**PART A**

(Part A: TO BE REFFERED BY STUDENTS)

**Experiment No. 08**

**A.1 AIM:**

Registration Forms using HTML and PHP

**A.2 Pre requisite:**

Basic Knowledge of HTML and CSS tags and PHP

**A.3 Outcome:**

After successful completion of this experiment students will be able to:

1. Design pages accepting user inputs in HTML
2. Design and study various elements of forms.
3. Study the difference between the GET and POST methods of a form.
4. Design and study various methods of PHP.

**A.4 Theory:**

HTML forms are used to pass data to a server.

An HTML form can contain input elements like text fields, checkboxes, radio-buttons, submit buttons and more. A form can also contain select lists, textarea, fieldset, legend, and label elements.

The <form> tag is used to create an HTML form

**HTML Forms - The Input Element**

The most important form element is the <input> element. The <input> element is used to select user information. An <input> element can vary in many ways, depending on the type attribute. An <input> element can be of type text field, checkbox, password, radio button, submit button, and more.

The most common input types are described below.

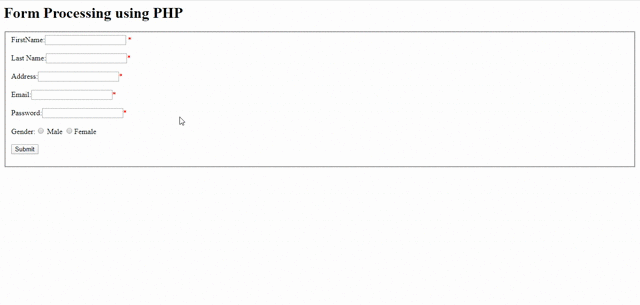
**Controls used in forms**: Form processing contains a set of controls through which the client and server can communicate and share information. The controls used in forms are:

* **Textbox**: Textbox allows the user to provide single-line input, which can be used for getting values such as names, search menu and etc.
* **Textarea**: Textarea allows the user to provide multi-line input, which can be used for getting values such as an address, message etc.
* **DropDown**: Dropdown or combobox allows the user to provide select a value from a list of values.
* **Radio** **Buttons**: Radio buttons allow the user to select only one option from the given set of options.
* **CheckBox**: Checkbox allows the user to select multiple options from the set of given options.
* **Buttons**: Buttons are the clickable controls that can be used to submit the form.

**Form Validation**: Form validation is done to ensure that the user has provided the relevant information. Basic validation can be done using HTML elements.

PHP methods and arrays used in form processing are:

* **isset**(): This function is used to determine whether the variable or a form control is having a value or not.
* **$\_GET[**]: It is used the retrieve the information from the form control through the parameters sent in the URL. It takes the attribute given in the url as the parameter.
* **$\_POST**[]: It is used the retrieve the information from the form control through the HTTP POST method. IT takes name attribute of corresponding form control as the parameter.
* **$**\_**REQUEST**[]: It is used to retrieve an information while using a database.



**A.5 Procedure/Task:**

1. Create a Registration page.

2. First page should accept all personal details from the user. Use all the form controls

1. Textbox
2. Textarea
3. DropDown
4. Radio Buttons
5. CheckBox
6. Buttons

3. clicking on submit button must open next page displays the details from user

4. Do the validation using JavaScipt

5. on clicking log in button a message should be displayed – “Registered Successfully”

6. Use HTML formatting tags to make your pages look attractive

7. Prepare the document. Save and close the file and name it as **EXP08\_Roll no\_Batch no.**

**Sample code:-**

<?php

if (isset($\_POST['submit']))

{

    if ((!isset($\_POST['firstname'])) || (!isset($\_POST['lastname'])) ||

        (!isset($\_POST['address'])) || (!isset($\_POST['emailaddress'])) ||

        (!isset($\_POST['password'])) || (!isset($\_POST['gender'])))

    {

        $error = "\*" . "Please fill all the required fields";

    }

 else

    {

        $firstname = $\_POST['firstname'];

………

**PART B**

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)

|  |  |
| --- | --- |
| Roll No.: N049 | Name: Tarun Tanmay |
| Class: MBATech CE | Batch: B3 |
| Date of Experiment: 17-10-2020 | Date/Time of Submission: 25-10-2020 |
| Grade: |  |

**B.1 Code:**

**PHP Code**

<html>

<body>

<?php

$username = $\_POST['n'];

$lastname = $\_POST['sn'];

$dateofbirth = $\_POST['dob'];

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

$mobile = $\_POST['mno'];

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

$state = $\_POST['c'];

echo 'User details are:'. '</br>';

echo '<i>First Name:</i>'. $username . '</br>';

echo '<i>Last Name:</i>'. $lastname . '</br>';

echo '<i>Date of Birth:</i>'. $dateofbirth . '</br>';

echo '<i>EMail ID:</i>'. $email . '</br>';

echo '<i>Mobile No.:</i>'. $mobile . '</br>';

echo '<i>Gender:</i>'. $gender . '</br>';

echo '<i>State:</i>'. $state . '</br>';

echo '<i>Languages:</i>';

if(isset($\_POST['submit'])){

if(!empty($\_POST['language']))

{

// Loop to store and display values of individual checked checkbox.

foreach($\_POST['language'] as $selected){

echo $selected."</br>";

}

}

}

?>

</body>

</html>

**HTML Registration form:**

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Form</title>

<script>

function validate()

{

var name = document.ContactForm.Name;

var email = document.ContactForm.mail;

var pass = document.ContactForm.password;

var pass1 = document.ContactForm.password1;

var chkbx = document.ContactForm.spam;

var mob = document.ContactForm.mbno;

var des = document.ContactForm.dessert;

if (name.value == "")

{

window.alert("Please enter your full name.");

name.focus();

return false;

}

if (email.value=='')

{

alert('Please enter your Email ID.');

email.focus();

return false;

}

if (email.value.indexOf("@", 0) < 0 || email.value.indexOf('.',0)<0 )

{

alert('Please enter a valid Email ID.')

email.focus();

return false;

}

if (pass.value=='')

{

alert('Enter Password');

pass.focus();

return false;

}

if (pass.value.length<8)

{

alert('Password must have atleast 8 characters.')

pass.focus();

return false;

}

if (pass1.value!=pass.value)

{

alert('Passwords dont match')

pass1.focus();

return false;

}

if ((chkbx.checked==true) && (mob.value==''))

{

alert('Please enter your mobile number.')

mob.focus();

return false;

}

if (mob.value.lenght<10)

{

alert('Please enter a valid number.')

mob.focus();

return false;

}

if (des.selectedIndex<1)

{

alert('Please select the most prefered item.')

des.focus();

return false;

}

}

function EnableDisable(chkbx)

{

if(chkbx.checked == false)

{

document.ContactForm.mbno.disabled = true;

document.ContactForm.mbno.placeholder='No need to enter number';

}

else

{

document.ContactForm.mbno.disabled = false;

document.ContactForm.mbno.placeholder='Mobile Number here';

}

}

</script>

</head>

<body>

<header style="background-color: lightgray; height: 100px; width: 100%">

<h1 style="font-size: 30px; color: black;text-align: center; padding-top: 15px;">CIAO</h1>

</header>

<center>

<div class="container col-md-8 col-md-offset-3" style="overflow: auto">

<h2 style="font-size: 23px;text-align: center;">Signup Now</h2>

<form method="POST" name="ContactForm" onsubmit="return validate();" >

<body style="background-color:lightblue">

<div>

<label>Name</label><br> <input type="text" name="name" placeholder="john doe"required><br>

</div><div>

<label>Number</label><br> <input type="number" name="number" placeholder="0123456789"><br>

</div><div>

<label>Email</label> <br><input type="email" name="email" placeholder="johndoe@gmail.com"><br>

</div><div>

<label>DOB</label> <br><input type="date" name="number"><br>

</div><div>

<label>Address</label><br>

<textarea name="" rows="4" col = "20"></textarea><br>

</div><div>

<label>Gender</label><br>

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

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

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

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

</div><div>

<label>Which mode of transport do you use?</label><br>

<input type="checkbox" id="vehicle1" name="vehicle1" value="Car">

<label for="vehicle1">Car</label><br>

<input type="checkbox" id="vehicle2" name="vehicle2" value="Bus">

<label for="vehicle2">Bus</label><br>

<input type="checkbox" id="vehicle3" name="vehicle3" value="Train">

<label for="vehicle3">Train</label><br>

</div><div>

<label>Password</label><br> <input type="password" name="password"><br>

</div>

<button type="submit" class="btn btn-primary">Submit</button>

</form>

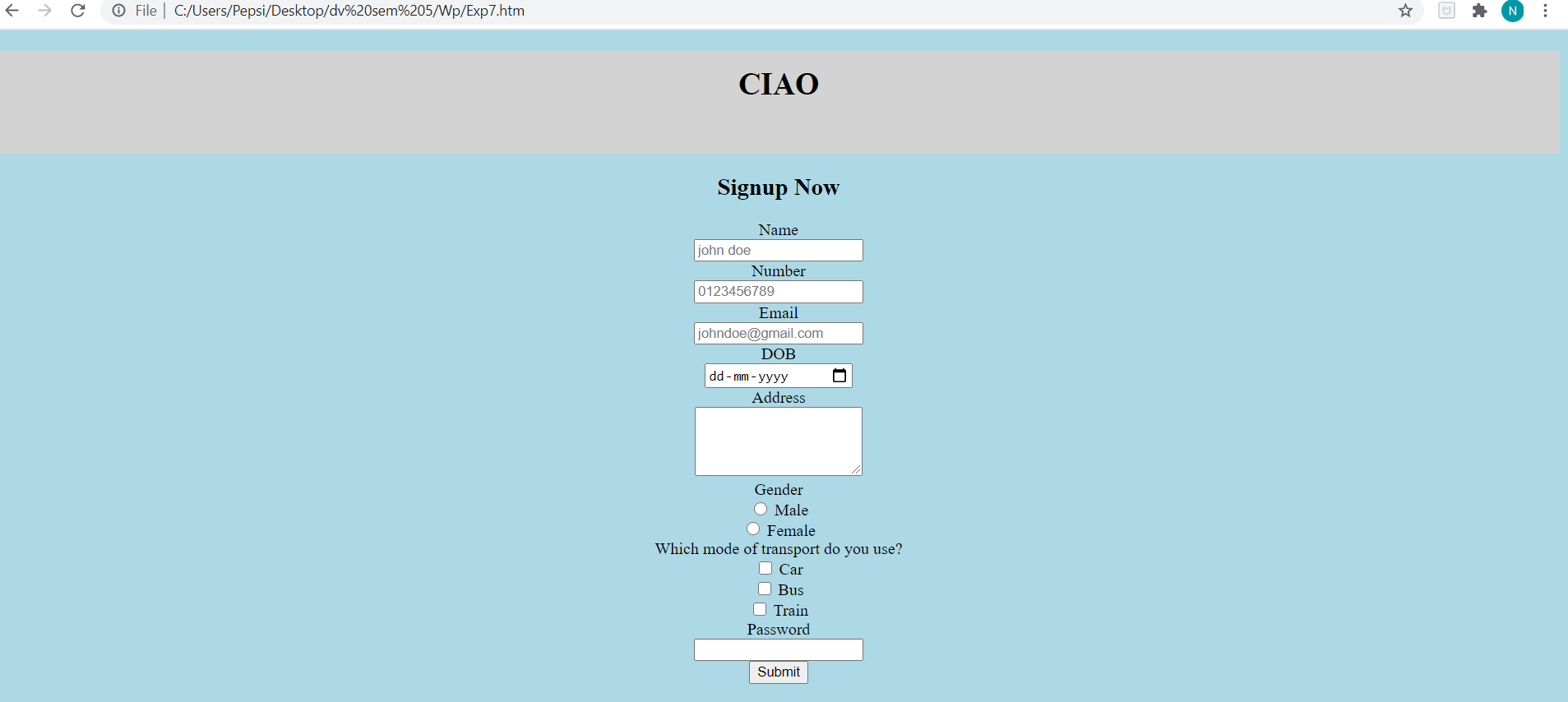
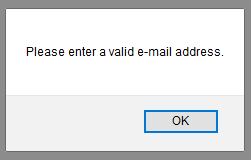
</center>

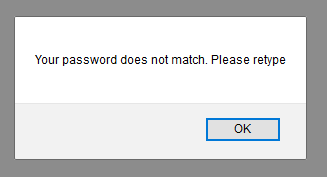
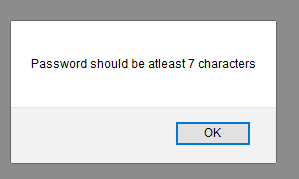
</div>

</body>

</html>

**B.2 Output:**

****

****

**B.3 Conclusion:**

We successfully designed registration page to accept user inputs in HTML using various elements of forms.

**B.3 Observations and Learning:**

We can use GET and POST methods to retrieve user input data and display it on the website from various input fields: required and optional text fields, radio buttons, and a submit button.

$\_GET is an array of variables passed to the current script via the URL parameters. While, $\_POST is an array of variables passed to the current script via the HTTP POST method.

Information sent from a form with the GET method is visible to everyone (all variable names and values are displayed in the URL). GET also has limits on the amount of information to send. We should never use GET when sending passwords or other sensitive information.

**B.4 Question of Curiosity**

**Q1. Difference between get and post method? Answer with appropriate reason.**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **GET** | **POST** |
| **1** | GET is used to request data from a specified resource. | POST is used to send data to a server to create/update a resource. |
| **2** | the query string (name/value pairs) is sent in the URL of a GET request:  /test/demo\_form.php?name1=  value1&name2=value | The data sent to the server with POST is stored in the request body of the HTTP request:  POST /test/demo\_form.php HTTP/1.1 Host: w3schools.com name1=value1&name2=value2 |
| **3** | GET requests can be cached | POST requests are never cached |
| **4** | GET requests remain in the browser history | POST requests do not remain in the browser history |
| **5** | GET requests can be bookmarked | POST requests cannot be bookmarked |
| **6** | GET requests have length restrictions | POST requests have no restrictions on data length |
| **7** | GET requests are only used to request data (not modify) | POST requests can be used to modify data |
| **8** | Harmless | Data will be re-submitted (the browser should alert the user that the data are about to be re-submitted) |
| **9** | Encoding Type:  application/x-www-form-urlencoded | Encoding Type:  application/x-www-form-urlencoded or multipart/form-data. Use multipart encoding for binary data |
| **10** | Only ASCII characters allowed | No restrictions. Binary data is also allowed |
| **11** | GET is less secure compared to POST because data sent is part of the URL  We should never use GET when sending passwords or other sensitive information. | POST is a little safer than GET because the parameters are not stored in browser history or in web server logs |
| **12** | Data is visible to everyone in the URL | Data is not displayed in the URL |

Q2. What are the various methods used for PHP form tag?

Ans. Forms are used to get input from the user and submit it to the web server for processing. It contains graphical user interface items such as input box, check boxes radio buttons etc. Form submission type can use POST or GET method

### **PHP POST method**

* This is the built in PHP super global array variable that is used to get values submitted via HTTP POST method. The array variable can be accessed from any script in the program; it has a global scope. This method is ideal when you do not want to display the form post values in the URL. A good example of using post method is when submitting login details to the server. It has the following syntax.

<?php

$\_POST['variable\_name'];

?>

* “$\_POST[…]” is the PHP array. “'variable\_name'” is the URL variable name.

### **PHP GET method**

* This is the built in PHP super global array variable that is used to get values submitted via HTTP GET method. The array variable can be accessed from any script in the program; it has a global scope. This method displays the form values in the URL. It’s ideal for search engine forms as it allows the users to book mark the results. It has the following syntax.

<?php

$\_GET['variable\_name'];

?>

* “$\_GET[…]” is the PHP array. “'variable\_name'” is the URL variable name.