

ASSIGNMENT – 4

REG_NO: 220701057

index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Login</title>
  <link rel="stylesheet" href="css/style.css">
</head>
<body>

  <div class="container" id="game-container">
    <h1>Welcome, <span id="username-display"></span>!</h1>
    <div id="score-display">Score: <span id="score">0</span></div>
    <div id="game-area"></div>
    <button type="button" id="start-game-btn">Start Game</button>
    <button type="button" onclick="switchtab()">Manage Game</button>
    <button type="button" id="stop-game-btn">End Game</button>
  </div>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script>
    $(document).ready(function () {
      $('#login-form').on('submit', function (event) {
        event.preventDefault(); // Prevent the default form submission

        $.ajax({
          url: 'http://localhost:3000/php/login.php', // Ensure this is the correct path
          type: 'POST', // Make sure this is set to POST
          data: $(this).serialize(), // Serialize form data
          success: function (response) {
            console.log('Login successful:', response);
            window.location.href= "game.html";
            // Handle successful login here
          },
          error: function (xhr, status, error) {
            console.error('Error during login:', error);
          }
        });
      });
    });
  </script>
</body>

</html>
```

Style.css:

```
/* Global Styles */
body {
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
  margin: 0;
  padding: 0;
  display: flex;
  justify-content: center;
  align-items: center;
```

```
height: 100vh;
background-color: #eef2f3; /* Soft pastel blue-gray */
color: #333;
}
/* Container Styles */
.container {
width: 100%;
max-width: 400px;
padding: 20px;
background-color: #ffffff;
border-radius: 10px;
box-shadow: 0 4px 15px rgba(0, 0, 0, 0.1);
text-align: center;
position: relative;
}
h1, h2, h3 {
margin: 0 0 15px;
color: #4a4a4a;
}
/* Buttons */
button {
padding: 10px 15px;
margin: 10px 0;
background-color: #007bff;
color: white;
border: none;
border-radius: 5px;
cursor: pointer;
font-size: 16px;
width: 100%;
}
button:hover {
background-color: #0056b3;
}
/* Forms */
form {
margin: 20px 0;
display: flex;
flex-direction: column;
}
input, select {
margin: 10px 0;
padding: 10px;
width: 100%;
border: 1px solid #ddd;
border-radius: 5px;
font-size: 14px;
}

/* Table Styles */
```

```

table {
  width: 100%;
  border-collapse: collapse;
  margin-top: 20px;
}
th, td {
  border: 1px solid #ddd;
  padding: 10px;
  text-align: center;
}
th {
  background-color: #f8f9fa; /* Light gray for headers */
  font-weight: bold;
}

```

game.js:

```

$(document).ready(function() {
  let score = 0;
  let spawnInterval; // For storing the spawn interval ID
  let spawnSpeed = 2000; // Default spawn speed in milliseconds (adjustable based on admin settings)
  // Fetch username for display
  $('#username-display').text(localStorage.getItem('username'));
  // Function to start the object spawning
  function startSpawning() {
    spawnInterval = setInterval(spawnObject, spawnSpeed); // Start spawning at defined intervals
  }
  // Spawn object function
  function spawnObject() {
    // Remove any existing objects
    $('.object').remove(); // Remove previous object if any
    // Create new object
    const object = $('<div class="object"></div>').css({
      position: 'absolute',
      width: '30px',
      height: '30px',
      backgroundColor: getRandomColor(),
      top: Math.random() * (370 - 30) + 'px', // Ensure it spawns within the visible area
      left: Math.random() * (270 - 30) + 'px' // Ensure it spawns within the visible area
    });
    // Add click event for the new object
    object.click(function() {
      $(this).fadeOut(300, function() {
        $(this).remove();
        score++;
        $('#score').text(score);
        updateScoreOnServer(score); // Update score in the database
      });
    });
  }

  // Append the new object to the game area

```

```

$('#game-area').append(object);
}
function updateScoreOnServer(newScore) {
    const username = localStorage.getItem('username'); // Retrieve the username from local storage

    $.ajax({
        url: 'php/update_score.php',
        type: 'POST',
        data: { username: username, score: newScore }, // Include the username in the request
        success: function(response) {
            const result = JSON.parse(response);
            console.log('Score update response:', result);
            if (result.status === 'error') {
                console.error('Error updating score:', result.message);
            }
        },
        error: function(xhr, status, error) {
            console.error('AJAX error:', error);
        }
    });
}
// Function to get a random color
function getRandomColor() {
    const letters = '0123456789ABCDEF';
    let color = '#';
    for (let i = 0; i < 6; i++) {
        color += letters[Math.floor(Math.random() * 16)];
    }
    return color;
}
// Start the game by spawning the first object
$('#start-game-btn').click(function() {
    startSpawning(); // Call function to stop spawning
}); // Start spawning objects

// Function to stop the spawning of objects
function stopSpawning() {
    clearInterval(spawnInterval); // Stop the spawning interval
    alert('Game Over! Your final score is ${score}.'); // Alert final score
    $('#game-area').empty(); // Clear the game area
    $('#score').text(0); // Show the final score
}
// Event listener for the Stop Game button
$('#stop-game-btn').click(function() {
    stopSpawning(); // Call function to stop spawning
});

// Example function to adjust game speed (to be called based on admin settings)
function setGameSpeed(newSpeed) {
    clearInterval(spawnInterval); // Clear the current interval

```

```

    spawnSpeed = newSpeed; // Update spawn speed
    startSpawning(); // Restart spawning with the new speed
}
// Simulate changing speed for testing (can be removed or replaced with actual admin logic)
setTimeout(() => {
    setGameSpeed(1000); // Example: Change speed to every 1 second after 10 seconds
}, 10000);
});

```

Login.php:

```

<?php
include 'db.php';
header("Access-Control-Allow-Origin: *"); // Allow all origins or specify your domain
header("Access-Control-Allow-Methods: POST, GET, OPTIONS"); // Allow specific methods
header("Access-Control-Allow-Headers: Content-Type"); // Allow specific headers

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Retrieve username and password
    $username = $_POST['username'];
    $password = $_POST['password'];

    // Prepare SQL statement
    $stmt = $pdo->prepare("SELECT * FROM players WHERE username = ? AND password = ?");
    $stmt->execute([$username, $password]);
    $user = $stmt->fetch();

    if ($user) {
        session_start();
        $_SESSION['user_id'] = $user['player_id']; // Use 'player_id' instead of 'id'
        $_SESSION['username'] = $user['username'];
        echo json_encode(['status' => 'success', 'redirect' => 'game.html']);
        exit();
    } else {
        echo json_encode(['status' => 'error', 'message' => 'Invalid username or password.']);
    }
} else {
    http_response_code(405); // Method Not Allowed
    echo json_encode(['status' => 'error', 'message' => 'Method Not Allowed']);
}
?>

```

IMPLEMENTATION SCREENSHOTS:

