

WEB DESIGN AND DEVELOPMENT

LAB FILE

(Subject Code : 18CSC311J)

B.Tech III Year / 6th Semester

Name: Rohan Singh

Reg No. RA2011028030003



Submitted to: Dr. Ashwani Kumar

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

SRM INSTITUTE OF SCIENCE & TECHNOLOGY, DELHI-NCR CAMPUS,
MODINAGAR SIKRI KALAN, DELHI MEERUT ROAD, DIST-GHAZIYABAD-
201204

Even Semester(2022-2023)

BONAFIDE CERTIFICATE

Registration no. RA2011028030003

Certified to be the bonafied record of work done by Rohan Singh of 6th Semester, 3rd Year B.Tech degree course in **SRM INSTITUTE OF SCIENCE & TECHNOLOGY, DELHI NCR** Campus for the Department of CSE(Cloud Computing), in **WEB DESIGNE AND DEVELOPMENT LABORATORY** during the academic year 2022-2023

Subject In charge

Head of the Department

Submitted for the end semester examination held on __/__/__
SRM INSTITUTE OF SCIENCE & TECHNOLOGY, DELHI NCR
Campus

Internal Examiner-I

Internal Examiner-II

INDEX

Exp. No.	Title of Experiment	Date of Experiment	Date of Completion of Experiment	Teacher's Signature
1	a) HTML program to work with Lists b) HTML program to work with tables	05-01-23	12-01-23	
2	a) HTML program to design login page, registration page b) HTML program to design feedback form	12-01-23	19-01-23	
3	a) CSS program to work with background and border properties b) JavaScript program to print multiplication table of given integer c) JavaScript program to validate the registration from contents with rules	19-01-23	09-02-23	
4	a) JDBC program to preform various DML Operations on the database using Statement	09-02-23	23-02-23	
5	a) Servlet program to read the parameters from user interface and display welcome message b) Servlet program to work with HttpSession Object c) Demo on generating dynamic content with form processing using Servlets	23-02-23	02-03-23	
6	a) JSP program to print multiplication table b) JSP program to handle the exception c) JSP program to retrieve the student data from database based on his roll number	02-03-23	09-03-23	
7	a) PHP program to work with associative arrays b) PHP program to find factorial using recursion c) PHP program to perform various DDL operations on MySQL database	09-03-23	16-03-23	
8	a) Write a JQuery Script to implement hide() and show() effects. b) Write a JQuery Script to apply various sliding effects	16-03-23	23-03-23	
9	a) Implement shopping cart with AngularJS b) Write a program to display data in tables in various forms	23-03-23	06-04-23	
10	a) Explain the step-by-step process how to create database & collection in MongoDB. Create a collection with student name and store 5 rows in that collection	06-04-23	13-04-23	

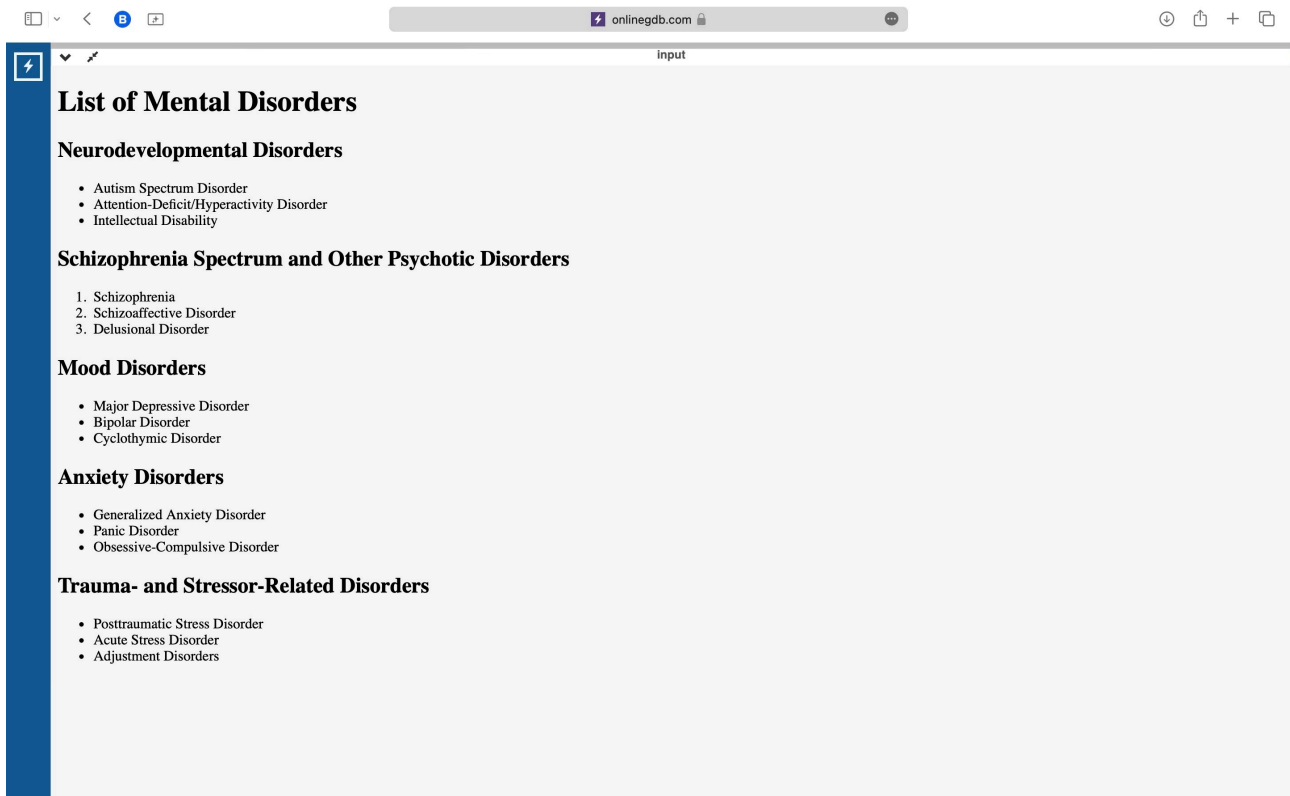
Experiment 1

HTML Program to work with Lists.

Code:

```
<!DOCTYPE html>
<html>
<head>
  <title>List of Mental Disorders</title>
</head>
<body>
  <h1>List of Mental Disorders</h1>
  <h2>Neurodevelopmental Disorders</h2>
  <ul>
    <li>Autism Spectrum Disorder</li>
    <li>Attention-Deficit/Hyperactivity Disorder</li>
    <li>Intellectual Disability</li>
  </ul>
  <h2>Schizophrenia Spectrum and Other Psychotic Disorders</h2>
  <ol>
    <li>Schizophrenia</li>
    <li>Schizoaffective Disorder</li>
    <li>Delusional Disorder</li>
  </ol>
  <h2>Mood Disorders</h2>
  <ul>
    <li>Major Depressive Disorder</li>
    <li>Bipolar Disorder</li>
    <li>Cyclothymic Disorder</li>
  </ul>
  <h2>Anxiety Disorders</h2>
  <ul>
    <li>Generalized Anxiety Disorder</li>
    <li>Panic Disorder</li>
    <li>Obsessive-Compulsive Disorder</li>
  </ul>
  <h2>Trauma- and Stressor-Related Disorders</h2>
  <ul>
    <li>Posttraumatic Stress Disorder</li>
    <li>Acute Stress Disorder</li>
    <li>Adjustment Disorders</li>
  </ul>
</body>
</html>
```

Output:



HTML Program to work with tables.

Code:

```
<!DOCTYPE html>
<html>
<head>
    <title>List of Mental Disorders</title>
    <style>
        table {
            border-collapse: collapse;
            width: 100%;
        }

        th, td {
            text-align: left;
            padding: 8px;
            border: 1px solid black;
        }

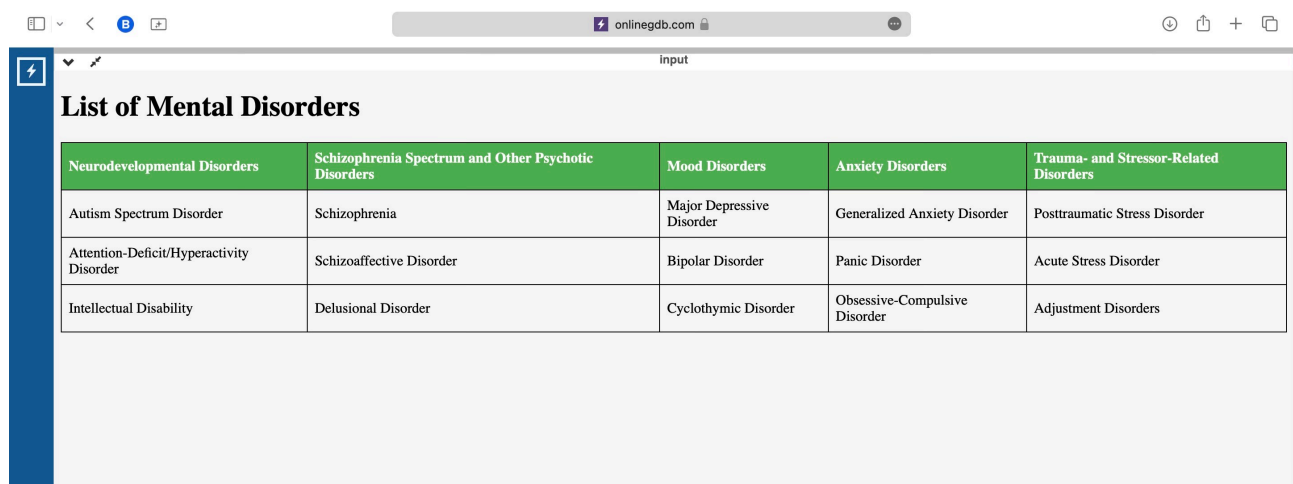
        th {
            background-color: #4CAF50;
            color: white;
        }
    </style>
</head>
```

```

<body>
  <h1>List of Mental Disorders</h1>
  <table>
    <thead>
      <tr>
        <th>Neurodevelopmental Disorders</th>
        <th>Schizophrenia Spectrum and Other Psychotic Disorders</th>
        <th>Mood Disorders</th>
        <th>Anxiety Disorders</th>
        <th>Trauma- and Stressor-Related Disorders</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>Autism Spectrum Disorder</td>
        <td>Schizophrenia</td>
        <td>Major Depressive Disorder</td>
        <td>Generalized Anxiety Disorder</td>
        <td>Posttraumatic Stress Disorder</td>
      </tr>
      <tr>
        <td>Attention-Deficit/Hyperactivity Disorder</td>
        <td>Schizoaffective Disorder</td>
        <td>Bipolar Disorder</td>
        <td>Panic Disorder</td>
        <td>Acute Stress Disorder</td>
      </tr>
      <tr>
        <td>Intellectual Disability</td>
        <td>Delusional Disorder</td>
        <td>Cyclothymic Disorder</td>
        <td>Obsessive-Compulsive Disorder</td>
        <td>Adjustment Disorders</td>
      </tr>
    </tbody>
  </table>
</body>
</html>

```

Output:



Neurodevelopmental Disorders	Schizophrenia Spectrum and Other Psychotic Disorders	Mood Disorders	Anxiety Disorders	Trauma- and Stressor-Related Disorders
Autism Spectrum Disorder	Schizophrenia	Major Depressive Disorder	Generalized Anxiety Disorder	Posttraumatic Stress Disorder
Attention-Deficit/Hyperactivity Disorder	Schizoaffective Disorder	Bipolar Disorder	Panic Disorder	Acute Stress Disorder
Intellectual Disability	Delusional Disorder	Cyclothymic Disorder	Obsessive-Compulsive Disorder	Adjustment Disorders

Experiment 2

HTML Program to design login page, registration page.

Code:

```
- Login page
<!DOCTYPE html>
<html>
<head>
    <title>Login Page</title>
    <style>
        body {
            font-family: Arial, sans-serif;
        }

        form {
            border: 1px solid #ccc;
            padding: 20px;
            margin: 50px auto;
            max-width: 400px;
        }

        label {
            display: block;
            margin-bottom: 10px;
        }

        input[type="text"], input[type="password"] {
            width: 100%;
            padding: 10px;
            margin-bottom: 20px;
            border: 1px solid #ccc;
            border-radius: 4px;
        }

        button[type="submit"] {
            background-color: #4CAF50;
            color: white;
            padding: 10px 20px;
            border: none;
            border-radius: 4px;
            cursor: pointer;
        }

        button[type="submit"]:hover {
            background-color: #45a049;
        }

        .error {
            color: red;
            margin-bottom: 10px;
        }
```

```

    }
  </style>
</head>
<body>
  <form action="login.php" method="post">
    <h2>Login</h2>
    <label for="username">Username</label>
    <input type="text" id="username" name="username" required>
    <label for="password">Password</label>
    <input type="password" id="password" name="password" required>
    <button type="submit">Login</button>
    <div class="error">Incorrect username or password.</div>
  </form>
</body>
</html>

```

The screenshot shows a web browser window with the address bar displaying 'onlinegdb.com'. The page title is 'input'. The main content area features a login form with a light gray background. The form has a title 'Login' in bold. Below the title are two input fields: 'Username' and 'Password'. A green 'Login' button is positioned below the password field. At the bottom of the form, a red error message reads 'Incorrect username or password.'

- Registration page

```

<!DOCTYPE html>
<html>
<head>
  <title>Registration Form</title>
  <style>
    label {
      display: block;
      margin-top: 10px;
    }

    input[type="text"], input[type="email"], input[type="password"], select {
      padding: 5px;
      border-radius: 5px;
    }
  </style>

```



```

        border: 1px solid #ccc;
        font-size: 16px;
        width: 100%;
        box-sizing: border-box;
        margin-top: 5px;
        margin-bottom: 10px;
    }

    input[type="submit"] {
        background-color: #4CAF50;
        color: white;
        padding: 10px 20px;
        border-radius: 5px;
        border: none;
        cursor: pointer;
        font-size: 16px;
    }

    input[type="submit"]:hover {
        background-color: #3e8e41;
    }

    h1 {
        text-align: center;
    }
</style>
</head>
<body>
    <h1>Registration Form</h1>
    <form>
        <label for="fname">First Name:</label>
        <input type="text" id="fname" name="fname" placeholder="Enter your first
name">

        <label for="lname">Last Name:</label>
        <input type="text" id="lname" name="lname" placeholder="Enter your last
name">

        <label for="email">Email:</label>
        <input type="email" id="email" name="email" placeholder="Enter your email">

        <label for="password">Password:</label>
        <input type="password" id="password" name="password" placeholder="Enter
your password">

        <label for="gender">Gender:</label>
        <select id="gender" name="gender">
            <option value="">--Select--</option>
            <option value="male">Male</option>
            <option value="female">Female</option>
            <option value="other">Other</option>
        </select>
        <input type="submit" value="Register">
    </form>
</body>
</html>

```

Output:

The screenshot shows a web browser window with the address bar displaying 'onlinegdb.com'. The page title is 'input'. The main content is a 'Registration Form' with the following fields and values:

- First Name: Harshit
- Last Name: Agarwal
- Email: harshitagarawal27@gmail.com
- Password: (masked with dots)
- Gender: Male

A green 'Register' button is located at the bottom left of the form.

HTML program to design feedback form.

Code:

```
<!DOCTYPE html>
<html>
<head>
  <title>Feedback Form</title>
  <style>
    body {
      font-family: Arial, sans-serif;
    }

    form {
      border: 1px solid #ccc;
      padding: 20px;
      margin: 50px auto;
      max-width: 600px;
    }

    label {
      display: block;
      margin-bottom: 10px;
    }

    textarea {
      width: 100%;
      padding: 10px;
      margin-bottom: 20px;
      border: 1px solid #ccc;
      border-radius: 4px;
    }
  </style>
</head>
</html>
```

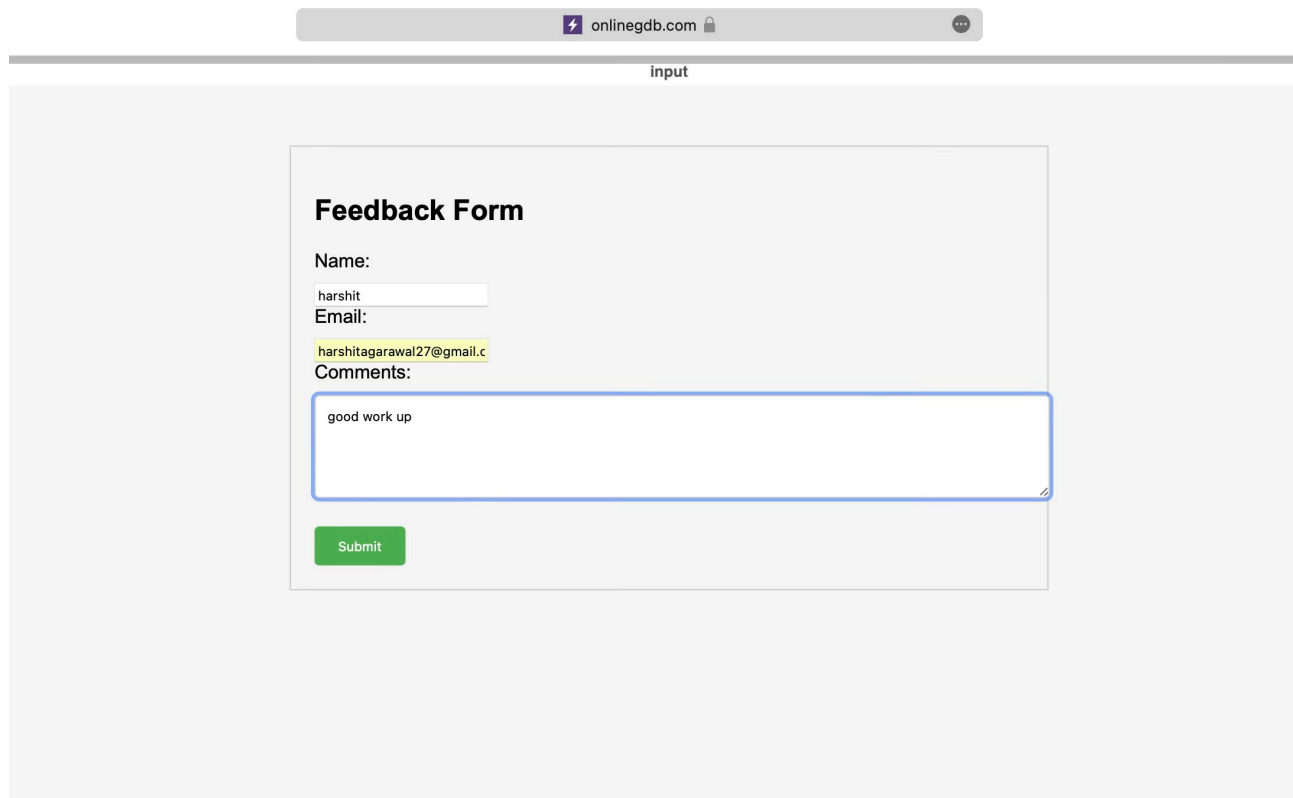
```

        button[type="submit"] {
            background-color: #4CAF50;
            color: white;
            padding: 10px 20px;
            border: none;
            border-radius: 4px;
            cursor: pointer;
        }

        button[type="submit"]:hover {
            background-color: #45a049;
        }
    </style>
</head>
<body>
    <form action="submit_feedback.php" method="post">
        <h2>Feedback Form</h2>
        <label for="name">Name:</label>
        <input type="text" id="name" name="name" required>
        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required>
        <label for="comments">Comments:</label>
        <textarea id="comments" name="comments" rows="5" required></textarea>
        <button type="submit">Submit</button>
    </form>
</body>
</html>

```

Output:



The screenshot shows a web browser window with the address bar displaying 'onlinegdb.com'. The page content is a feedback form titled 'Feedback Form'. The form includes the following elements:

- Name:** A text input field containing the value 'harshit'.
- Email:** An email input field containing the value 'harshitagarawal27@gmail.c'.
- Comments:** A text area containing the text 'good work up'.
- Submit:** A green button labeled 'Submit'.

Experiment 3

A CSS Program to work with background and border properties.

Code:

```
/* Set the background color to light blue */
body {
  background-color: #d3e9f9;
}

/* Add a border to all elements with the class "border-example" */
.border-example {
  border: 2px solid #ccc;
  padding: 10px;
}

/* Add a red border to all elements with the class "red-border-example" */
.red-border-example {
  border: 2px solid red;
  padding: 10px;
}

/* Add a blue border to all elements with the class "blue-border-example" on hover */
.blue-border-example:hover {
  border: 2px solid blue;
  padding: 10px;
}
```

B Java script program to print multiplication table of the given integer.

Code:

```
let num = prompt("Enter an integer: ");

for (let i = 1; i <= 10; i++) {
  console.log(num + " x " + i + " = " + num*i);
}
```

Output

```
node /tmp/e8h21RPbCr.js
Enter an integer: 7
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70
|
```

C Java script program to validate the registration form contents with the following rules

- i. Username Must starts with Uppercase followed by set of lowercase letters or digits.
- ii. Password must contain only uppercase letters and length must be in between 8 to 12.
- iii. Phone number contains 10 digits.
- iv. E-mail must follow some predefined format(example@domain.com)

Code:

```
<!DOCTYPE html>
<html>
<head>
  <title>Registration Form</title>
  <script>
    function validateForm() {
      let username = document.getElementById("username").value;
      let password = document.getElementById("password").value;
      let phone = document.getElementById("phone").value;
      let email = document.getElementById("email").value;

      // Check if username is valid
      let usernameRegex = /^[A-Z][a-z0-9]+$/;
      if (!usernameRegex.test(username)) {
        alert("Username must start with an uppercase letter, followed by
lowercase letters or digits");
        return false;
      }

      // Check if password is valid
      let passwordRegex = /^[A-Z]{8,12}$/;
      if (!passwordRegex.test(password)) {
        alert("Password must contain only uppercase letters and be
between 8 and 12 characters long");
        return false;
      }

      // Check if phone number is valid
      let phoneRegex = /^\d{10}$/;
      if (!phoneRegex.test(phone)) {
        alert("Phone number must contain 10 digits");
        return false;
      }

      // Check if email is valid
      let emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
      if (!emailRegex.test(email)) {
        alert("Email must follow the format example@domain.com");
        return false;
      }
    }
  </script>
</head>
</html>
```

```

        // If all fields are valid, return true
        return true;
    }
</script>
</head>
<body>
    <h2>Registration Form</h2>
    <form onsubmit="return validateForm()">
        <label for="username">Username:</label>
        <input type="text" id="username" name="username" required>
        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required>
        <label for="phone">Phone:</label>
        <input type="tel" id="phone" name="phone" required>
        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required>
        <button type="submit">Submit</button>
    </form>
</body>
</html>

```

Output:

The screenshot shows a web browser window with the address bar displaying 'onlinegdb.com'. The page title is 'Registration Form'. The form contains the following fields and values:

Field	Value
Username:	Harshit
Password:	
Phone:	4206942069
Email:	harshitagarawal27@gmail.c

A 'Submit' button is located to the right of the Email field.

Experiment 4

JDBC Program to perform various DML Operations on the database using Statement.

Code:

```
import java.sql.*;

public class DmlStatementExample {
    public static void main(String[] args) {
        // Database credentials
        String url = "jdbc:mysql://localhost:3306/mydb";
        String user = "root";
        String password = "password";

        // SQL statements
        String insertSql = "INSERT INTO employees (id, name, age, salary) VALUES (1, 'John Doe', 25, 50000)";
        String updateSql = "UPDATE employees SET salary = 60000 WHERE name = 'John Doe'";
        String deleteSql = "DELETE FROM employees WHERE name = 'John Doe'";

        try {
            // Create a connection to the database
            Connection conn = DriverManager.getConnection(url, user, password);

            // Create a statement object
            Statement stmt = conn.createStatement();

            // Insert a record
            int rowsAffected = stmt.executeUpdate(insertSql);
            System.out.println(rowsAffected + " row(s) inserted.");

            // Update a record
            rowsAffected = stmt.executeUpdate(updateSql);
            System.out.println(rowsAffected + " row(s) updated.");

            // Delete a record
            rowsAffected = stmt.executeUpdate(deleteSql);
            System.out.println(rowsAffected + " row(s) deleted.");

            // Close the statement and connection
            stmt.close();
            conn.close();
        } catch (SQLException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

Experiment 6

A JSP Program to print multiplication table.

Code::

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <title>Multiplication Table</title>
  </head>
  <body>
    <h1>Multiplication Table</h1>
    <table>
      <%
        int num = Integer.parseInt(request.getParameter("num"));
        for(int i=1; i<=10; i++){
          out.println("<tr>");
          out.println("<td>" + num + " x " + i + "</td>");
          out.println("<td>="</td>");
          out.println("<td>" + (num * i) + "</td>");
          out.println("</tr>");
        }
      %>
    </table>
  </body>
</html>
```

Output:

B JSP Program to handle the exceptions.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <title>Exception Handling</title>
  </head>
  <body>
    <h1>Exception Handling</h1>
    <%
      try {
        // Code that may throw an exception
        int result = 10 / 0;
        out.println("Result: " + result);
      } catch(Exception e) {
        // Code to handle the exception
        out.println("<p>An error occurred: " + e.getMessage() + "</p>");
      }
    %>
  </body>
```


</html>

Output:

C JSP Program to retrieve the student data from database based on his roll number.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="java.sql.*" %>
<!DOCTYPE html>
<html>
  <head>
    <title>Student Data</title>
  </head>
  <body>
    <h1>Student Data</h1>
    <%
      // Retrieve the roll number parameter
      int rollNum = Integer.parseInt(request.getParameter("rollNum"));

      // Connect to the database
      String url = "jdbc:mysql://localhost:3306/students";
      String username = "root";
      String password = "password";
      Connection conn = DriverManager.getConnection(url, username, password);

      // Retrieve the student data from the database
      PreparedStatement pstmt = conn.prepareStatement("SELECT * FROM students WHERE
roll_num = ?");
      pstmt.setInt(1, rollNum);
      ResultSet rs = pstmt.executeQuery();

      // Output the student data
      if(rs.next()) {
        out.println("<p>Name: " + rs.getString("name") + "</p>");
        out.println("<p>Roll Number: " + rs.getInt("roll_num") + "</p>");
        out.println("<p>Age: " + rs.getInt("age") + "</p>");
      } else {
        out.println("<p>Student with roll number " + rollNum + " not found</p>");
      }

      // Close the database connection
      conn.close();
    %>
  </body>
</html>
```

Experiment 7

A PHP program to work with associative arrays.

Code:

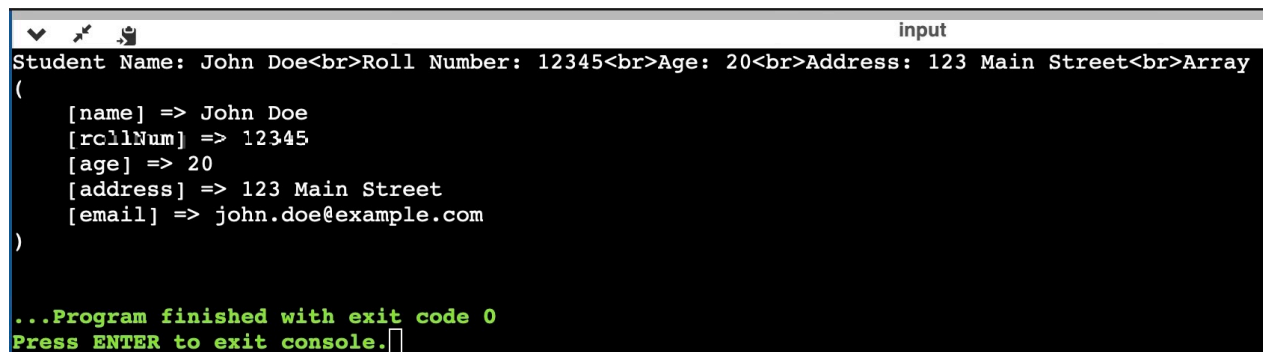
```
<?php
// Define an associative array with key-value pairs
$student = array(
    "name" => "John Doe",
    "rollNum" => 12345,
    "age" => 20,
    "address" => "123 Main Street"
);

// Access and print values using keys
echo "Student Name: " . $student["name"] . "<br>";
echo "Roll Number: " . $student["rollNum"] . "<br>";
echo "Age: " . $student["age"] . "<br>";
echo "Address: " . $student["address"] . "<br>";

// Add a new key-value pair to the array
$student["email"] = "john.doe@example.com";

// Print the updated array
print_r($student);
?>
```

Output:



```
input
Student Name: John Doe<br>Roll Number: 12345<br>Age: 20<br>Address: 123 Main Street<br>Array
(
    [name] => John Doe
    [rollNum] => 12345
    [age] => 20
    [address] => 123 Main Street
    [email] => john.doe@example.com
)

...Program finished with exit code 0
Press ENTER to exit console.
```

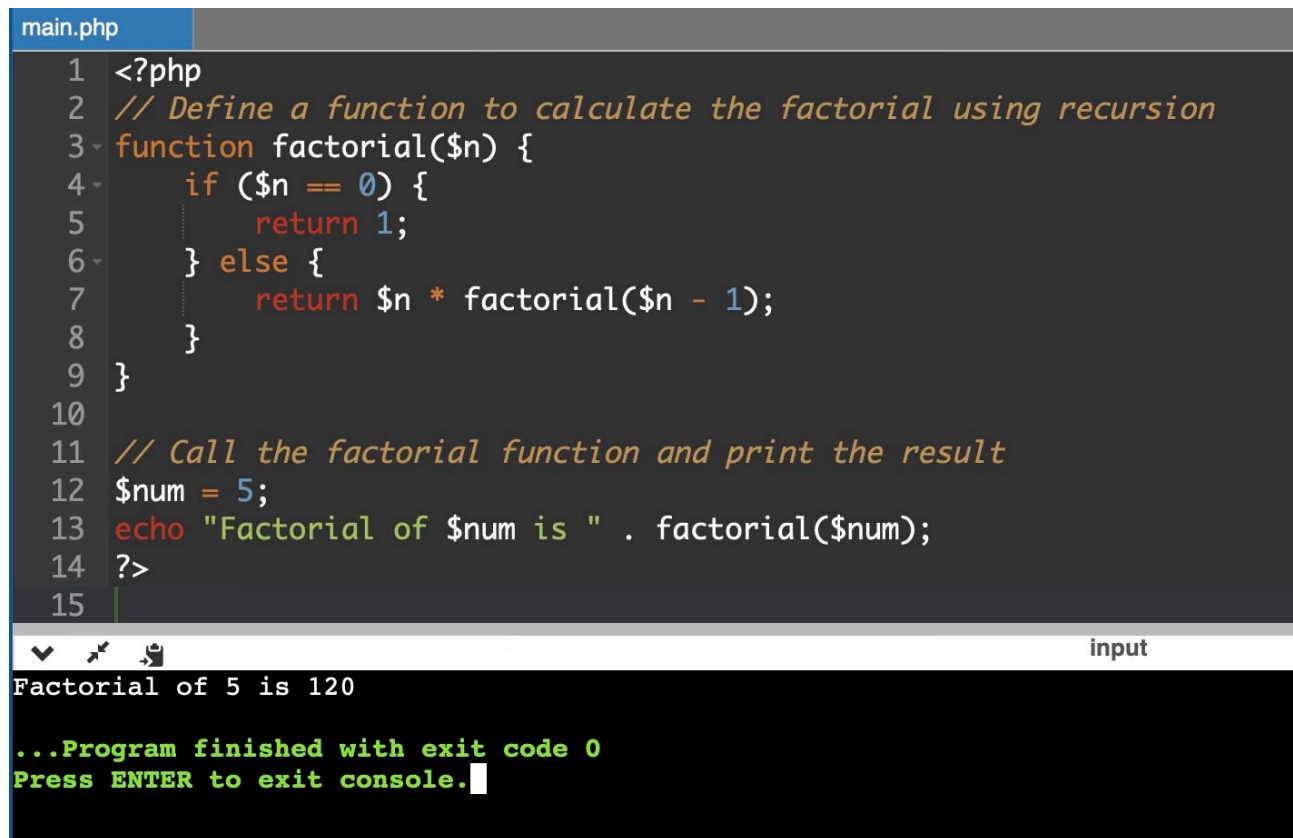
B PHP program to find factorial using Recursion.

Code:

```
<?php
// Define a function to calculate the factorial using recursion
function factorial($n) {
    if ($n == 0) {
        return 1;
    } else {
        return $n * factorial($n - 1);
    }
}

// Call the factorial function and print the result
$num = 5;
echo "Factorial of $num is " . factorial($num);
?>
```

Output:

A screenshot of a terminal window showing the execution of a PHP script. The terminal has a dark background with light-colored text. At the top, the file name 'main.php' is visible in a blue header. The script content is displayed with line numbers from 1 to 15. The output of the script is 'Factorial of 5 is 120'. Below the output, a green message indicates the program finished with exit code 0 and prompts the user to press ENTER to exit the console.

```
main.php
1 <?php
2 // Define a function to calculate the factorial using recursion
3 function factorial($n) {
4     if ($n == 0) {
5         return 1;
6     } else {
7         return $n * factorial($n - 1);
8     }
9 }
10
11 // Call the factorial function and print the result
12 $num = 5;
13 echo "Factorial of $num is " . factorial($num);
14 ?>
15

input
Factorial of 5 is 120

...Program finished with exit code 0
Press ENTER to exit console.
```

C PHP Program to perform various DDL operations on MySQL database.

Code:

```
<?php
// Create a connection to the database
$servername = "localhost";
```

```

$username = "root";
$password = "";
$dbname = "mydb";

$conn = mysqli_connect($servername, $username, $password, $dbname);

// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

// Create a table
$sql = "CREATE TABLE customers (
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
)";

if (mysqli_query($conn, $sql)) {
    echo "Table created successfully";
} else {
    echo "Error creating table: " . mysqli_error($conn);
}

// Alter a table
$sql = "ALTER TABLE customers ADD phone VARCHAR(15)";

if (mysqli_query($conn, $sql)) {
    echo "Table altered successfully";
} else {
    echo "Error altering table: " . mysqli_error($conn);
}

// Drop a table
$sql = "DROP TABLE customers";

if (mysqli_query($conn, $sql)) {
    echo "Table dropped successfully";
} else {
    echo "Error dropping table: " . mysqli_error($conn);
}

// Close the connection
mysqli_close($conn);
?>

```

Experiment 8

Write a JQuery Script to implement hide() and show() effects.

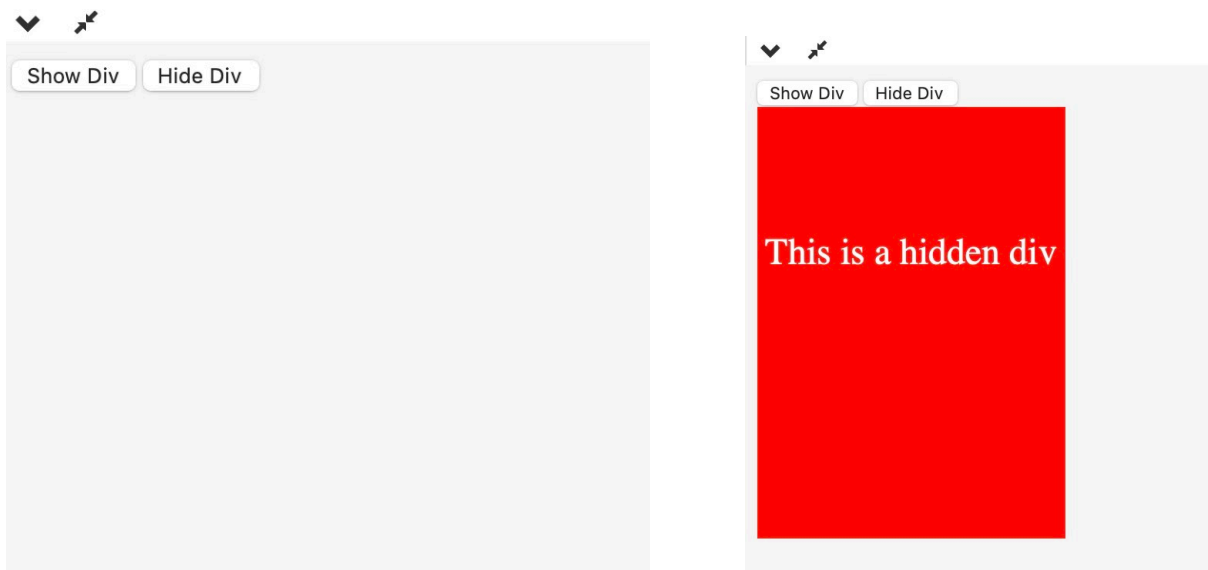
Code:

```
<!DOCTYPE html>
<html>
<head>
  <title>Hide and Show Effects</title>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <style type="text/css">
    #myDiv {
      width: 200px;
      height: 200px;
      background-color: red;
      color: white;
      text-align: center;
      padding-top: 80px;
      font-size: 24px;
      display: none;
    }
  </style>
  <script type="text/javascript">
    $(document).ready(function() {
      // Hide the div on page load
      $("#myDiv").hide();

      // Show the div when the button is clicked
      $("#showBtn").click(function() {
        $("#myDiv").show();
      });

      // Hide the div when the button is clicked
      $("#hideBtn").click(function() {
        $("#myDiv").hide();
      });
    });
  </script>
</head>
<body>
  <button id="showBtn">Show Div</button>
  <button id="hideBtn">Hide Div</button>
  <div id="myDiv">This is a hidden div</div>
</body>
</html>
```

Output:



B Write a JQuery Script to apply various sliding effects.

Code:

```
<!DOCTYPE html>
<html>
<head>
  <title>Sliding Effects with jQuery</title>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <script>
    $(document).ready(function() {
      // Slide down effect
      $("#slide-down-button").click(function() {
        $("#slide-down-div").slideDown();
      });

      // Slide up effect
      $("#slide-up-button").click(function() {
        $("#slide-up-div").slideUp();
      });

      // Slide toggle effect
      $("#slide-toggle-button").click(function() {
        $("#slide-toggle-div").slideToggle();
      });

      // Slide left effect
      $("#slide-left-button").click(function() {
        $("#slide-left-div").animate({
          left: '-=250px'
        }, 1000);
      });

      // Slide right effect
      $("#slide-right-button").click(function() {
```

```

        $("#slide-right-div").animate({
            left: '+=250px'
        }, 1000);
    });
});
</script>
<style>
#slide-down-div,
#slide-up-div,
#slide-toggle-div,
#slide-left-div,
#slide-right-div {
    display: none;
    width: 200px;
    height: 100px;
    background-color: #eee;
    margin-bottom: 10px;
    text-align: center;
    line-height: 100px;
}
</style>
</head>
<body>
<h1>Sliding Effects with jQuery</h1>

<p>Click the buttons to apply various sliding effects:</p>

<button id="slide-down-button">Slide Down</button>
<button id="slide-up-button">Slide Up</button>
<button id="slide-toggle-button">Slide Toggle</button>
<button id="slide-left-button">Slide Left</button>
<button id="slide-right-button">Slide Right</button>

<div id="slide-down-div">Slide Down Effect</div>
<div id="slide-up-div">Slide Up Effect</div>
<div id="slide-toggle-div">Slide Toggle Effect</div>
<div id="slide-left-div">Slide Left Effect</div>
<div id="slide-right-div">Slide Right Effect</div>
</body>
</html>

```

Output:

Sliding Effects with jQuery

Click the buttons to apply various sliding effects:



Experiment 9

Implement shopping cart with Angular JS.

Code:

HTML

```
<div ng-app="shoppingCartApp" ng-controller="shoppingCartCtrl">
  <h2>Shopping Cart</h2>
  <table>
    <tr>
      <th>Product</th>
      <th>Price</th>
      <th>Quantity</th>
      <th>Total</th>
    </tr>
    <tr ng-repeat="item in items">
      <td>{{item.name}}</td>
      <td>{{item.price}}</td>
      <td>
        <button ng-click="decreaseQuantity(item)">-</button>
        {{item.quantity}}
        <button ng-click="increaseQuantity(item)">+</button>
      </td>
      <td>{{item.price * item.quantity}}</td>
    </tr>
    <tr>
      <td colspan="3">Total:</td>
      <td>{{total()}}</td>
    </tr>
  </table>
</div>
```

JAVASCRIPT(Angular JS)

```
var app = angular.module('shoppingCartApp', []);

app.controller('shoppingCartCtrl', function($scope) {
  $scope.items = [
    {name: 'Item 1', price: 10, quantity: 1},
    {name: 'Item 2', price: 20, quantity: 2},
    {name: 'Item 3', price: 30, quantity: 3}
  ];

  $scope.total = function() {
    var total = 0;
    angular.forEach($scope.items, function(item) {
      total += item.price * item.quantity;
    });
    return total;
  };
});
```



```

$scope.increaseQuantity = function(item) {
    item.quantity++;
};

$scope.decreaseQuantity = function(item) {
    if (item.quantity > 1) {
        item.quantity--;
    }
};
});

```

Output:

Shopping Cart

Product	Price	Quantity	Total
{{item.name}}	{{item.price}}	- {{item.quantity}} +	{{item.price * item.quantity}}
Total:			{{total()}}

Write a program to display data in tables in various forms.

HTML:

```

<!DOCTYPE html>
<html>
<head>
    <title>Data Table</title>
    <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
    <h1>Data Table</h1>

    <table>
        <thead>
            <tr>
                <th>Name</th>
                <th>Age</th>
                <th>Gender</th>
                <th>City</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td>John Doe</td>
                <td>30</td>
                <td>Male</td>
                <td>New York</td>
            </tr>
            <tr>
                <td>Jane Doe</td>
                <td>25</td>
            </tr>
        </tbody>
    </table>

```

```

        <td>Female</td>
        <td>Los Angeles</td>
    </tr>
    <tr>
        <td>Bob Smith</td>
        <td>40</td>
        <td>Male</td>
        <td>Chicago</td>
    </tr>
    <tr>
        <td>Sara Lee</td>
        <td>35</td>
        <td>Female</td>
        <td>Miami</td>
    </tr>
    <tr>
        <td>Tom Jones</td>
        <td>50</td>
        <td>Male</td>
        <td>San Francisco</td>
    </tr>
</tbody>
</table>

</body>
</html>

```

CSS:

```

body {
    font-family: Arial, sans-serif;
    background-color: #f2f2f2;
}

h1 {
    text-align: center;
    color: #333;
    margin-top: 50px;
    margin-bottom: 50px;
}

table {
    border-collapse: collapse;
    margin: 0 auto;
    background-color: #fff;
    box-shadow: 0px 0px 10px #888888;
}

table th {
    padding: 10px;
    background-color: #666;
    color: #fff;
    font-weight: bold;
    text-align: left;
}

table td {
    padding: 10px;
    border: 1px solid #ddd;
}


```

```
}

table tr:nth-child(even) {
    background-color: #f2f2f2;
}

table tr:hover {
    background-color: #ccc;
}
```

Output:

 >

Data Table			
Name	Age	Gender	City
John Doe	30	Male	New York
Jane Doe	25	Female	Los Angeles
Bob Smith	40	Male	Chicago
Sara Lee	35	Female	Miami
Tom Jones	50	Male	San Francisco

Experiment 10

Explain the step by step process how to create database & collection in MongoDB. Create a collection with student name and store 5 rows in that collection.

Here are the step by step process to create a database and collection in MongoDB:

- Install MongoDB: Firstly, download and install MongoDB on your system. You can follow the official documentation to do the same.
- Start MongoDB server: After installation, start the MongoDB server by running the following command on your terminal: `mongod`
- Open MongoDB shell: Open another terminal and run the following command to open MongoDB shell: `mongo`. This will open up the MongoDB shell and you can execute commands from here.
- Create a database: To create a new database, use the `USE` command followed by the name of the database you want to create. For example, to create a database named "school", run the following command: `use school`
- Create a collection: To create a collection in the database, use the `db.createCollection()` command followed by the name of the collection you want to create. For example, to create a collection named "students", run the following command:
`db.createCollection("students")`
- Insert data into the collection: To insert data into the collection, use the `db.collectionName.insert()` command followed by the data you want to insert. For example, to insert 5 rows of data into the "students" collection, run the following command:

```
db.students.insert([
  { name: "John Doe", age: 25, gender: "Male", course: "Engineering" },
  { name: "Jane Smith", age: 22, gender: "Female", course: "Medical" },
  { name: "Bob Johnson", age: 27, gender: "Male", course: "Law" },
  { name: "Emily Williams", age: 20, gender: "Female", course: "Arts" },
  { name: "Mike Brown", age: 23, gender: "Male", course: "Business" }
])
```

Experiment 5

Servlet program to read the parameters from user interface and display welcome message.

Code:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class WelcomeServlet extends HttpServlet {

    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html");

        PrintWriter out = response.getWriter();

        // Read the parameters from the HTML form
        String firstName = request.getParameter("firstName");
        String lastName = request.getParameter("lastName");

        // Display a welcome message
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Welcome " + firstName + " " + lastName + "</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Welcome " + firstName + " " + lastName + "!</h1>");
        out.println("</body>");
        out.println("</html>");
    }
}
```

Servlet program to work with HttpSession Object.

Code:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class SessionServlet extends HttpServlet {

    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html");

        PrintWriter out = response.getWriter();
```

```

// Get the HttpSession object
HttpSession session = request.getSession(true);

// Get the value of the "name" attribute from the session
String name = (String) session.getAttribute("name");

// Check if the name attribute is present in the session
if (name != null) {
    out.println("<html>");
    out.println("<head>");
    out.println("<title>Welcome back " + name + "</title>");
    out.println("</head>");
    out.println("<body>");
    out.println("<h1>Welcome back " + name + "!</h1>");
    out.println("</body>");
    out.println("</html>");
} else {
    // Get the value of the "firstName" parameter from the request
    String firstName = request.getParameter("firstName");

    // Set the "name" attribute in the session
    session.setAttribute("name", firstName);

    out.println("<html>");
    out.println("<head>");
    out.println("<title>Welcome " + firstName + "</title>");
    out.println("</head>");
    out.println("<body>");
    out.println("<h1>Welcome " + firstName + "!</h1>");
    out.println("</body>");
    out.println("</html>");
}
}
}

```

Demo on Generating dynamic content with form processing using Servlets.

Code:

```

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class FormServlet extends HttpServlet {

    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html");

        PrintWriter out = response.getWriter();

        // Get the values of the form parameters
        String firstName = request.getParameter("firstName");
        String lastName = request.getParameter("lastName");
    }
}

```

```

String gender = request.getParameter("gender");
String[] hobbies = request.getParameterValues("hobbies");

// Generate dynamic content based on the form parameters
out.println("<html>");
out.println("<head>");
out.println("<title>Form Processing Result</title>");
out.println("</head>");
out.println("<body>");
out.println("<h1>Form Processing Result</h1>");
out.println("<p>Your name is " + firstName + " " + lastName + ".</p>");
out.println("<p>Your gender is " + gender + ".</p>");

if (hobbies != null && hobbies.length > 0) {
    out.println("<p>Your hobbies are:</p>");
    out.println("<ul>");
    for (String hobby : hobbies) {
        out.println("<li>" + hobby + "</li>");
    }
    out.println("</ul>");
}

out.println("</body>");
out.println("</html>");
}
}

```