**Internet Programming Task 2 Explanation**

**Poovarasan Sandhanam**

Overview:

The provided PHP scripts manage a Premier League team management system, allowing users to login, add, delete, and edit teams, and generate reports using Chart.js. They enforce user authentication and handle form submissions for team data. With MySQL database integration, they ensure data persistence. The HTML forms facilitate user interaction, while the CSS styles enhance the visual appeal. Chart.js integration enables dynamic visualization of team statistics through pie and bar charts, providing insightful reports. The code's clarity and modularity facilitate maintenance and scalability. Overall, these scripts create a robust platform for managing Premier League teams and analyzing their performance efficiently.

Index.php

<?php

session\_start(); // Start a new session or resume the existing session.

include 'db.php'; // Include the 'db.php' file which presumably contains database connection details.

// Fetch teams data from the database.

$sql = "SELECT \* FROM football\_team";

$teams\_result = mysqli\_query($conn, $sql); // Execute the SQL query and store the result in $teams\_result.

?>

<!DOCTYPE html>

<html>

<head>

<title>Team Management</title>

<style>

/\* CSS styles for the webpage \*/

</style>

</head>

<body>

<!-- Header section -->

<div class="header container">

<h1>Premier League</h1>

<!-- Action buttons: Add New Team and Logout -->

</div>

<!-- Main content section -->

<div class="container">

<!-- Table to display football team data -->

<table>

<thead>

<!-- Table header row -->

</thead>

<tbody>

<?php

// PHP code to dynamically generate table rows based on fetched data

?>

</tbody>

</table>

<!-- Button to generate report -->

</div>

<!-- Footer section -->

<div class="footer container">

<!-- Copyright information -->

</div>

<!-- JavaScript code for client-side interactions -->

<script>

// JavaScript functions for generating reports and logging out

</script>

</body>

</html>

Login.php

<?php

session\_start(); // Start a new session or resume the existing session.

include 'db.php'; // Include the 'db.php' file which presumably contains database connection details.

if ($\_SERVER["REQUEST\_METHOD"] == "POST") { // Check if the request method is POST.

$username = $\_POST['username']; // Retrieve the username from the POST data.

$password = $\_POST['password']; // Retrieve the password from the POST data.

$sql = "SELECT \* FROM users WHERE username='$username'"; // SQL query to retrieve user data based on the provided username.

$result = mysqli\_query($conn, $sql); // Execute the SQL query.

$user = mysqli\_fetch\_assoc($result); // Fetch the user data as an associative array.

if (password\_verify($password, $user['password'])) { // Verify the provided password with the hashed password stored in the database.

$\_SESSION['userid'] = $user['userid']; // Set the 'userid' session variable to the user's ID.

header("Location: index.php"); // Redirect the user to the index.php page upon successful login.

} else {

echo "Invalid username or password"; // Display an error message if the username or password is invalid.

}

}

?>

<!DOCTYPE html>

<html>

<head>

<title>Login</title>

<link rel="stylesheet" type="text/css" href="styles.css"> <!-- Link to an external CSS file for styling -->

</head>

<body>

<!-- Login form -->

<div class="container">

<h1>Login</h1>

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

<label for="username">Username:</label><br>

<input type="text" id="username" name="username"><br> <!-- Input field for username -->

<label for="password">Password:</label><br>

<input type="password" id="password" name="password"><br> <!-- Input field for password -->

<input type="submit" value="Login"> <!-- Submit button for the login form -->

</form>

<!-- Link to the registration page -->

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

</div>

</body>

</html>

Register.php

<?php

session\_start(); // Start a new session or resume the existing session.

include 'db.php'; // Include the 'db.php' file which presumably contains database connection details.

if ($\_SERVER["REQUEST\_METHOD"] == "POST") { // Check if the request method is POST.

$username = $\_POST['username']; // Retrieve the username from the POST data.

$password = $\_POST['password']; // Retrieve the password from the POST data.

// Validate input: Check if username or password is empty

if (empty($username) || empty($password)) {

echo "Username and password cannot be empty.";

exit; // Exit the script if validation fails.

}

// Hash the password

$hashed\_password = password\_hash($password, PASSWORD\_DEFAULT); // Hash the password using the default algorithm.

// Use prepared statements to insert the data

$sql = "INSERT INTO users (username, password) VALUES (?, ?)"; // SQL query to insert new user data.

$stmt = $conn->prepare($sql); // Prepare the SQL statement.

if ($stmt) {

$stmt->bind\_param("ss", $username, $hashed\_password); // Bind parameters to the prepared statement.

if ($stmt->execute()) { // Execute the prepared statement.

echo "New user registered successfully"; // Display success message if registration is successful.

} else {

echo "Error: " . $stmt->error; // Display error message if execution fails.

}

$stmt->close(); // Close the prepared statement.

} else {

echo "Error: " . $conn->error; // Display error message if preparing the statement fails.

}

$conn->close(); // Close the database connection.

}

?>

<!DOCTYPE html>

<html>

<head>

<title>Register</title>

<link rel="stylesheet" type="text/css" href="styles.css"> <!-- Link to an external CSS file for styling -->

</head>

<body>

<!-- Registration form -->

<div class="container">

<h1>Register</h1>

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

<label for="username">Username:</label><br>

<input type="text" id="username" name="username" required><br> <!-- Input field for username -->

<label for="password">Password:</label><br>

<input type="password" id="password" name="password" required><br> <!-- Input field for password -->

<input type="submit" value="Register"> <!-- Submit button for the registration form -->

</form>

<!-- Link to the login page -->

<p>Already registered? <a class="login-link" href="login.php">Login here</a></p>

</div>

</body>

</html>

edit.php

<?php

session\_start(); // Start a new session or resume the existing session.

include 'db.php'; // Include the 'db.php' file which presumably contains database connection details.

if (!isset($\_SESSION['userid'])) { // Check if the user is logged in by verifying if the 'userid' session variable is set.

header("Location: login.php"); // Redirect the user to the login page if not logged in.

exit(); // Stop further execution of the script.

}

if (!isset($\_GET['id']) || !is\_numeric($\_GET['id'])) { // Check if the 'id' parameter is set in the URL and is a valid number.

// Handle the case where id is not set or not valid

// You can redirect the user or show an error message

exit('Invalid team ID'); // Exit the script and display an error message.

}

$id = $\_GET['id']; // Retrieve the 'id' parameter from the URL.

$sql = "SELECT \* FROM football\_team WHERE id='$id'"; // SQL query to select team information based on the provided ID.

$result = mysqli\_query($conn, $sql); // Execute the SQL query.

// Check if the query was successful and if a team was found

if (!$result || mysqli\_num\_rows($result) == 0) {

// Handle the case where no team was found

// You can redirect the user or show an error message

exit('Team not found'); // Exit the script and display an error message if no team was found.

}

$team = mysqli\_fetch\_assoc($result); // Fetch the team information as an associative array.

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Process form data

}

?>

<!DOCTYPE html>

<html>

<head>

<title>Edit Team</title>

<style>

/\* CSS styles for the webpage \*/

</style>

</head>

<body>

<!-- Edit Team form -->

<div class="container">

<h1>Edit Team</h1>

<?php if(isset($error\_message)): ?>

<div class="error"><?php echo $error\_message; ?></div> <!-- Display error message if exists -->

<?php endif; ?>

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

<!-- Form fields to edit team information -->

</form>

</div>

</body>

</html>

Delete.php

<?php

session\_start(); // Start a new session or resume the existing session.

include 'db.php'; // Include the 'db.php' file which presumably contains database connection details.

if (!isset($\_SESSION['userid'])) { // Check if the user is logged in by verifying if the 'userid' session variable is set.

header("Location: login.php"); // Redirect the user to the login page if not logged in.

exit(); // Stop further execution of the script.

}

$id = $\_GET['id']; // Retrieve the 'id' parameter from the URL.

if ($stmt = $conn->prepare("DELETE FROM football\_team WHERE id = ?")) { // Prepare a SQL statement to delete the team record.

$stmt->bind\_param("i", $id); // Bind the 'id' parameter to the prepared statement.

if ($stmt->execute()) { // Execute the prepared statement to delete the record.

header("Location: index.php"); // Redirect the user to the index.php page after successful deletion.

} else {

echo "Error deleting record: " . $stmt->error; // Display an error message if deletion fails.

}

$stmt->close(); // Close the prepared statement.

} else {

echo "Error preparing statement: " . $conn->error; // Display an error message if preparing the statement fails.

}

$conn->close(); // Close the database connection.

?>

Charts.php

<?php

session\_start(); // Start a new session or resume the existing session.

include 'db.php'; // Include the 'db.php' file which presumably contains database connection details.

if (!isset($\_SESSION['userid'])) { // Check if the user is logged in by verifying if the 'userid' session variable is set.

header("Location: login.php"); // Redirect the user to the login page if not logged in.

exit(); // Stop further execution of the script.

}

$ids = explode(',', $\_GET['ids']); // Split the 'ids' parameter from the URL into an array.

$teams\_sql = "SELECT \* FROM football\_team WHERE id IN (" . implode(',', $ids) . ")"; // Construct SQL query to select teams based on the provided IDs.

$teams\_result = mysqli\_query($conn, $teams\_sql); // Execute the SQL query to fetch the teams data.

$teams\_data = []; // Initialize an empty array to store teams data.

while ($row = mysqli\_fetch\_assoc($teams\_result)) { // Loop through the query result and store each row (team data) in the array.

$teams\_data[] = $row;

}

$colors = ['#3498DB', '#9B59B6', '#E74C3C', '#F39C12']; // Define an array of colors for chart visualization.

$colorIndex = 0; // Initialize a color index counter.

?>

Add.php

<?php

// Start or resume a session

if (session\_status() == PHP\_SESSION\_NONE) {

session\_start();

}

// Include the database connection file

include 'db.php';

// Redirect to login page if the user is not authenticated

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

header("Location: login.php");

exit();

}

// Handle form submission

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Retrieve form data

$name = $\_POST['name'];

$played = $\_POST['played'];

$won = $\_POST['won'];

$drawn = $\_POST['drawn'];

$lost = $\_POST['lost'];

$goals\_for = $\_POST['goals\_for'];

$goals\_against = $\_POST['goals\_against'];

$goal\_difference = $\_POST['goal\_difference'];

$points = $\_POST['points'];

$form = $\_POST['form'];

// Construct SQL query to insert data into the 'teams' table

$sql = "INSERT INTO teams (name, played, won, drawn, lost, goals\_for, goals\_against, goal\_difference, points, form) VALUES ('$name', '$played', '$won', '$drawn', '$lost', '$goals\_for', '$goals\_against', '$goal\_difference', '$points', '$form')";

// Execute the SQL query

if (mysqli\_query($conn, $sql)) {

echo "Team added successfully";

} else {

echo "Error: " . $sql . "<br>" . mysqli\_error($conn);

}

}

?>