' seconds ago';

}

// Function to update time ago for all comments

function updateTimeAgo() {

const commentElements = document.querySelectorAll('.comment-time');

commentElements.forEach(element => {

const dateString = element.getAttribute('data-created-at');

element.textContent = getTimeAgo(dateString);

});

}

// Function to fetch and display comments with time ago format

function fetchComments(employeeId) {

fetch('fetch\_comments.php', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({ employee\_id: employeeId })

})

.then(response => response.json())

.then(data => {

const commentList = document.querySelector('.modal-body .comment-list');

commentList.innerHTML = ''; // Clear existing comments

if (data.comments && data.comments.length > 0) {

data.comments.forEach(comment => {

// Create comment element

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

commentElement.classList.add('comment');

commentElement.innerHTML = `

<div class="comment-header">

<span class="comment-author">${comment.username || 'Anonymous'}</span>

<span class="comment-time" data-created-at="${comment.created\_at}">${getTimeAgo(comment.created\_at)}</span>

</div>

<div class="comment-content">${comment.comment}</div>

<div class="comment-actions">

<span class="comment-action"><i class="bi bi-hand-thumbs-up"></i> Like</span>

<span class="comment-action"><i class="bi bi-reply"></i> Reply</span>

</div>

`;

commentList.appendChild(commentElement);

});

// Update the comment count

const commentCountElement = document.getElementById(`comment-count-${employeeId}`);

if (commentCountElement) {

commentCountElement.textContent = data.total\_comments;

}

} else {

// Show empty state

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

emptyState.classList.add('empty-comments');

emptyState.innerHTML = `

<i class="bi bi-chat-square-text"></i>

<p>No comments yet. Be the first to comment!</p>

`;

commentList.appendChild(emptyState);

// Update the comment count to 0

const commentCountElement = document.getElementById(`comment-count-${employeeId}`);

if (commentCountElement) {

commentCountElement.textContent = 0;

}

}

// Start updating time ago every 1 second

setInterval(updateTimeAgo, 1000);

})

.catch(error => console.error('Error:', error));

}

function showNextEmployee(categoryIndex) {

const totalEmployees = document.querySelectorAll(`#category-${categoryIndex} .employee-card`).length;

if (!currentEmployeeIndex[categoryIndex]) {

currentEmployeeIndex[categoryIndex] = 1;

}

document.getElementById(`employee-${categoryIndex}-${currentEmployeeIndex[categoryIndex]}`).style.display = 'none';

currentEmployeeIndex[categoryIndex] = (currentEmployeeIndex[categoryIndex] % totalEmployees) + 1;

document.getElementById(`employee-${categoryIndex}-${currentEmployeeIndex[categoryIndex]}`).style.display = 'block';

}

function showPreviousEmployee(categoryIndex) {

const totalEmployees = document.querySelectorAll(`#category-${categoryIndex} .employee-card`).length;

if (!currentEmployeeIndex[categoryIndex]) {

currentEmployeeIndex[categoryIndex] = 1;

}

document.getElementById(`employee-${categoryIndex}-${currentEmployeeIndex[categoryIndex]}`).style.display = 'none';

currentEmployeeIndex[categoryIndex] = (currentEmployeeIndex[categoryIndex] - 1) || totalEmployees;

document.getElementById(`employee-${categoryIndex}-${currentEmployeeIndex[categoryIndex]}`).style.display = 'block';

}

window.onload = function() {

// Show the first category and first employee immediately

document.getElementById(`category-1`).style.display = 'block';

document.getElementById(`employee-1-1`).style.display = 'block';

// Auto display all employees

document.getElementById(`employee-all-1`).style.display = 'block';

};

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

const commentList = document.querySelector('.comment-list');

commentList.addEventListener('mouseenter', function () {

commentList.classList.add('scrolling');

});

commentList.addEventListener('mouseleave', function () {

commentList.classList.remove('scrolling');

});

commentList.addEventListener('scroll', function () {

if (commentList.scrollTop + commentList.clientHeight >= commentList.scrollHeight) {

// Load more comments if needed

console.log('Reached the bottom of the comment list');

}

});

});

// Function to update the reaction count

function updateReactionCount(employeeId) {

fetch('../supervisor/fetch\_reaction\_count.php', {

method: 'POST',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify({ employee\_id: employeeId })

})

.then(response => response.json())

.then(data => {

if (data.status === 'success') {

// Update the reaction counts in the UI

const reactionCountElement = document.getElementById(`reaction-count-${employeeId}`);

if (reactionCountElement) {

let reactionText = '';

for (const [reaction, count] of Object.entries(data.counts)) {

reactionText += `${reaction}: ${count} `;

}

reactionCountElement.textContent = reactionText.trim();

}

} else {

console.error('Failed to fetch reaction counts');

}

})

.catch(error => console.error('Error:', error));

}

</script>

<!-- Add these lines to your HTML to include the new PHP files -->

<script src="fetch\_comments.php"></script>

<script src="fetch\_reaction\_count.php"></script>

</body>

</html>

<?php

// Close the database connection

$conn->close();

?>