HiLCoE School of Computer Science & Technology

Chapter Two: MYSQL DataBase

Course Title : Web Technologies II

Instructor name: Yitayew Solomon
E-mail address: yitayewsolomon3@gmail.com

Select Data From a MySQL Database

To select and display data from a MySQL database in PHP, you can use the following approach:

Steps:

1. **Connect to the MySQL database** using PHP.
2. **Write an SQL** `SELECT` **query** to retrieve data.
3. **Execute the query** and process the result.
4. **Display the data** on a webpage.

Cont. ...

C:\xampp\htdocs\web\Database\Select_Data.php - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

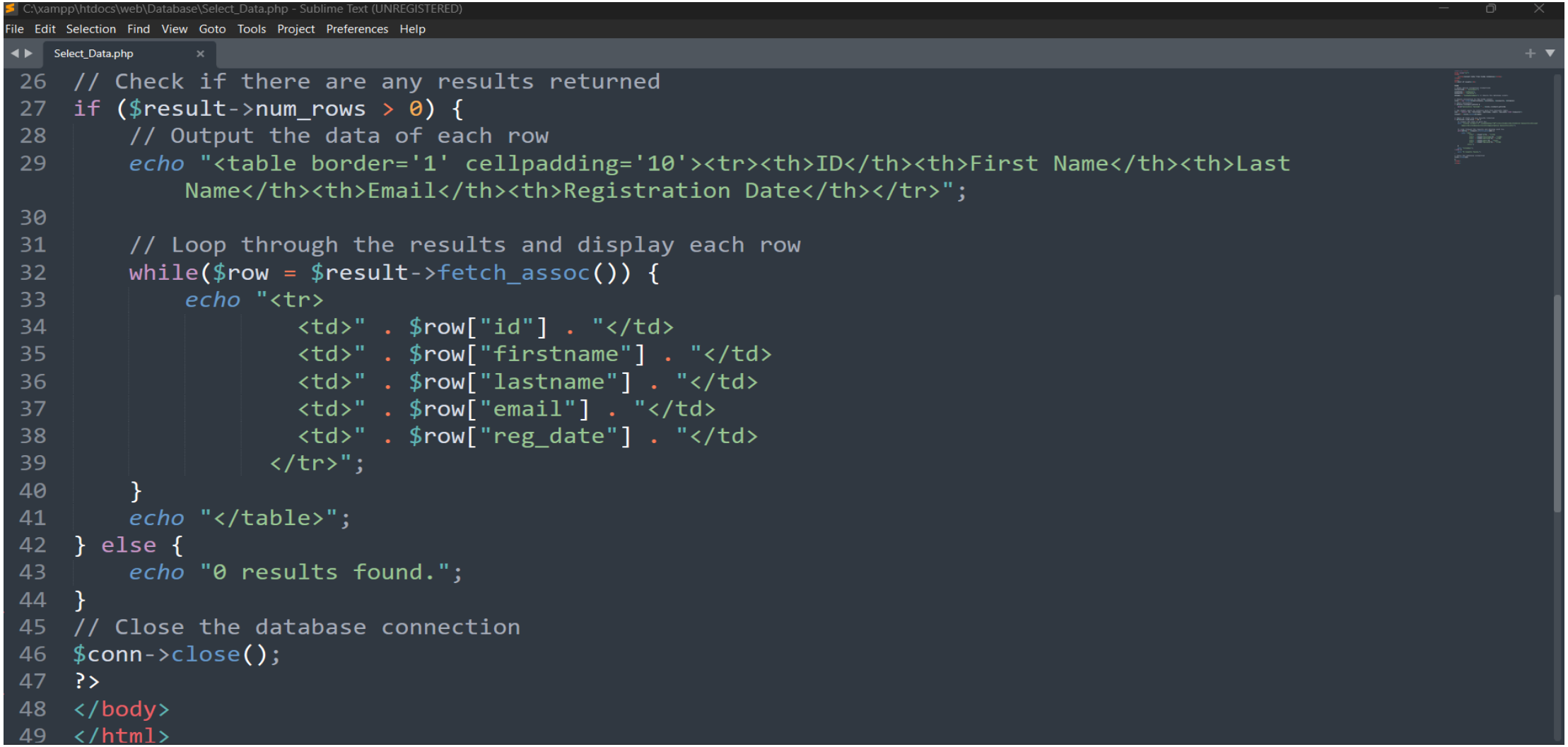
Select_Data.php

```php
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Select Data from MySQL Database</title>
</head>
<body>
<h2>List of Guests</h2>

<?php
// MySQL server connection credentials
$servername = "localhost";
$username = "yitayew";
$password = "test@123";
$dbname = "myNewDatabase"; // Ensure the database exists

// Create connection to the MySQL server
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
// SQL SELECT query to retrieve data from MyGuests table
$sql = "SELECT id, firstname, lastname, email, reg_date FROM MyGuests";
$result = $conn->query($sql);
```

Cont. ...


C:\xampp\htdocs\web\Database\Select_Data.php - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Select_Data.php
// Check if there are any results returned
if ($result->num_rows > 0) {
// Output the data of each row
echo "<table border='1' cellpadding='10'><tr><th>ID</th><th>First Name</th><th>Last Name</th><th>Email</th><th>Registration Date</th></tr>";
// Loop through the results and display each row
while($row = $result->fetch_assoc()) {
echo "<tr>
<td>" . $row["id"] . "</td>
<td>" . $row["firstname"] . "</td>
<td>" . $row["lastname"] . "</td>
<td>" . $row["email"] . "</td>
<td>" . $row["reg_date"] . "</td>
</tr>";
}
echo "</table>";
} else {
echo "0 results found.";
}
// Close the database connection
$conn->close();
?>
</body>
</html>

Cont. ...

List of Guests

ID	First Name	Last Name	Email	Registration Date
1	yitayew	Solomon	yitayewsolomon3@gmail.com	2024-10-07 11:04:24
2	Haileab	Solomon	haileabsolomon@gmail.com	2024-10-07 11:12:31
3	Yared	Solomon	yaredsolomon@gmail.com	2024-10-07 11:12:31
4	Natanim	Yitayew	natanimyitayew@gmail.com	2024-10-07 11:20:57
5	Dureti	Guye	duretiguye@gmail.com	2024-10-07 11:20:57

Cont. ...

Explanation of the Code:

1. **Database Connection:**
   - We connect to the MySQL server using the `mysqli` object and specify the server name (`localhost`), username (`root`), password (if any), and database name (`myNewDatabase`).
   - If the connection fails, the script will stop and display the error message.
2. **SQL Query:**
   - The SQL query `SELECT id, firstname, lastname, email, reg_date FROM MyGuests` retrieves the `id`, `firstname`, `lastname`, `email`, and `reg_date` fields from the `MyGuests` table.
3. **Executing the Query:**
   - The `query()` function is used to execute the `SELECT` query.
   - The result of the query is stored in the `$result` variable.

Cont. ...

4. **Processing the Results:**
   - We check if the query returned any rows using `$result->num_rows`.
   - If rows are returned, we loop through each row using `fetch_assoc()` to fetch the result as an associative array.
   - Each row of data is displayed in a table.
5. **Displaying Data:**
   - If data is found, it's displayed inside an HTML table, with columns for `ID`, `First Name`, `Last Name`, `Email`, and `Registration Date`.
   - If no data is found, the message `0 results found` is displayed.

Cont. ...

MySQL Table Structure

Make sure your `MyGuests` table in the `myNewDatabase` database has the following structure:

sql Copy code

```sql
CREATE TABLE MyGuests (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    firstname VARCHAR(30) NOT NULL,
    lastname VARCHAR(30) NOT NULL,
    email VARCHAR(50),
    reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);
```

Select Data Using Form

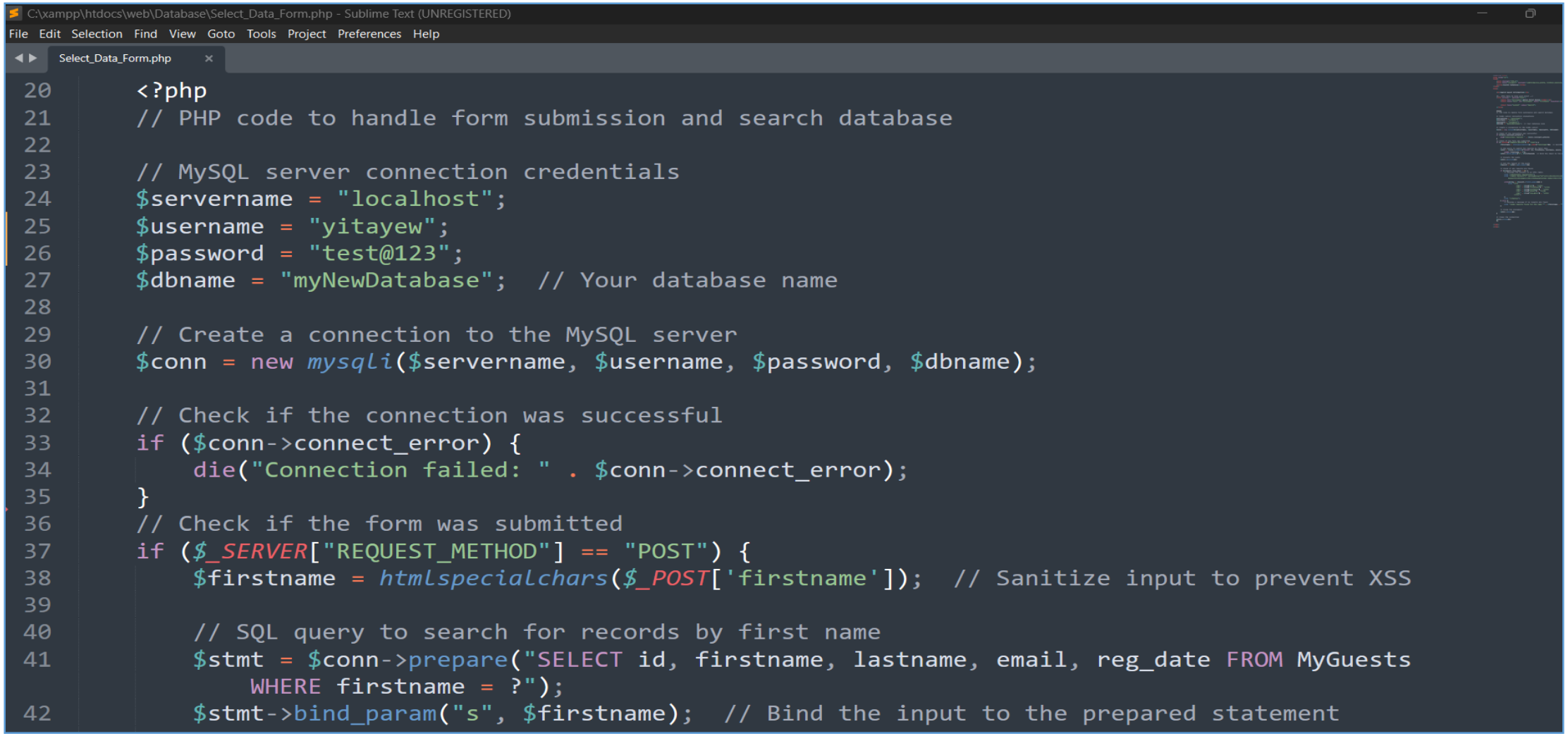
C:\xampp\htdocs\web\Database\Select_Data_Form.php - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

Select_Data_Form.php

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Search Database</title>
7  </head>
8  <body>
9
10     <h2>Search Guest Information</h2>
11
12     <!-- HTML Form to take user input -->
13     <form action="" method="POST">
14         <label for="firstname">Enter First Name:</label><br>
15         <input type="text" id="firstname" name="firstname" required><br><br>
16
17         <input type="submit" value="Search">
18     </form>
19
```

Cont. ...


C:\xampp\htdocs\web\Database\Select_Data_Form.php - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Select_Data_Form.php
20 <?php
21 // PHP code to handle form submission and search database
22
23 // MySQL server connection credentials
24 $servername = "localhost";
25 $username = "yitayew";
26 $password = "test@123";
27 $dbname = "myNewDatabase"; // Your database name
28
29 // Create a connection to the MySQL server
30 $conn = new mysqli($servername, $username, $password, $dbname);
31
32 // Check if the connection was successful
33 if ($conn->connect_error) {
34 die("Connection failed: " . $conn->connect_error);
35 }
36 // Check if the form was submitted
37 if ($_SERVER["REQUEST_METHOD"] == "POST") {
38 $firstname = htmlspecialchars($_POST['firstname']); // Sanitize input to prevent XSS
39
40 // SQL query to search for records by first name
41 $stmt = $conn->prepare("SELECT id, firstname, lastname, email, reg_date FROM MyGuests
WHERE firstname = ?");
42 $stmt->bind_param("s", $firstname); // Bind the input to the prepared statement

Cont. ...

C:\xampp\htdocs\web\Database\Select_Data_Form.php - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

Select_Data_Form.php

```php
        // Execute the query
        $stmt->execute();

        // Get the result of the query
        $result = $stmt->get_result();

        // Check if any results are found
        if ($result->num_rows > 0) {
            // Display the results in an HTML table
            echo "<h3>Results Found:</h3>";
            echo "<table border='1' cellpadding='10'><tr><th>ID</th><th>First Name</th><th>Last
                Name</th><th>Email</th><th>Registration Date</th></tr>";

            while($row = $result->fetch_assoc()) {
                echo "<tr>
                        <td>" . $row["id"] . "</td>
                        <td>" . $row["firstname"] . "</td>
                        <td>" . $row["lastname"] . "</td>
                        <td>" . $row["email"] . "</td>
                        <td>" . $row["reg_date"] . "</td>
                      </tr>";
            }
            echo "</table>";
        } else {
```

Cont. ...

C:\xampp\htdocs\web\Database\Select_Data_Form.php - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

Select_Data_Form.php

```
67                  // Display a message if no results are found
68                  echo "<p>No results found for the name '" . $firstname . "'</p>";
69              }
70
71              // Close the statement
72              $stmt->close();
73          }
74
75          // Close the connection
76          $conn->close();
77          ?>
78
79      </body>
80      </html>
```

Search Guest Information

Enter First Name:

yitayew

Search

Results Found:

ID	First Name	Last Name	Email	Registration Date
1	yitayew	Solomon	yitayewsolomon3@gmail.com	2024-10-07 11:04:24
6	yitayew	Solomon	yitayewsolomon3@gmail.com	2024-10-07 11:44:14

Cont. ...

Explanation:

1. **HTML Form:**
   - The form accepts a **first name** as input from the user.
   - When the user submits the form, the data is sent using the `POST` method to the same page.
2. **PHP Script:**
   - **Database Connection**: The connection to the MySQL database is established using `mysqli`.
   - **Form Handling**: Once the form is submitted, the PHP script checks if the request method is `POST`.
   - **Input Sanitization**: The input (`firstname`) is sanitized using `htmlspecialchars()` to prevent cross-site scripting (XSS).

Cont. ...

- **Prepared Statement**: The SQL query is written using a prepared statement (`$stmt = $conn->prepare()`), which helps prevent SQL injection attacks. We bind the input (`firstname`) to the query using `bind_param()` .
- **Query Execution**: The query is executed, and the results are fetched using `fetch_assoc()` .
- **Display Results**: If results are found, they are displayed in a table. If no results are found, an appropriate message is displayed.

3. **Table Display**:
   - The data retrieved from the database is displayed in an HTML table.
   - Columns displayed: `ID` , `First Name` , `Last Name` , `Email` , and `Registration Date` .

Cont. ...

Database Structure:

The `MyGuests` table should have the following structure in your MySQL database:

sql Copy code

```sql
CREATE TABLE MyGuests (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    firstname VARCHAR(30) NOT NULL,
    lastname VARCHAR(30) NOT NULL,
    email VARCHAR(50),
    reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);
```

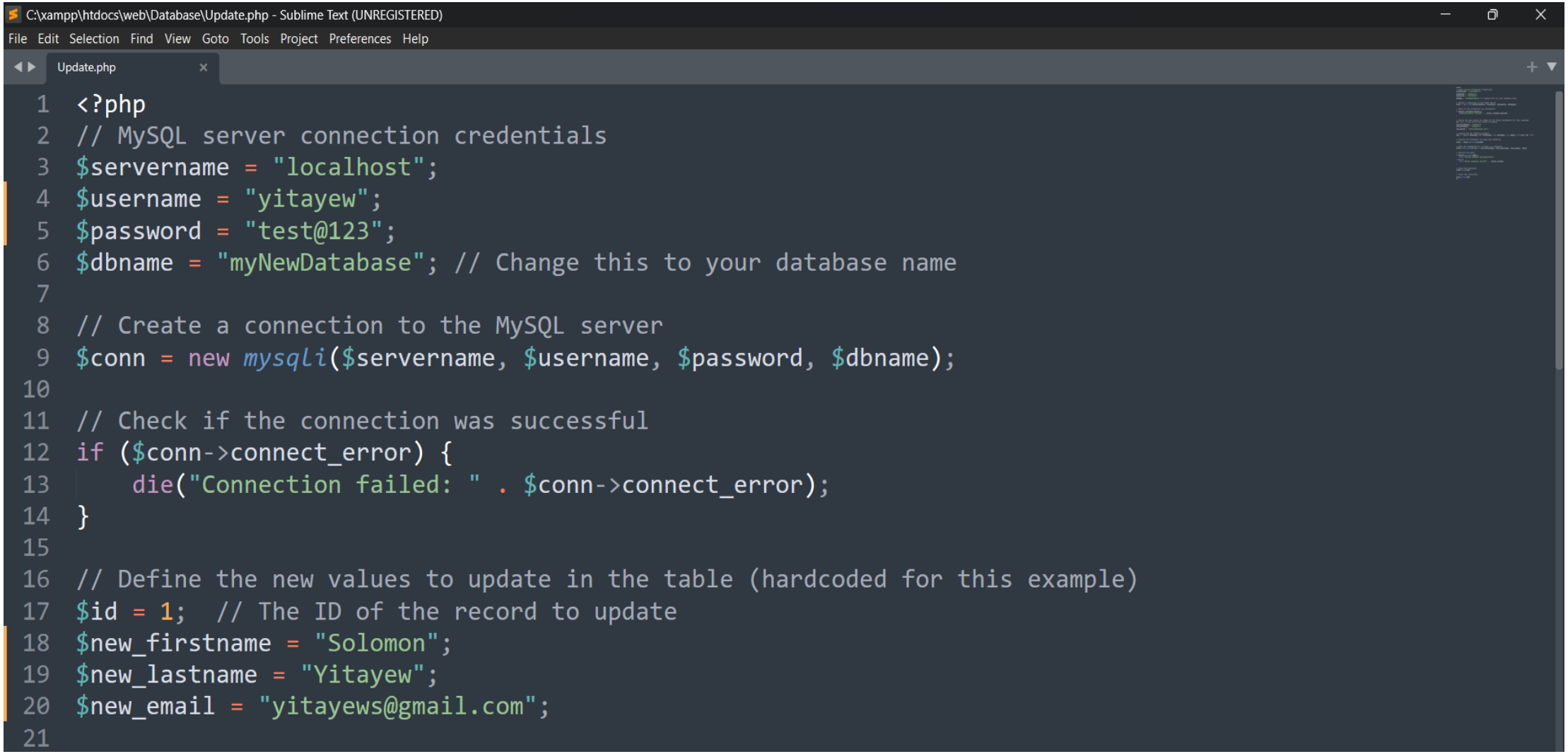
PHP MySQL Update Data

To update data in a MySQL database using **PHP and MySQLi** without an HTML form, you can directly embed the update logic in your PHP script. Here's a step-by-step guide with an example:

Step-by-Step Guide:

1. **Create a connection** to the MySQL database using MySQLi.
2. **Write the SQL** `UPDATE` **query** with the new values for the columns you want to update.
3. **Execute the query** to update the data in the database.
4. **Check if the update was successful**, and display a message accordingly.

Cont. ...


C:\xampp\htdocs\web\Database\Update.php - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Update.php
<?php
// MySQL server connection credentials
$servername = "localhost";
$username = "yitayew";
$password = "test@123";
$dbname = "myNewDatabase"; // Change this to your database name
// Create a connection to the MySQL server
$conn = new mysqli($servername, $username, $password, $dbname);
// Check if the connection was successful
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}
// Define the new values to update in the table (hardcoded for this example)
$id = 1; // The ID of the record to update
$new_firstname = "Solomon";
$new_lastname = "Yitayew";
$new_email = "yitayews@gmail.com";

Cont. ...

C:\xampp\htdocs\web\Database\Update.php - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

Update.php

```php
// Prepare the SQL UPDATE statement
$sql = "UPDATE MyGuests SET firstname = ?, lastname = ?, email = ? WHERE id = ?";

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

// Bind the parameters (s = string, i = integer)
$stmt->bind_param("sssi", $new_firstname, $new_lastname, $new_email, $id);

// Execute the query
if ($stmt->execute()) {
    echo "Record updated successfully!";
} else {
    echo "Error updating record: " . $conn->error;
}

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

// Close the connection
$conn->close();
?>
```

Record updated successfully!

←T→	id	firstname	lastname	email	reg_date
☐ Edit Copy Delete	1	Solomon	Yitayew	yitayews@gmail.com	2024-10-09 09:31:47

Cont. ...

Explanation:

1. **Database Connection:**
   - The script connects to the MySQL database using the MySQLi object-oriented method.
   - If the connection fails, it throws an error using `die()`.
2. **SQL `UPDATE` Query:**
   - The `UPDATE` query is used to modify the `firstname`, `lastname`, and `email` fields of a specific row in the `MyGuests` table, where the `id` matches a specific value.
   - The SQL statement includes placeholders (`?`) for the values to be updated, which are later bound to variables using `bind_param()`.

Cont. ...

3. **Prepared Statements**:
    - Prepared statements help to prevent SQL injection. The `bind_param()` function is used to bind the new values to the placeholders in the SQL query.
    - The `sssi` parameter in `bind_param()` indicates the types of the bound variables:
        - **s** for string (`firstname` , `lastname` , `email`)
        - **i** for integer (`id`)
4. **Executing the Query**:
    - The query is executed using `$stmt->execute()` , and the script checks whether the update was successful.
5. **Closing the Connection**:
    - Once the query execution is complete, both the statement and the database connection are closed.

Cont. ...

Database Structure:

Ensure you have a table like this in your MySQL database:

sql

Copy code

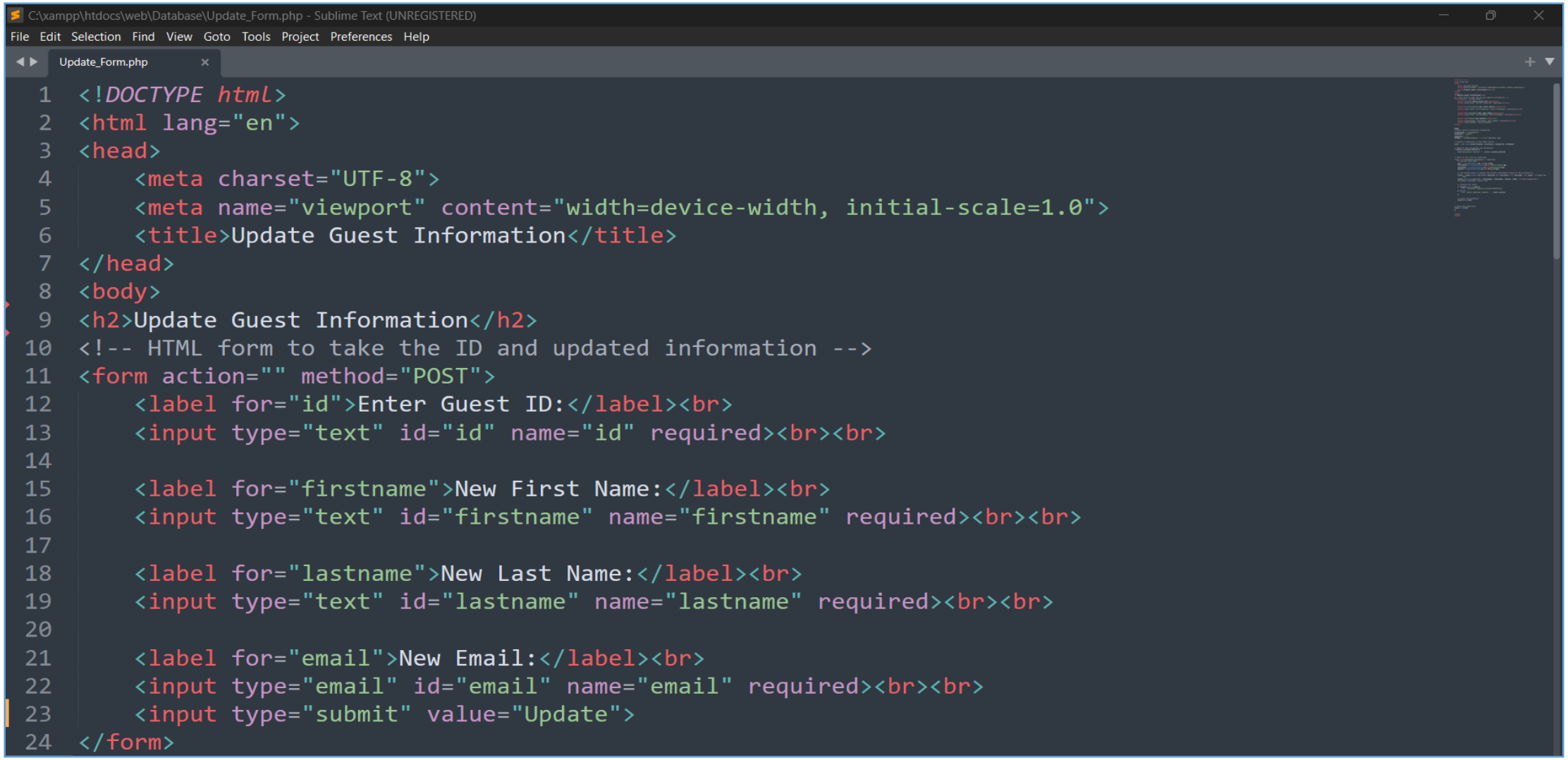
```sql
CREATE TABLE MyGuests (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    firstname VARCHAR(30) NOT NULL,
    lastname VARCHAR(30) NOT NULL,
    email VARCHAR(50),
    reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);
```

Update Using Form

To **update data** in a MySQL database using PHP, the general steps are as follows:

1. **Connect to the MySQL database** using PHP.
2. **Create an HTML form** to take user input (i.e., the data you want to update).
3. **Write an SQL** `UPDATE` **query** using the input from the form.
4. **Execute the query** to update the data in the MySQL database.

Cont. ...


C:\xampp\htdocs\web\Database\Update_Form.php - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Update_Form.php
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>Update Guest Information</title>
7 </head>
8 <body>
9 <h2>Update Guest Information</h2>
10 <!-- HTML form to take the ID and updated information -->
11 <form action="" method="POST">
12 <label for="id">Enter Guest ID:</label>
13 <input type="text" id="id" name="id" required>

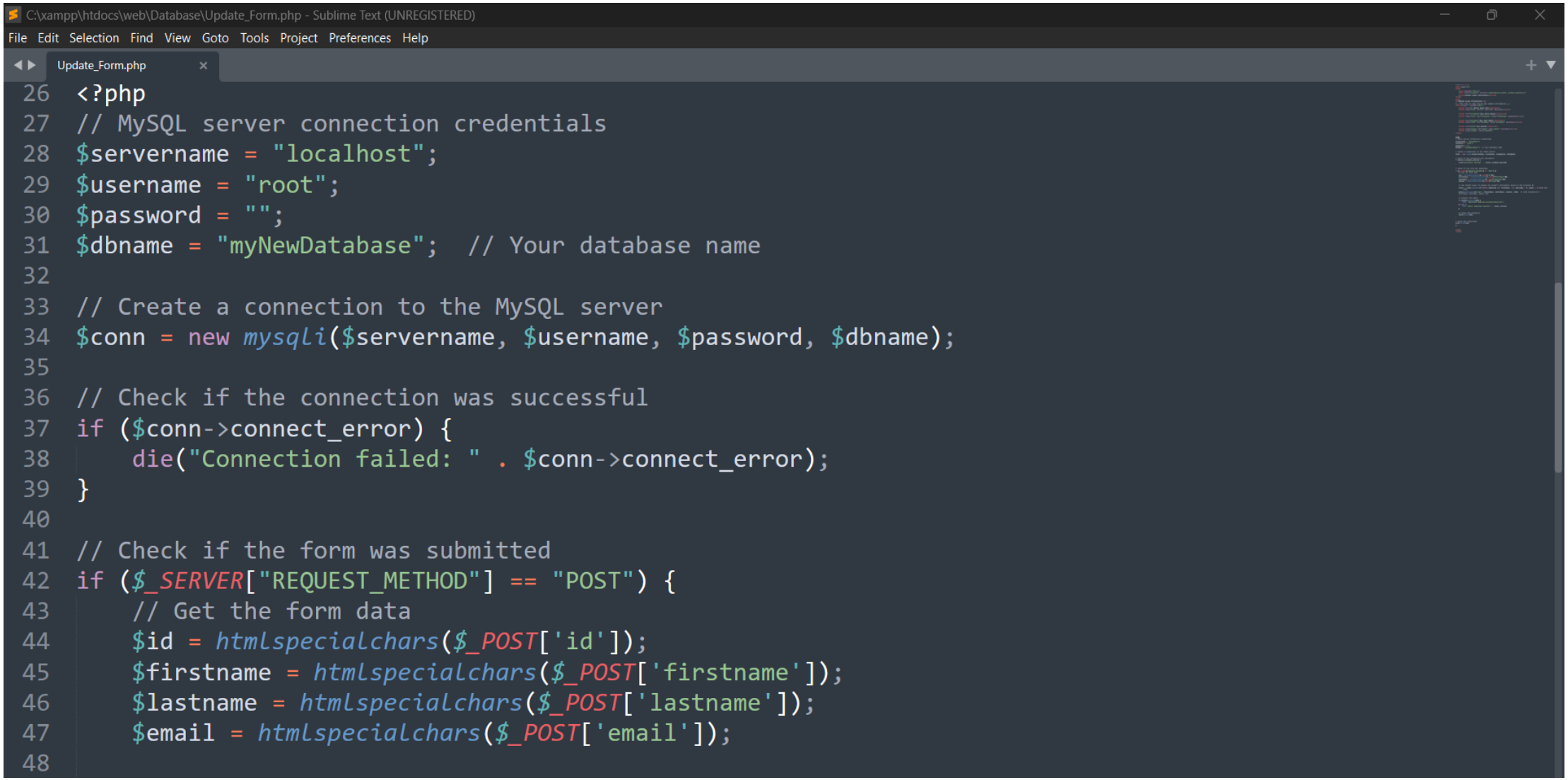
14
15 <label for="firstname">New First Name:</label>
16 <input type="text" id="firstname" name="firstname" required>

17
18 <label for="lastname">New Last Name:</label>
19 <input type="text" id="lastname" name="lastname" required>

20
21 <label for="email">New Email:</label>
22 <input type="email" id="email" name="email" required>

23 <input type="submit" value="Update">
24 </form>

Cont. ...


C:\xampp\htdocs\web\Database\Update_Form.php - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Update_Form.php
26 <?php
27 // MySQL server connection credentials
28 $servername = "localhost";
29 $username = "root";
30 $password = "";
31 $dbname = "myNewDatabase"; // Your database name
32
33 // Create a connection to the MySQL server
34 $conn = new mysqli($servername, $username, $password, $dbname);
35
36 // Check if the connection was successful
37 if ($conn->connect_error) {
38   die("Connection failed: " . $conn->connect_error);
39 }
40
41 // Check if the form was submitted
42 if ($_SERVER["REQUEST_METHOD"] == "POST") {
43   // Get the form data
44     $id = htmlspecialchars($_POST['id']);
45     $firstname = htmlspecialchars($_POST['firstname']);
46     $lastname = htmlspecialchars($_POST['lastname']);
47     $email = htmlspecialchars($_POST['email']);
48

Cont. ...

C:\xampp\htdocs\web\Database\Update_Form.php - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

Update_Form.php

```php
    // SQL UPDATE query to update the guest's information based on the provided ID
    $stmt = $conn->prepare("UPDATE MyGuests SET firstname = ?, lastname = ?, email = ? WHERE id =
        ?");
    $stmt->bind_param("sssi", $firstname, $lastname, $email, $id);  // Bind parameters (
    firstname, lastname, email, id)

    // Execute the query
    if ($stmt->execute()) {
        echo "<p>Record updated successfully!</p>";
    } else {
        echo "Error updating record: " . $conn->error;
    }

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

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

</body>
</html>
```

Update Guest Information

Enter Guest ID:

New First Name:

New Last Name:

New Email:

Update

Record updated successfully!

☐	Edit	Copy	Delete	9	Martha	Solomon	marta123@gmail.com	2024-10-09 09:44:32

Cont. ...

1. **HTML Form:**
   - The form takes **four inputs**: the `ID` of the record you want to update, the new `firstname`, `lastname`, and `email`.
   - The form uses the `POST` method to submit data to the server.
2. **PHP Script:**
   - **Database Connection**: The script connects to the MySQL database using the `mysqli` object.
   - **Form Handling**: When the form is submitted, the `POST` data is retrieved and sanitized using `htmlspecialchars()` to prevent XSS.
   - **Prepared Statement**: The `UPDATE` query is written using a prepared statement, which helps protect against SQL injection. The query updates the `firstname`, `lastname`, and `email` fields for the record with the specified `id`.
   - **Execution and Feedback**: If the query is successful, a success message is displayed. If there's an error, the error message is shown.

Cont. ...

3. SQL `UPDATE` Query:

   - The SQL query updates the fields in the `MyGuests` table:

```sql
UPDATE MyGuests SET firstname = ?, lastname = ?, email = ? WHERE id = ?
```

   - The `?` placeholders are used for the values that will be bound dynamically through the PHP `bind_param()` method.

Cont. ...

Database Structure:

Ensure the `MyGuests` table exists in your MySQL database and has the following structure:

sql Copy code

```sql
CREATE TABLE MyGuests (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    firstname VARCHAR(30) NOT NULL,
    lastname VARCHAR(30) NOT NULL,
    email VARCHAR(50),
    reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);
```

Cont. ...

Summary:

- This code demonstrates how to **update a record in a MySQL database** using PHP and an HTML form.
- It uses **prepared statements** to prevent SQL injection.
- The result of the update operation is displayed on the page (whether it was successful or if an error occurred).

You can customize the form fields and SQL query as needed for different database tables and columns.

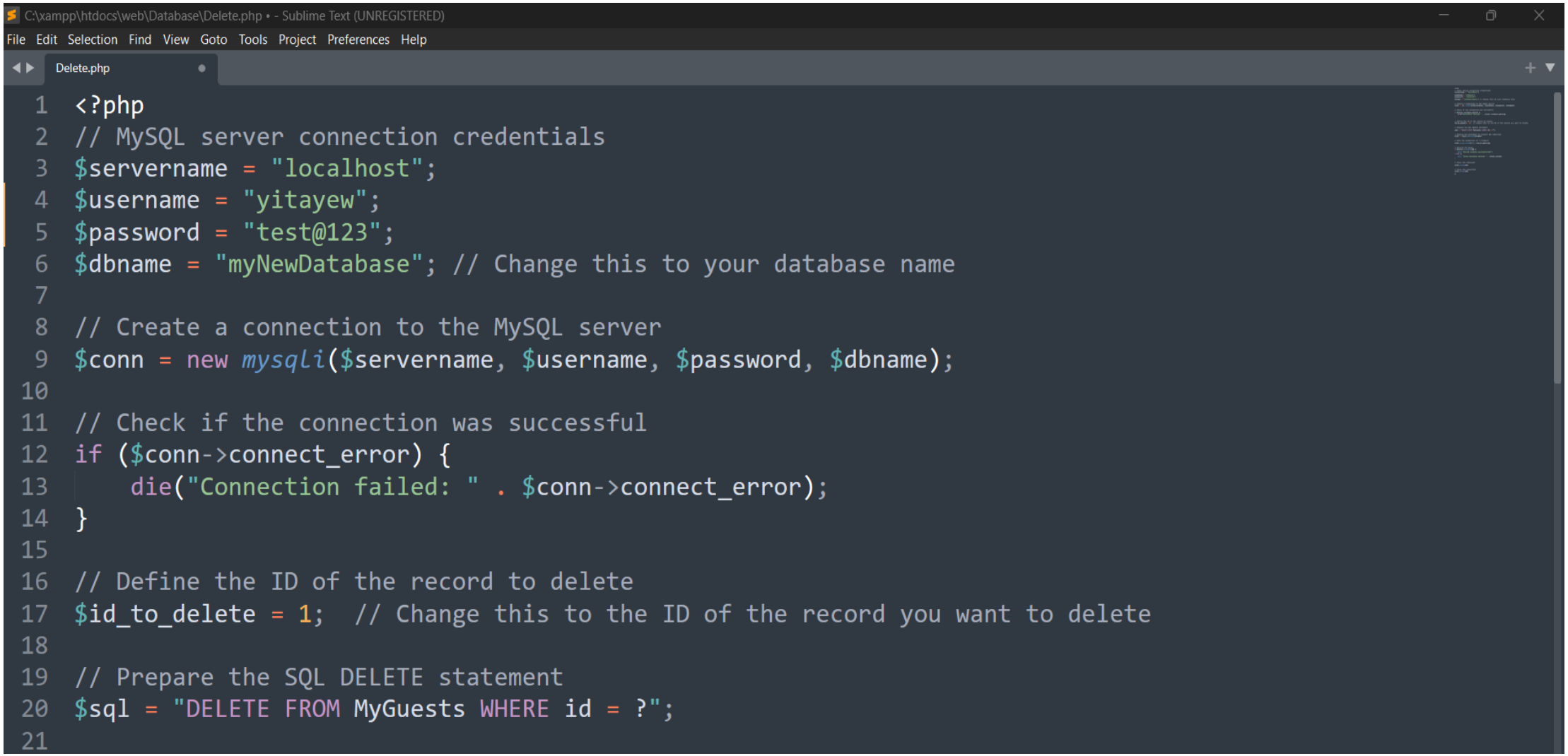
PHP MySQL Delete Data

To delete data from a MySQL database using **PHP and MySQLi**, you can use the `DELETE` SQL statement. Below is a step-by-step guide on how to delete data, along with an example.

Steps to Delete Data:

1. **Connect** to the MySQL database using PHP.
2. **Write an SQL** `DELETE` **query** to specify which record(s) to delete.
3. **Execute the query** to remove the data from the database.
4. **Verify** if the deletion was successful.

Cont. ...


C:\xampp\htdocs\web\Database\Delete.php • - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Delete.php
<?php
// MySQL server connection credentials
$servername = "localhost";
$username = "yitayew";
$password = "test@123";
$dbname = "myNewDatabase"; // Change this to your database name
// Create a connection to the MySQL server
$conn = new mysqli($servername, $username, $password, $dbname);
// Check if the connection was successful
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}
// Define the ID of the record to delete
$id_to_delete = 1; // Change this to the ID of the record you want to delete
// Prepare the SQL DELETE statement
$sql = "DELETE FROM MyGuests WHERE id = ?";

Cont. ...

C:\xampp\htdocs\web\Database\Delete.php • - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

Delete.php

```php
// Prepare the statement to prevent SQL injection
$stmt = $conn->prepare($sql);

// Bind the parameters (i = integer)
$stmt->bind_param("i", $id_to_delete);

// Execute the query
if ($stmt->execute()) {
    echo "Record deleted successfully!";
} else {
    echo "Error deleting record: " . $conn->error;
}

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

// Close the connection
$conn->close();
?>
```

Record deleted successfully!

	id	firstname	lastname	email	reg_date
Edit Copy Delete	2	Haileab	Solomon	haileabsolomon@gmail.com	2024-10-07 11:12:31
Edit Copy Delete	3	Yared	Solomon	yaredsolomon@gmail.com	2024-10-07 11:12:31
Edit Copy Delete	4	Natanim	Yitayew	natanimyitayew@gmail.com	2024-10-07 11:20:57

Cont. ...

Explanation:

1. **Database Connection**:
   - The script establishes a connection to the MySQL database using MySQLi's object-oriented method.
   - If the connection fails, it outputs an error message and terminates the script using `die()`.
2. **SQL `DELETE` Query**:
   - The `DELETE` query is structured to remove a row from the `MyGuests` table based on the `id` column:

```sql
DELETE FROM MyGuests WHERE id = ?
```

   - The `?` is a placeholder for the `id` value, which will be provided dynamically through PHP.

Cont. ...

3. **Prepared Statements:**
    - Prepared statements are used to prevent SQL injection.
    - The `bind_param()` function binds the `$id_to_delete` variable to the `?` placeholder in the SQL query. In this case, the variable type is `i` (integer).
4. **Executing the Query:**
    - The query is executed using `$stmt->execute()`. If successful, a message indicating successful deletion is displayed. If there's an error, the script outputs the error message.
5. **Closing the Connection:**
    - After executing the query, the script closes both the prepared statement and the MySQL database connection.

Cont. ...

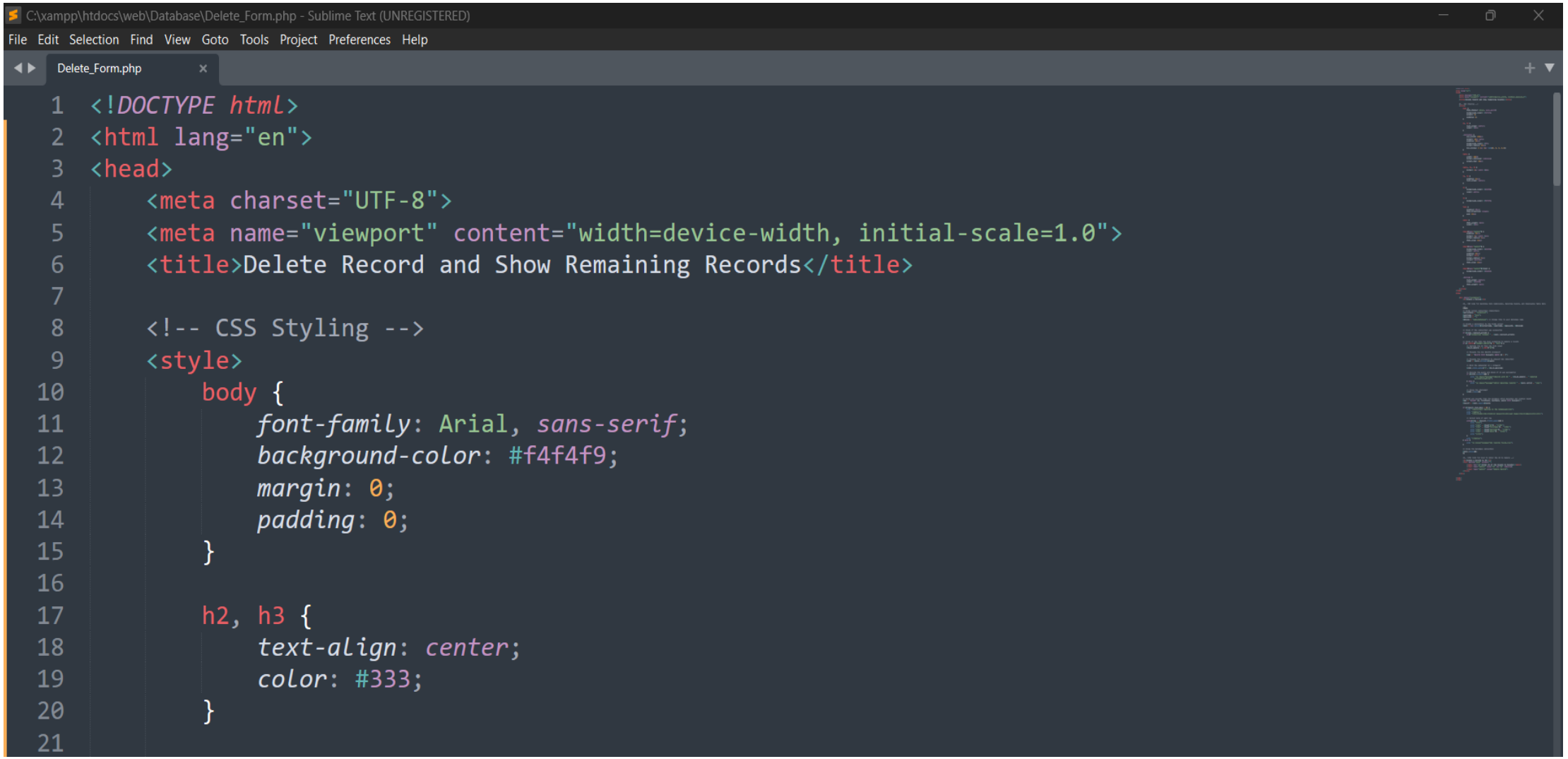
Database Structure:

The table in this example is called `MyGuests` and has the following structure:

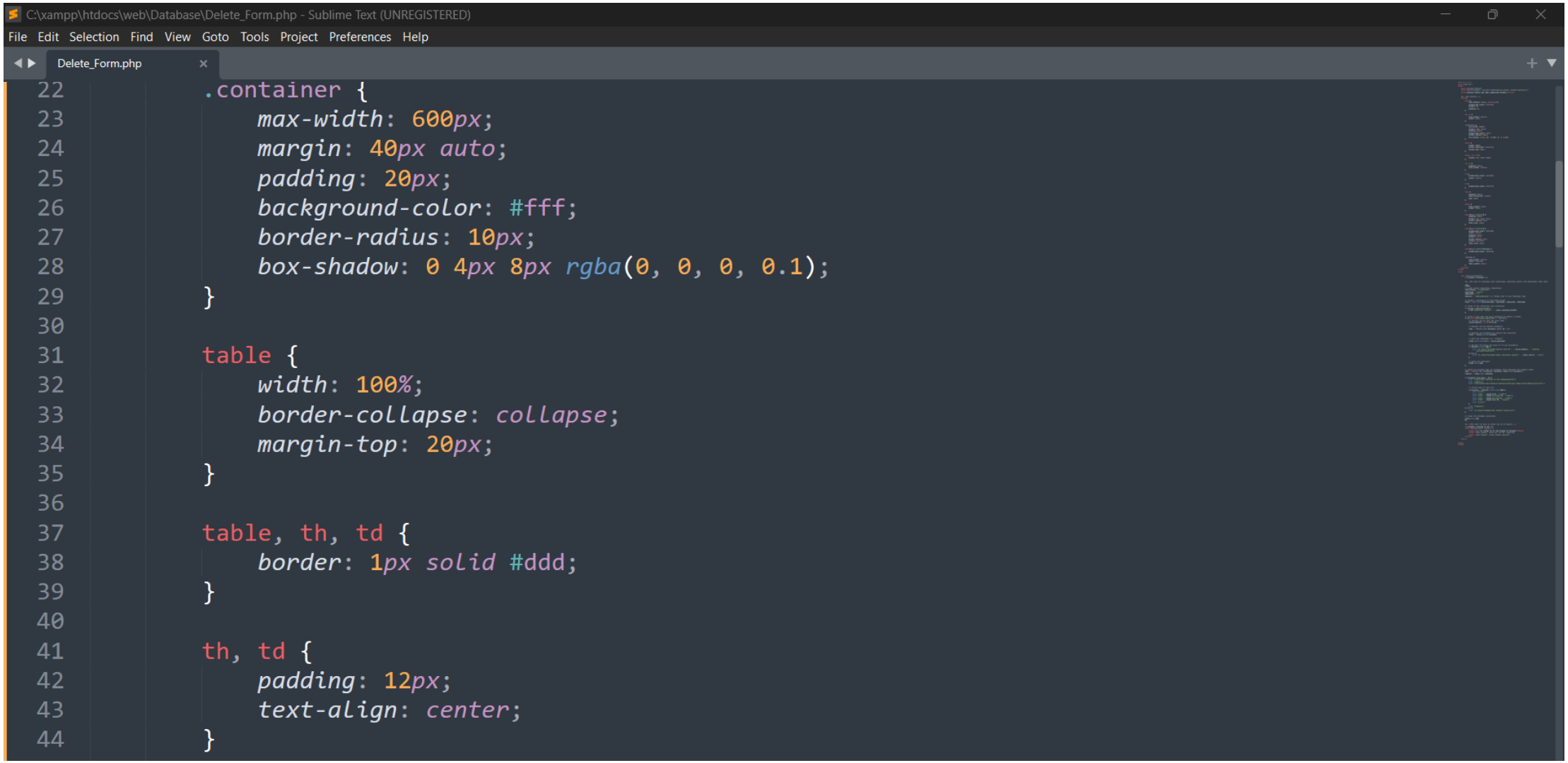
sql Copy code

```sql
CREATE TABLE MyGuests (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    firstname VARCHAR(30) NOT NULL,
    lastname VARCHAR(30) NOT NULL,
    email VARCHAR(50),
    reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);
```

Deleting using Form



Cont. ...


C:\xampp\htdocs\web\Database\Delete_Form.php - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Delete_Form.php
22 .container {
23 max-width: 600px;
24 margin: 40px auto;
25 padding: 20px;
26 background-color: #fff;
27 border-radius: 10px;
28 box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
29 }
30
31 table {
32 width: 100%;
33 border-collapse: collapse;
34 margin-top: 20px;
35 }
36
37 table, th, td {
38 border: 1px solid #ddd;
39 }
40
41 th, td {
42 padding: 12px;
43 text-align: center;
44 }

Cont. ...

C:\xampp\htdocs\web\Database\Delete_Form.php - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

Delete_Form.php

```
46      th {
47          background-color: #4CAF50;
48          color: white;
49      }
50
51      td {
52          background-color: #f9f9f9;
53      }
54
55      form {
56          display: flex;
57          flex-direction: column;
58          gap: 10px;
59      }
60
61      label {
62          font-weight: bold;
63          color: #333;
64      }
65
```

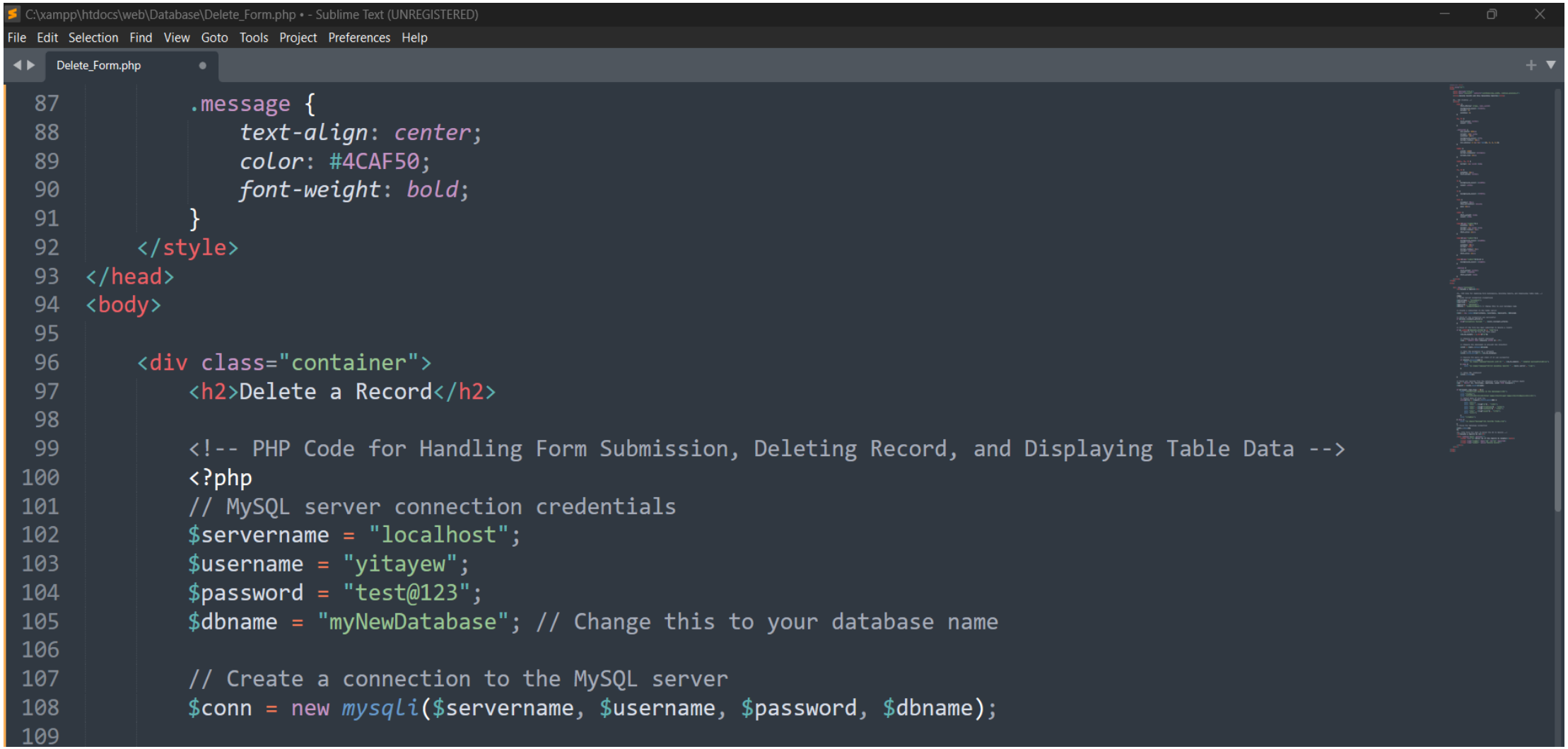
C:\xampp\htdocs\web\Database\Delete_Form.php - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

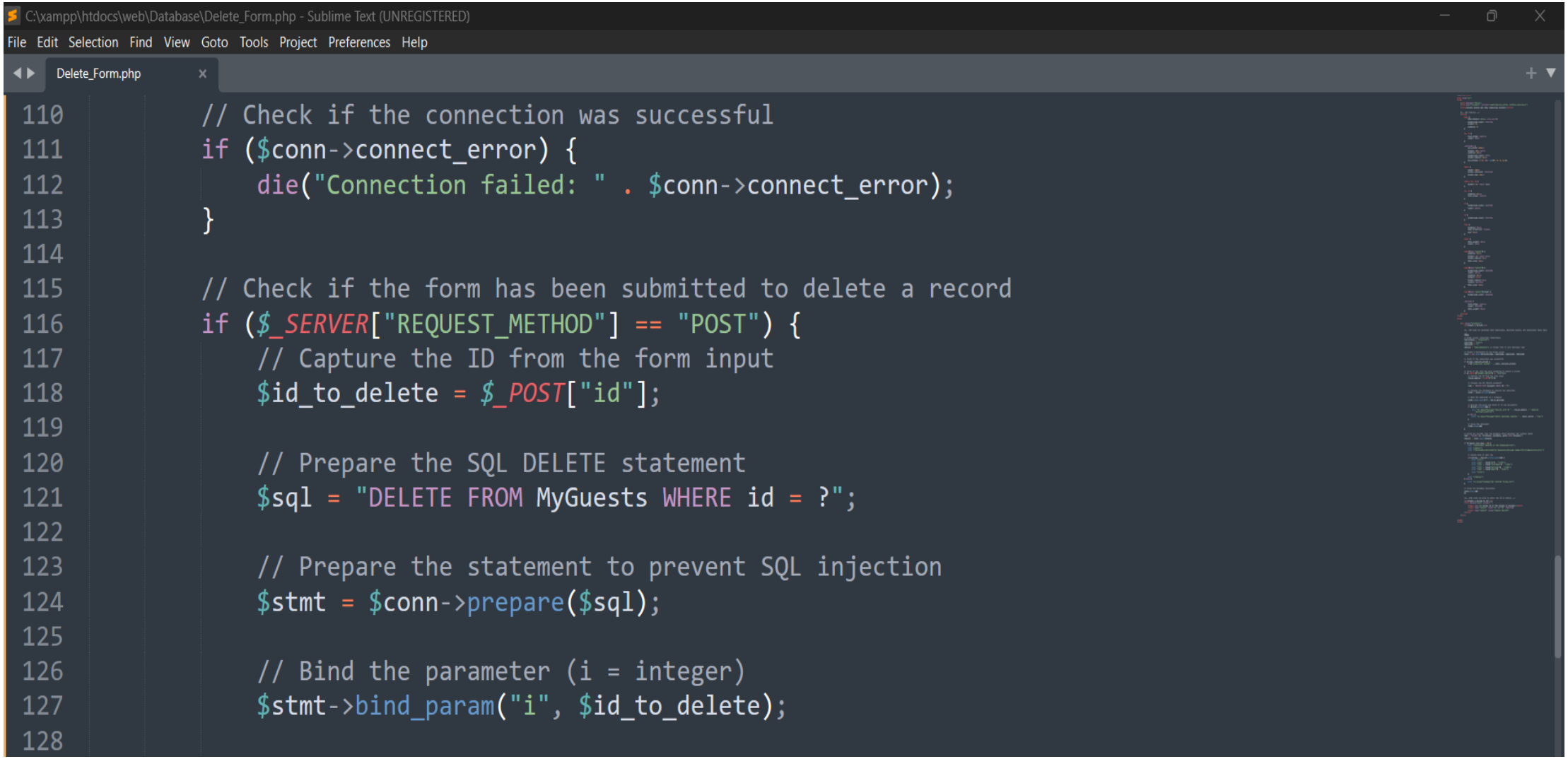
Delete_Form.php

```
66      input[type="number"] {
67          padding: 10px;
68          border: 1px solid #ccc;
69          border-radius: 5px;
70          font-size: 16px;
71      }
72
73      input[type="submit"] {
74          background-color: #4CAF50;
75          color: white;
76          padding: 10px;
77          border: none;
78          border-radius: 5px;
79          cursor: pointer;
80          font-size: 16px;
81      }
82
83      input[type="submit"]:hover {
84          background-color: #45a049;
85      }
86
```

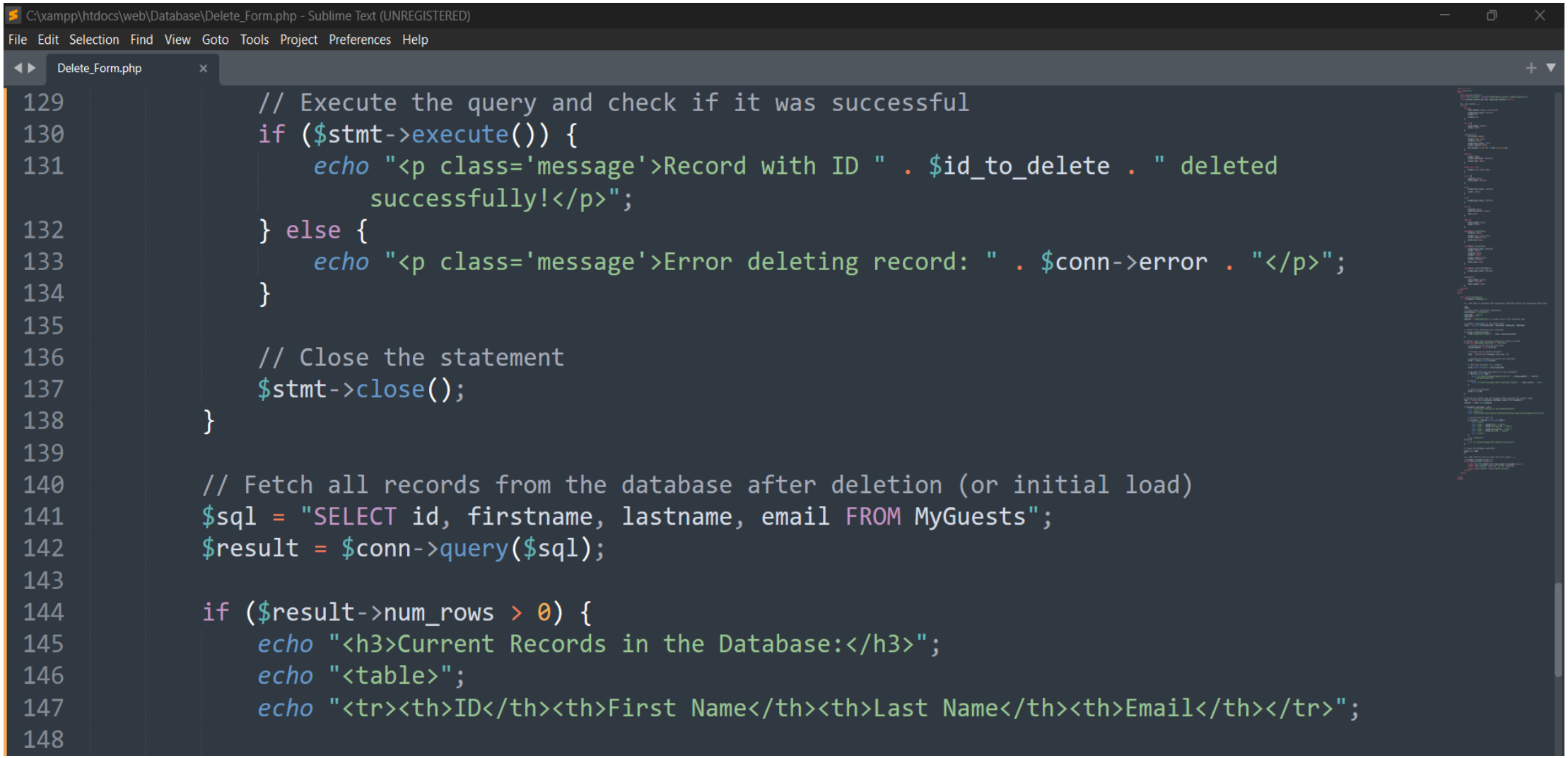
Cont. ...


C:\xampp\htdocs\web\Database\Delete_Form.php • - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Delete_Form.php
87 .message {
88 text-align: center;
89 color: #4CAF50;
90 font-weight: bold;
91 }
92 </style>
93 </head>
94 <body>
95
96
97 <h2>Delete a Record</h2>
98
99 <!-- PHP Code for Handling Form Submission, Deleting Record, and Displaying Table Data -->
100 <?php
101 // MySQL server connection credentials
102 $servername = "localhost";
103 $username = "yitayew";
104 $password = "test@123";
105 $dbname = "myNewDatabase"; // Change this to your database name
106
107 // Create a connection to the MySQL server
108 $conn = new mysqli($servername, $username, $password, $dbname);
109

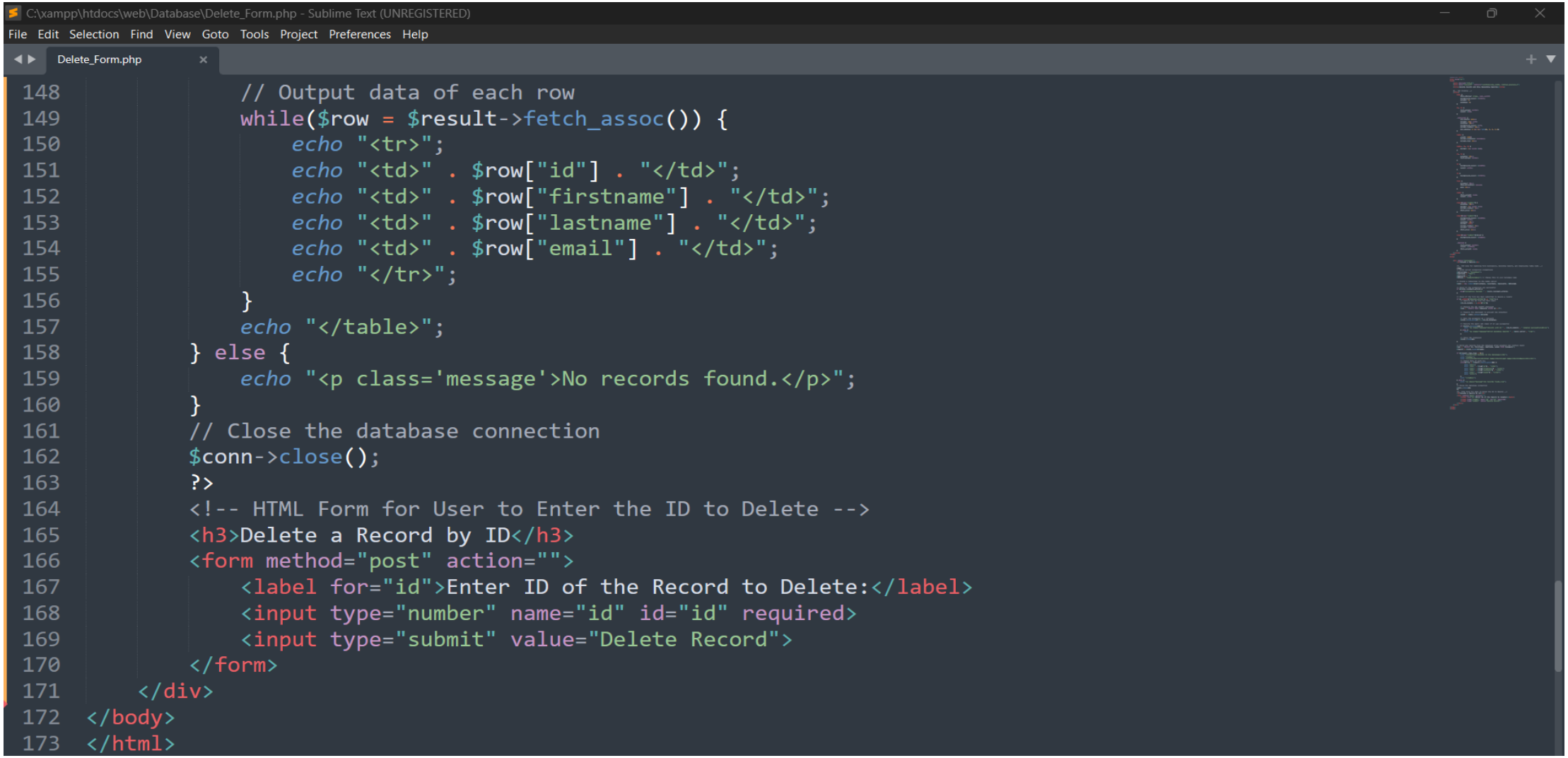
Cont. ...


C:\xampp\htdocs\web\Database\Delete_Form.php - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Delete_Form.php
110 // Check if the connection was successful
111 if ($conn->connect_error) {
112 die("Connection failed: " . $conn->connect_error);
113 }
114
115 // Check if the form has been submitted to delete a record
116 if ($_SERVER["REQUEST_METHOD"] == "POST") {
117 // Capture the ID from the form input
118 $id_to_delete = $_POST["id"];
119
120 // Prepare the SQL DELETE statement
121 $sql = "DELETE FROM MyGuests WHERE id = ?";
122
123 // Prepare the statement to prevent SQL injection
124 $stmt = $conn->prepare($sql);
125
126 // Bind the parameter (i = integer)
127 $stmt->bind_param("i", $id_to_delete);
128

Cont. ...


C:\xampp\htdocs\web\Database\Delete_Form.php - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Delete_Form.php
129 // Execute the query and check if it was successful
130 if ($stmt->execute()) {
131 echo " Record with ID " . $id_to_delete . " deleted successfully! ";
132 } else {
133 echo " Error deleting record: " . $conn->error . " ";
134 }
135
136 // Close the statement
137 $stmt->close();
138 }
139
140 // Fetch all records from the database after deletion (or initial load)
141 $sql = "SELECT id, firstname, lastname, email FROM MyGuests";
142 $result = $conn->query($sql);
143
144 if ($result->num_rows > 0) {
145 echo "<h3>Current Records in the Database:</h3>";
146 echo "<table>";
147 echo "<tr><th>ID</th><th>First Name</th><th>Last Name</th><th>Email</th></tr>";
148

Cont. ...


C:\xampp\htdocs\web\Database\Delete_Form.php - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
Delete_Form.php
148 // Output data of each row
149 while($row = $result->fetch_assoc()) {
150 echo "<tr>";
151 echo "<td>" . $row["id"] . "</td>";
152 echo "<td>" . $row["firstname"] . "</td>";
153 echo "<td>" . $row["lastname"] . "</td>";
154 echo "<td>" . $row["email"] . "</td>";
155 echo "</tr>";
156 }
157 echo "</table>";
158 } else {
159 echo " No records found. ";
160 }
161 // Close the database connection
162 $conn->close();
163 ?>
164 <!-- HTML Form for User to Enter the ID to Delete -->
165 <h3>Delete a Record by ID</h3>
166 <form method="post" action="">
167 <label for="id">Enter ID of the Record to Delete:</label>
168 <input type="number" name="id" id="id" required>
169 <input type="submit" value="Delete Record">
170 </form>
171
172 </body>
173 </html>

Cont. ...

Delete a Record

Record with ID 8 deleted successfully!

Current Records in the Database:

ID	First Name	Last Name	Email
2	Haileab	Solomon	haileabsolomon@gmail.com
3	Yared	Solomon	yaredsolomon@gmail.com
4	Natanim	Yitayew	natanimyitayew@gmail.com
5	Dureti	Guye	duretiguye@gmail.com
6	yitayew	Solomon	yitayewsolomon3@gmail.com
7	Mihiret	Gashaw	yitayewsolomon3@gmail.com

Delete a Record by ID

Enter ID of the Record to Delete:

Delete Record

Delete a Record

Record with ID 8 deleted successfully!

Current Records in the Database:

ID	First Name	Last Name	Email
2	Haileab	Solomon	haileabsolomon@gmail.com
3	Yared	Solomon	yaredsolomon@gmail.com
4	Natanim	Yitayew	natanimyitayew@gmail.com
5	Dureti	Guye	duretiguye@gmail.com
6	yitayew	Solomon	yitayewsolomon3@gmail.com
7	Mihiret	Gashaw	yitayewsolomon3@gmail.com

Delete a Record by ID

Enter ID of the Record to Delete:

8

Delete Record

Cont. ...

1. **HTML Form**:
    - The user inputs the `ID` of the record they want to delete via a form. The form is submitted using the `POST` method to the same page (`action=""`).
2. **PHP Script**:
    - **Database Connection**: A connection to the MySQL database is created at the start of the script using the provided credentials.
    - **Delete Logic**: If the form is submitted (`if ($_SERVER["REQUEST_METHOD"] == "POST")`), the script retrieves the `ID` from the form input and prepares the `DELETE` SQL query. It executes the query to delete the record where the `id` matches the input. Feedback is provided on whether the deletion was successful.
    - **Fetching and Displaying Records**: After deleting (or if no deletion has been done), the script retrieves the remaining records from the `MyGuests` table using a `SELECT` query. If there are records, they are displayed in an HTML table. If no records are found, a message saying "No records found" is displa

Cont. ...

3. **Dynamic Table:**
    - The script uses the `while($row = $result->fetch_assoc())` loop to iterate over each row in the `MyGuests` table and outputs the data into an HTML table.
4. **User Feedback:**
    - If the deletion is successful, a message such as `Record with ID 1 deleted successfully!` is shown.
    - After that, the remaining records are displayed in the table format.

Cont. ...

Database Table Example:

Ensure you have a table like this in your MySQL database for testing:

sql Copy code

```sql
CREATE TABLE MyGuests (
    id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    firstname VARCHAR(30) NOT NULL,
    lastname VARCHAR(30) NOT NULL,
    email VARCHAR(50),
    reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);
```

Cont. ...

CSS Enhancements:

1. **Body Styling**:
   - `background-color: #f4f4f9;` provides a light background to make the content stand out.
   - The font family is set to `Arial, sans-serif` for a clean and readable appearance.
2. **Container Styling**:
   - The `.container` class centers the content, adds padding, a white background, and a subtle shadow to make it look like a card.
   - The form and table are placed inside this container.
3. **Table Styling**:
   - The table is styled with borders and padding for a clean look.
   - `th` (table headers) have a green background, and `td` (table data) alternate with a light background for contrast.

Cont. ...

4. **Form Styling**:
    - The form uses a `flex` layout for vertical alignment, with space between the form fields.
    - Input fields (`<input type="number">` and submit button) are styled with padding, border radius, and custom hover effects for the button.
5. **Message Styling**:
    - The success message after deletion is displayed with a green color and centered on the page.

How It Looks:

- The **form** is clean, centered, and responsive, with padding for better usability.
- The **table** displaying the records looks organized, with alternating row colors to improve readability.
- After a successful deletion, the user will see a clear success message, and the updated table will show the remaining records.

Thank you!

Appreciate your action.