Project Report: Tambola Game Application

1. Project Design:

Overview: The Tambola Game Application is a web-based platform designed to facilitate the popular game of Tambola (also known as Bingo or Housie). The application allows users to create new game sessions, join existing sessions, generate and view Tambola tickets, and participate in the game by marking numbers on their tickets.

Components:

1. **User Interface:**

- The application provides a user-friendly interface designed with HTML, CSS, and JavaScript.
- Users can start a new game, join an existing game, view their tickets, and mark numbers during gameplay.

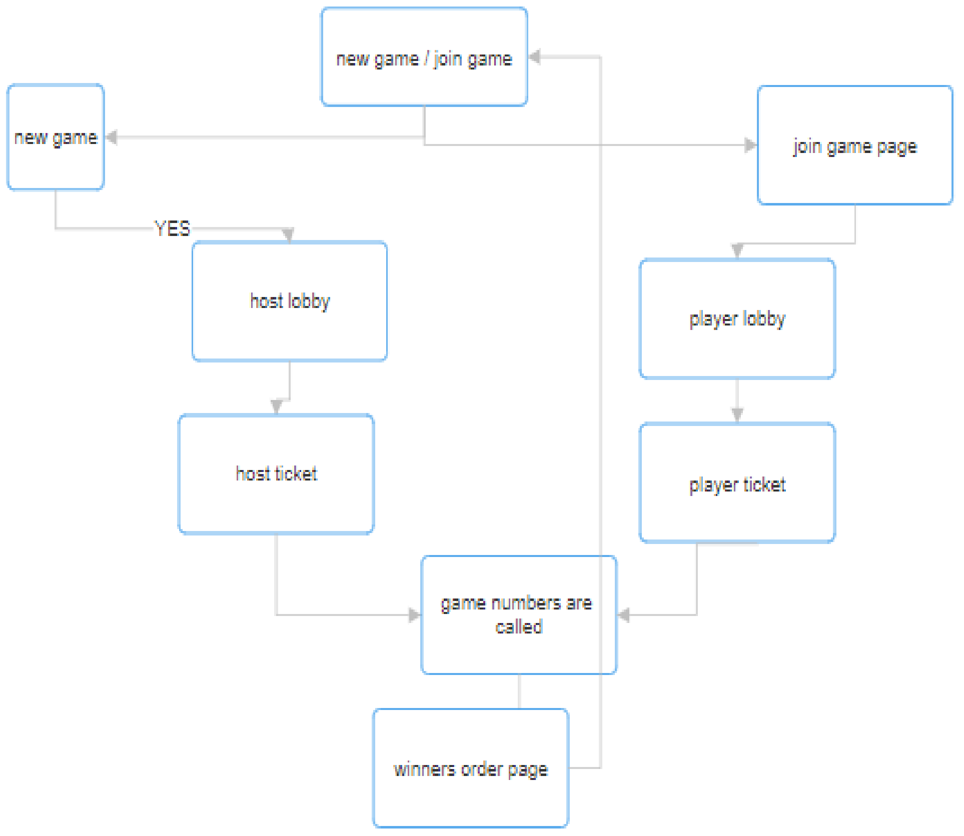
2. **Server-Side Logic:**

- Backend functionalities are implemented using PHP scripts.
- These scripts handle database operations, user authentication, game session management, and ticket generation.
- We have set up a server using xampp backend technology.

3. **Database Management:**

- MySQL database is utilized to store game sessions, player information, called numbers, and tickets.
- The database schema includes tables for GameSessions, Players, CalledNumbers, and Tickets.
- Queries: Optimize database queries for efficient data retrieval and storage.

2. FLOWCHART:



3. Functionality:

Starting a New Game:

- Host users can initiate a new game session by accessing the "New Game" option.
- Upon starting a new game, a unique invitation code is generated, and the host is redirected to the lobby page.
- Other players can join the game using this invitation code.

Joining a Game:

- Players can join an existing game session by entering the invitation code provided by the host.
- The application verifies the validity of the invitation code and allows the player to enter their name.
- Once joined, players are redirected to the lobby page where they can view other players and wait for the host to start the game.

Game Lobby:

- The lobby page displays the list of players who have joined the game.
- Host users have the option to start the game once all players are ready.

- Players can view their tickets and participate in the game.

Ticket Generation:

- Upon joining the game, each player receives a unique Tambola ticket.
- Tickets are generated randomly, ensuring each player has a distinct set of numbers.

Gameplay:

- During gameplay, called numbers are displayed to all players.
- Players can mark numbers on their tickets by clicking on the corresponding buttons.
- The application checks for winning conditions and declares winners accordingly.

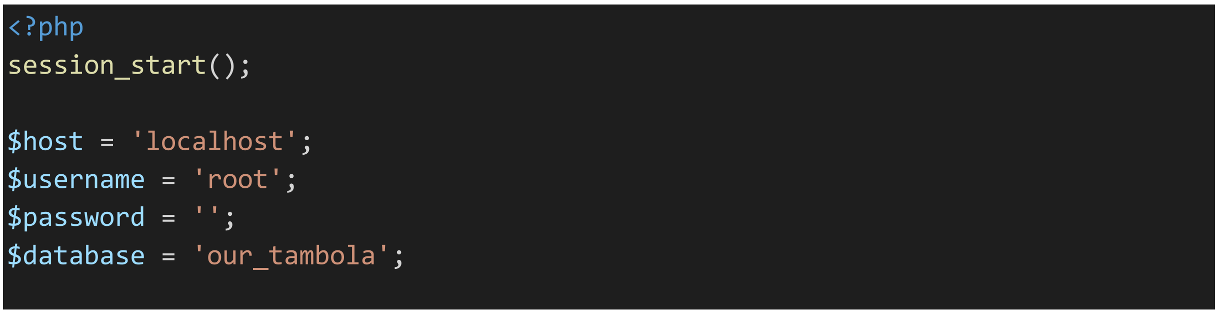
4. Limitations:

1. Scalability: The current design may face challenges in handling a large number of concurrent users or multiple game sessions simultaneously.
2. UI/UX: While functional, the user interface could be further enhanced for better aesthetics and usability.
3. Security: The application lacks robust security measures such as data encryption and input validation, making it vulnerable to potential security threats.
4. Cross-Browser Compatibility: The application may not be fully optimized for all web browsers, potentially leading to inconsistencies in user experience.
5. Error Handling: Error handling mechanisms could be improved to provide more informative and user-friendly error messages.
6. Refresh: when a ticket page is refreshed its ticket position will change.

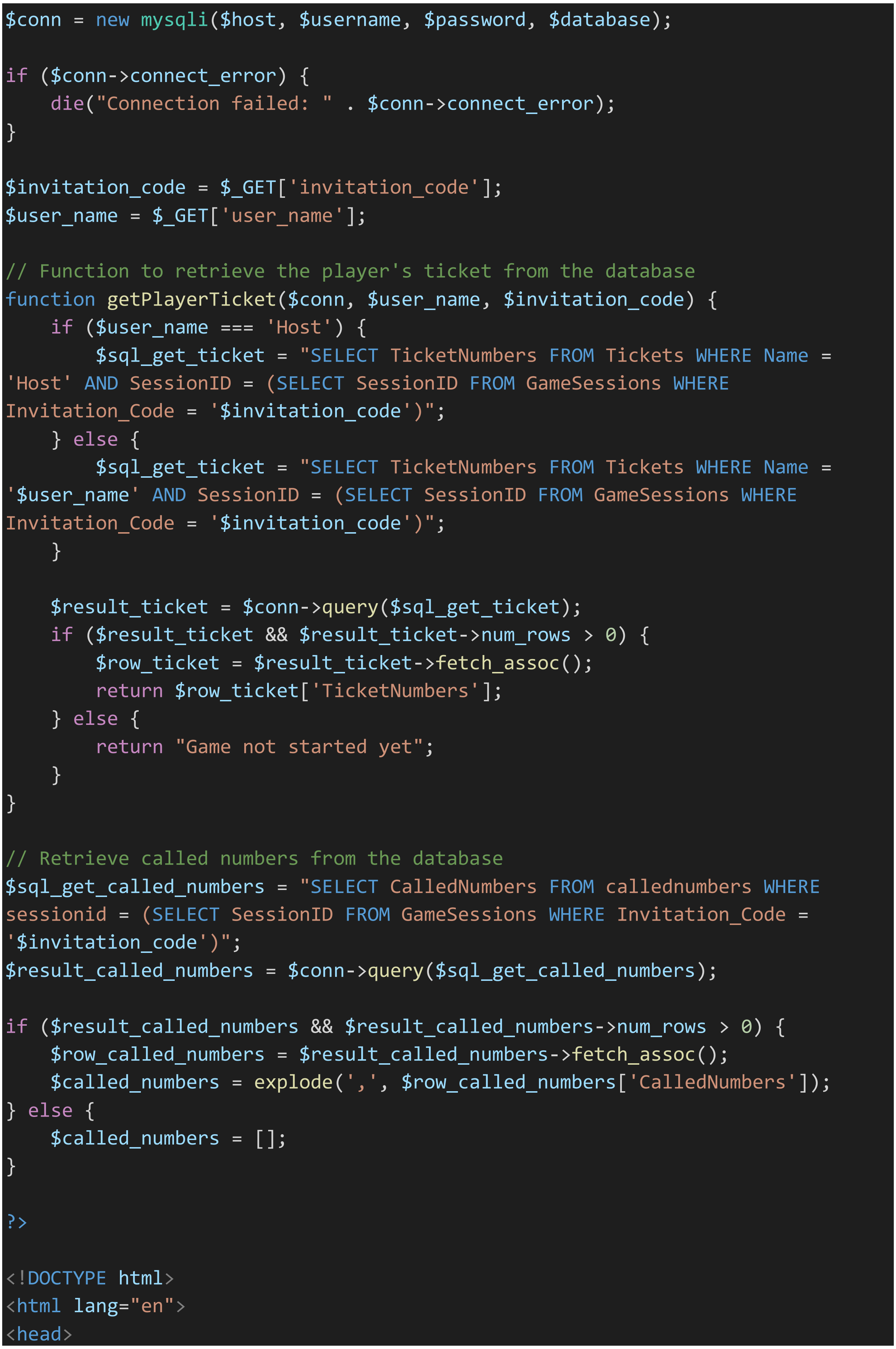
5. DIFFICULTIES THAT WE HAVE FACED:

7. Creating lobby dynamically for every player.
8. Displaying player ticket in an grid format like an tambola ticket.
9. Calling same number for every player.
10. Putting colour to the pressed button if they are called.

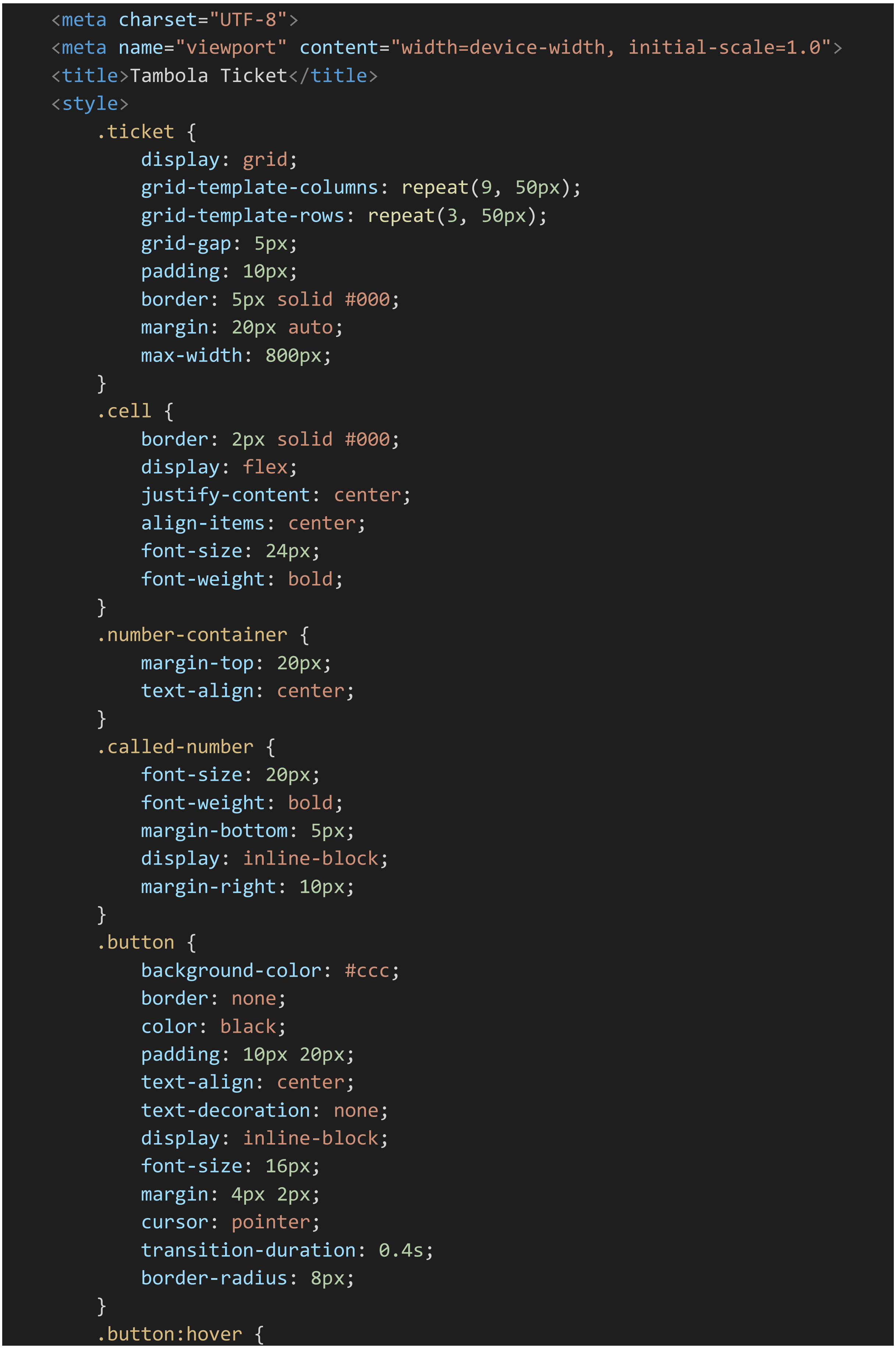
6. IMPORTANT PIECE OF CODE : VIEW_TICKETS.PHP



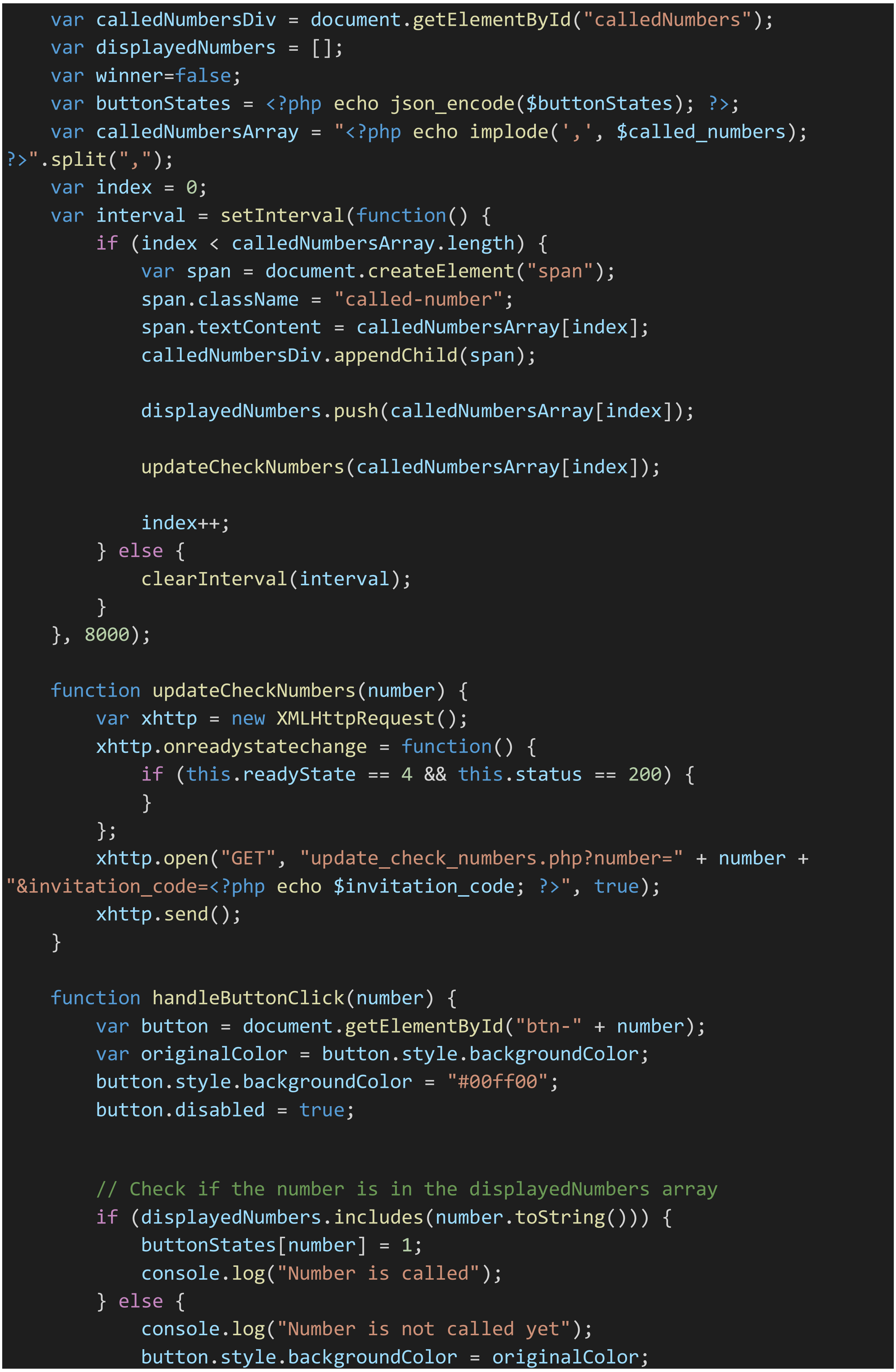
```php
<?php
session_start();

$host = 'localhost';
$username = 'root';
$password = '';
$database = 'our_tambola';
```



```
$conn = new mysqli($host, $username, $password, $database);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$invitation_code = $_GET['invitation_code'];
$user_name = $_GET['user_name'];

// Function to retrieve the player's ticket from the database
function getPlayerTicket($conn, $user_name, $invitation_code) {
    if ($user_name === 'Host') {
        $sql_get_ticket = "SELECT TicketNumbers FROM Tickets WHERE Name =
'Host' AND SessionID = (SELECT SessionID FROM GameSessions WHERE
Invitation_Code = '$invitation_code')";
    } else {
        $sql_get_ticket = "SELECT TicketNumbers FROM Tickets WHERE Name =
'$user_name' AND SessionID = (SELECT SessionID FROM GameSessions WHERE
Invitation_Code = '$invitation_code')";
    }

    $result_ticket = $conn->query($sql_get_ticket);
    if ($result_ticket && $result_ticket->num_rows > 0) {
        $row_ticket = $result_ticket->fetch_assoc();
        return $row_ticket['TicketNumbers'];
    } else {
        return "Game not started yet";
    }
}

// Retrieve called numbers from the database
$sql_get_called_numbers = "SELECT CalledNumbers FROM callednumbers WHERE
sessionid = (SELECT SessionID FROM GameSessions WHERE Invitation_Code =
'$invitation_code')";
$result_called_numbers = $conn->query($sql_get_called_numbers);

if ($result_called_numbers && $result_called_numbers->num_rows > 0) {
    $row_called_numbers = $result_called_numbers->fetch_assoc();
    $called_numbers = explode(',', $row_called_numbers['CalledNumbers']);
} else {
    $called_numbers = [];
}

?>

<!DOCTYPE html>
<html lang="en">
<head>
```



```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Tambola Ticket</title>
    <style>
        .ticket {
            display: grid;
            grid-template-columns: repeat(9, 50px);
            grid-template-rows: repeat(3, 50px);
            grid-gap: 5px;
            padding: 10px;
            border: 5px solid #000;
            margin: 20px auto;
            max-width: 800px;
        }
        .cell {
            border: 2px solid #000;
            display: flex;
            justify-content: center;
            align-items: center;
            font-size: 24px;
            font-weight: bold;
        }
        .number-container {
            margin-top: 20px;
            text-align: center;
        }
        .called-number {
            font-size: 20px;
            font-weight: bold;
            margin-bottom: 5px;
            display: inline-block;
            margin-right: 10px;
        }
        .button {
            background-color: #ccc;
            border: none;
            color: black;
            padding: 10px 20px;
            text-align: center;
            text-decoration: none;
            display: inline-block;
            font-size: 16px;
            margin: 4px 2px;
            cursor: pointer;
            transition-duration: 0.4s;
            border-radius: 8px;
        }
        .button:hover {
            background-color: #555;
            color: white;
        }
        body {
            background-image: url('https://mir-s3-cdn-cf.behance.net/projects/404/5d9b6e174828525.Y3JvcCwxMDQxLDgxNCwwLDE4Mg.gif'); /* Replace 'background.jpg' with the path to your background image */
            background-size: cover;
            background-position: center;
            font-family: Arial, sans-serif;
            color: #fff;
        }
    </style>
</head>
<body>

<div class="ticket">
    <?php
    $player_ticket = getPlayerTicket($conn, $user_name, $invitation_code);

    // Display the player's ticket
    $ticket_numbers = explode(',', $player_ticket);
    $index = range(1, 27);
    shuffle($index);
    $buttonStates = [];
    for ($col = 0; $col < 9; $col++) {
        for ($row = 0; $row < 3; $row++) {
            $rand_index = array_rand($index);
            $number = isset($ticket_numbers[$rand_index]) ? $ticket_numbers[$rand_index] : '';
            unset($index[$rand_index]);
            echo "<button class='cell' id='btn-$number' onclick='handleButtonClick($number)'>$number</button>";
            $buttonStates[$number] = 0;
        }
    }
    ?>
</div>

<div class="number-container">
    <h3>Called Numbers</h3>
    <div id="calledNumbers" style="display: flex; flex-wrap: wrap;"></div>
</div>

<button class="button" onclick="submitTicket('<?php echo $invitation_code; ?>', '<?php echo $user_name; ?>')">Submit Ticket</button>


<script>
```



```
    var calledNumbersDiv = document.getElementById("calledNumbers");
    var displayedNumbers = [];
    var winner=false;
    var buttonStates = <?php echo json_encode($buttonStates); ?>;
    var calledNumbersArray = "<?php echo implode(',', $called_numbers);
?>".split(",");
    var index = 0;
    var interval = setInterval(function() {
        if (index < calledNumbersArray.length) {
            var span = document.createElement("span");
            span.className = "called-number";
            span.textContent = calledNumbersArray[index];
            calledNumbersDiv.appendChild(span);

            displayedNumbers.push(calledNumbersArray[index]);

            updateCheckNumbers(calledNumbersArray[index]);

            index++;
        } else {
            clearInterval(interval);
        }
    }, 8000);

    function updateCheckNumbers(number) {
        var xhttp = new XMLHttpRequest();
        xhttp.onreadystatechange = function() {
            if (this.readyState == 4 && this.status == 200) {
            }
        };
        xhttp.open("GET", "update_check_numbers.php?number=" + number +
"&invitation_code=<?php echo $invitation_code; ?>", true);
        xhttp.send();
    }

    function handleButtonClick(number) {
        var button = document.getElementById("btn-" + number);
        var originalColor = button.style.backgroundColor;
        button.style.backgroundColor = "#00ff00";
        button.disabled = true;


        // Check if the number is in the displayedNumbers array
        if (displayedNumbers.includes(number.toString())) {
            buttonStates[number] = 1;
            console.log("Number is called");
        } else {
            console.log("Number is not called yet");
            button.style.backgroundColor = originalColor;
            button.disabled = false;
        }
    }

    function submitTicket(invitation_code, user_name) {
        var count = 0;
        var gameEnded = false;

        for (var number in buttonStates) {
            if (buttonStates[number] === 1) {
                count++;
            }
        }

        // Check if there are at least 5 clicked buttons
        if (count >= 1) {
            alert("Congratulations! Your game is completed!");
            gameEnded = true;
        } else {
            alert("Sorry, you need at least 5 clicked buttons to win.");
        }

        if (gameEnded) {
            window.location.href = 'index.php?invitation_code=<?php echo $invitation_code; ?>&user_name=<?php echo $user_name; ?>&gameEnded=<?php echo time(); ?>';
        }
    }


</script>

</body>
</html>

<?php
$conn->close();
?>
```

7. Future Project Extensions:

1. Multiplayer Support: Enhance the application to support multiplayer interactions, enabling users to compete against each other in real-time.
2. Customization Options: Introduce features that allow hosts to customize game settings, such as ticket layouts, number ranges, and winning patterns.

3. Mobile Compatibility: Develop mobile-responsive designs to ensure seamless access to the application across various devices.
4. Social Integration: Implement social media integration to facilitate sharing game invitations and results with friends and contacts.
5. Advanced Gameplay Features: Introduce advanced gameplay features such as power-ups, bonus rounds, and themed variations of the game to enhance user engagement.

Conclusion: The Tambola Game Application provides an entertaining platform for users to enjoy the classic game of Tambola with friends and family. While the current version offers basic functionality, there is ample room for improvement and expansion to deliver a more immersive and engaging gaming experience in the future.