**Name**: Aarya Tiwari  **RollNo.:** 16010421119

**Form Validation**

**Code:**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>Registration Form</title>**

**<style>**

**body {**

**font-family: Arial, sans-serif;**

**}**

**h1 {**

**color: #333;**

**text-align: center;**

**}**

**form {**

**max-width: 400px;**

**margin: 0 auto;**

**}**

**label {**

**display:inline-block;**

**margin-top: 25px;**

**font-weight: bold;**

**}**

**input[type="text"],**

**input[type="email"] {**

**width: 100%;**

**padding: 8px;**

**margin-bottom: 10px;**

**border: 1px solid #ccc;**

**border-radius: 4px;**

**}**

**input[type="radio"],**

**input[type="checkbox"] {**

**margin-right: 5px;**

**margin-bottom: 10px;**

**}**

**input[type="submit"] {**

**background-color: #4CAF50;**

**color: white;**

**padding: 10px 20px;**

**border: none;**

**border-radius: 4px;**

**cursor: pointer;**

**margin-top: 25px;**

**}**

**.error {**

**color: red;**

**margin-bottom: 10px;**

**}**

**table {**

**width: 100%;**

**border-collapse: collapse;**

**margin-top: 20px;**

**}**

**th, td {**

**padding: 15px;**

**border-bottom: 1px solid #ddd;**

**/\* display : flex; \*/**

**}**

**th {**

**background-color: #f2f2f2;**

**}**

**</style>**

**</head>**

**<body>**

**<h1>Registration Form</h1>**

**<form action="<?php echo $\_SERVER["PHP\_SELF"]; ?>" method="POST">**

**<label for="name">Name:</label>**

**<input type="text" id="name" name="name" ><br>**

**<label for="email">Email:</label>**

**<input type="text" id="email" name="email" ><br>**

**<label for="gender">Gender:</label>**

**<input type="radio" id="male" name="gender" value="male" >**

**<label for="male">Male</label>**

**<input type="radio" id="female" name="gender" value="female" >**

**<label for="female">Female</label><br>**

**<label for="hobbies">Hobbies:</label>**

**<input type="checkbox" id="reading" name="hobbies[]" value="reading">**

**<label for="reading">Reading</label>**

**<input type="checkbox" id="sports" name="hobbies[]" value="sports">**

**<label for="sports">Sports</label>**

**<input type="checkbox" id="music" name="hobbies[]" value="music">**

**<label for="music">Music</label><br>**

**<input type="submit" value="Submit">**

**</form>**

**</body>**

**</html>**

**<?php**

**$name = $email = $gender = " ";**

**if (empty($\_POST['name'])) {**

**$errors[] = "Name is required";**

**} else {**

**// echo nl2br($name ."\n ");**

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

**// echo nl2br($name ."\n ");**

**$name = trim($name);**

**// echo nl2br($name."\n ");**

**$name = stripslashes($name);**

**// echo nl2br($name."\n ");**

**$name = htmlspecialchars($name);**

**// echo nl2br($name."\n ");**

**}**

**if (empty($\_POST['email'])) {**

**$errors[] = "Email is required";**

**} elseif (!filter\_var($\_POST['email'], FILTER\_VALIDATE\_EMAIL)) {**

**$errors[] = "Invalid email format";**

**} else {**

**$email = $\_POST['email'];**

**}**

**if (empty($\_POST['gender'])) {**

**$errors[] = "Gender is required";**

**} else {**

**$gender = $\_POST['gender'];**

**}**

**if (empty($\_POST['hobbies'])) {**

**$errors[] = "At least one hobby must be selected";**

**} else {**

**$hobbies = $\_POST['hobbies'];**

**}**

**if (!empty($errors)) {**

**foreach ($errors as $error) {**

**echo $error . "<br>";**

**}**

**} else {?>**

**<table>**

**<th>Name</th>**

**<th>email</th>**

**<th>Gender</th>**

**<th>Hobbies</th>**

**<tr>**

**<td class="center"> <?php echo $name ?> </td>**

**<td class="center"> <?php echo $email ?> </td>**

**<td class="center"> <?php echo $gender ?> </td>**

**<td class="center"> <?php print\_r ($hobbies) ?> </td>**

**</tr>**

**</table>**

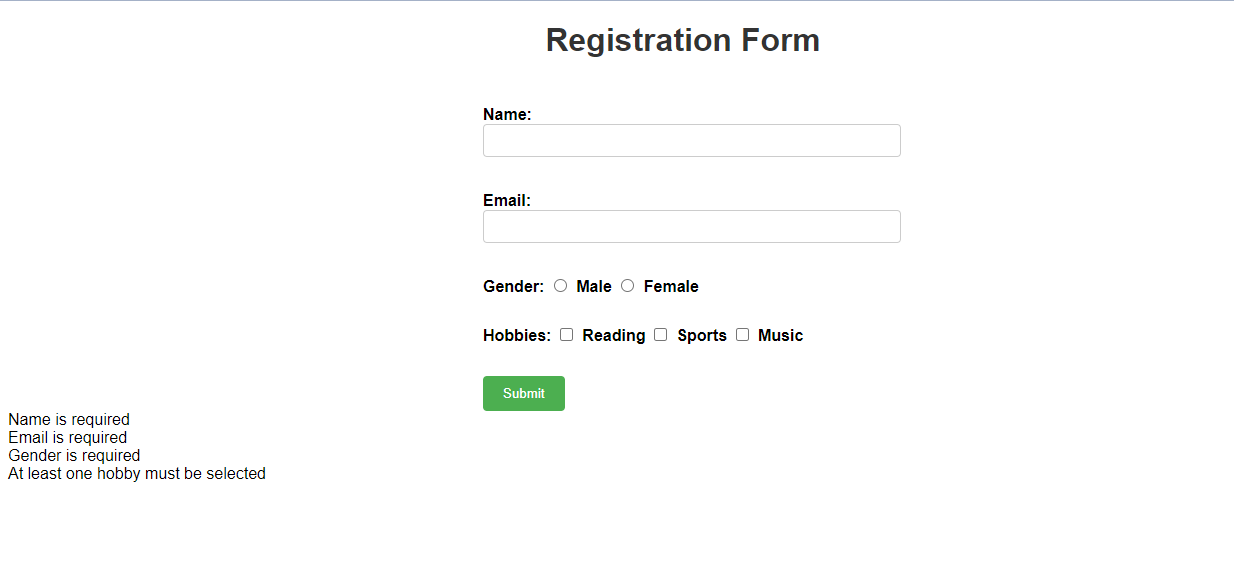
**<?php**

**// Display a success message or redirect to another page**

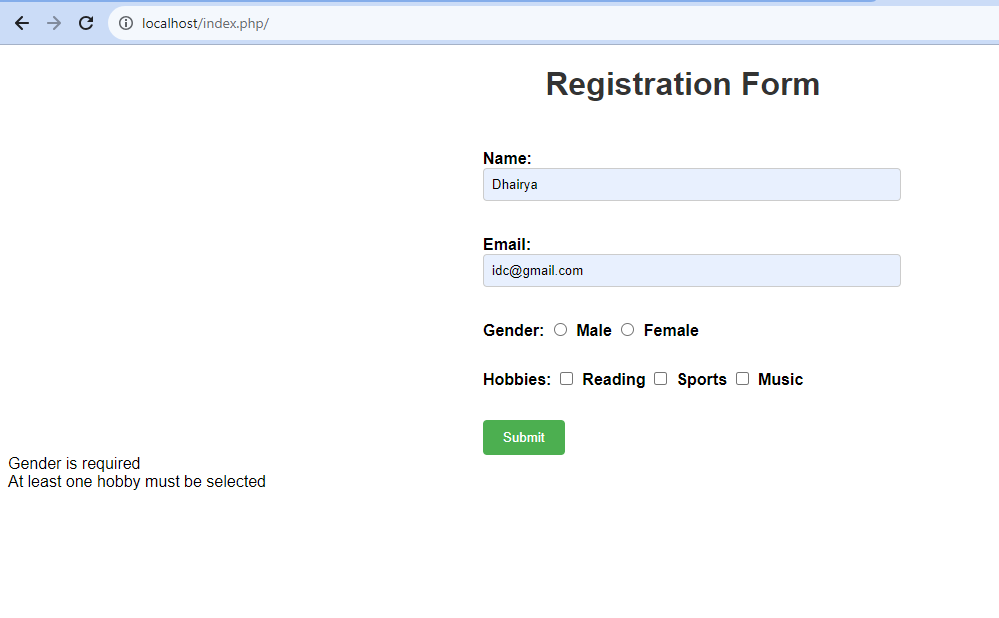
**}**

**?>**

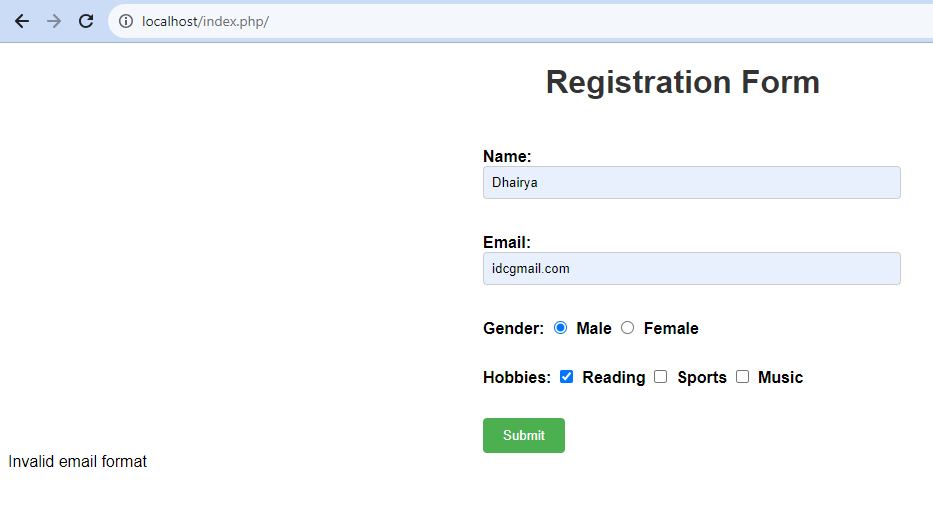
**0 fields input:**

****

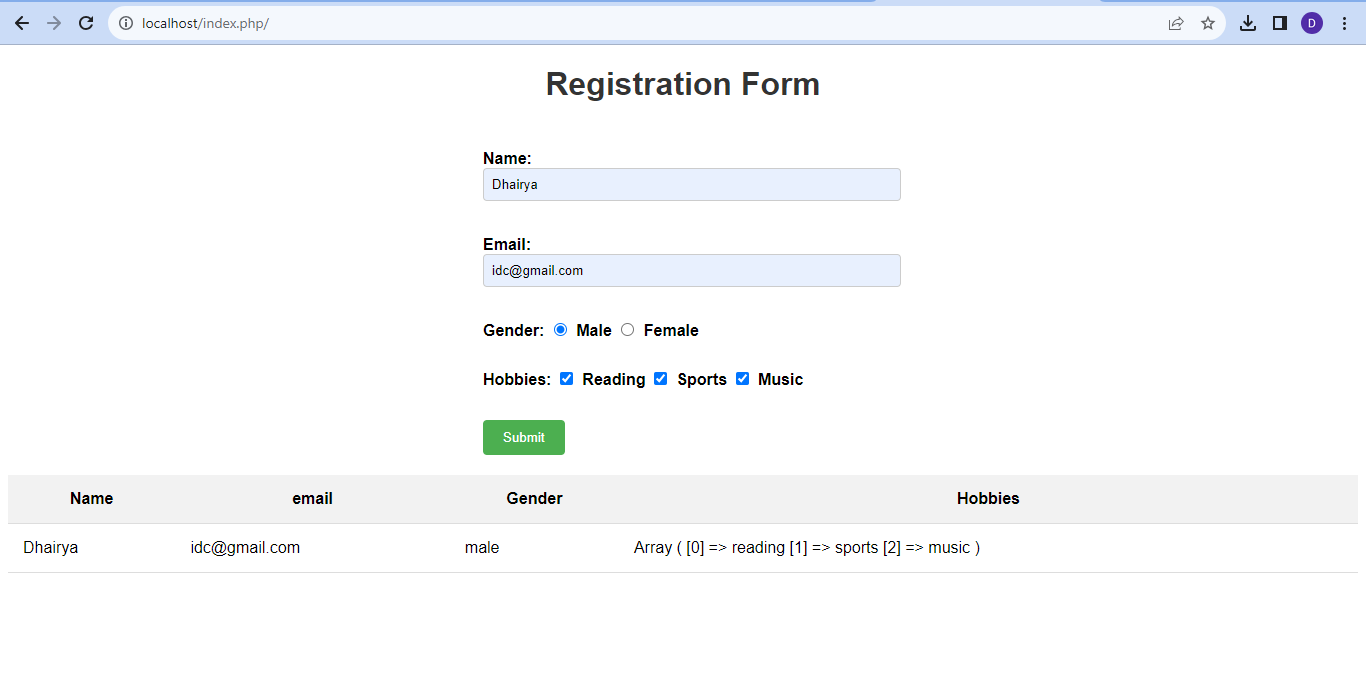
**Incomplete Input**

****

**Invalid Input:**

****

**Correct input:**

****

**Post Lab Questions:**

1. **What is the difference between GET and POST method?**

**Ans:**

|  |  |
| --- | --- |
| HTTP GET | HTTP POST |
| In GET method we can not send large amount of data rather limited data is sent because the request parameter is appended into the URL. | In POST method large amount of data can be sent because the request parameter is appended into the body. |
| GET request is comparatively less secure because the data is exposed in the URL bar. | POST request is comparatively more secure because the data is not exposed in the URL bar. |
| Request made through GET method are stored in Browser history. | Request made through POST method is not stored in Browser history. |
| GET method request can be saved as bookmark in browser. | POST method request can not be saved as bookmark in browser. |

1. **Explain the importance of $\_Server Method?**

**Ans:** $\_SERVER allows you to access a wide range of server-related information such as server name, server software, server IP address, server port, and more. This information can be used for debugging, logging, or customizing behavior based on the server environment. It provides details about the current HTTP request, including the request method (GET, POST, etc.), the requested URL, the query string, and more. This information is crucial for routing requests, processing form submissions, and building dynamic content. $\_SERVER information can be used for security purposes, such as preventing unauthorized access, validating user agents, or blocking specific IP addresses.

1. **How Form validation can help to prevent crosssite scripting attack?**

**Ans:** Form validation involves checking user inputs for potentially malicious content. It's important to sanitize input data by removing or escaping characters that could be interpreted as code. This prevents attackers from injecting malicious scripts that could be executed by other users' browsers. Form validation helps ensure that only valid data is accepted by the application. This can prevent attackers from submitting inputs that contain malicious scripts or other unauthorized content. Validating input against a predefined set of rules or patterns helps ensure that only safe and expected inputs are accepted. Any inputs that don't match these rules can be rejected.

**Outcomes:**

CO1: Illustrate use of basic PHP concepts to develop applications

CO2: Design forms and use session handling mechanism with web applications  

**Conclusion:**

In this experiment we understood form handling using GET and POST method and also validated the form using php.