

ASSIGNMENT – 3

REPORT

Submitted To:

Indraprastha Adhyayan Kendra



Submitted By:

Kanika Bagri

Email: kanikabagri111@gmail.com

Contact: +91 8920565888

INTRODUCTION

The Assignment 3 includes the concepts of PHP to be applied for Form Validation. It also includes submitting the form so that our entries are saved to the Database.

FORM VALIDATION

For validating a form using PHP, ***“name”*** attribute is to be assigned to every field of the form. This ***“name”*** attribute is used to access the value written in every field.

The concept that ***“When the Submit Button is Clicked, then the form will be validated as whether the user has given all the information before submitting the form. If the user leaves any field empty or violate the rules of the field, An ERROR MESSAGE will be shown.”***

```
if(isset($_POST['submit'])) {  
    $fname = $_POST['Fname'];  
    $lname = $_POST['Lname'];  
    $email = $_POST['Email'];  
    $password = $_POST['Password'];  
    $confirm = $_POST['Conpassword'];  
    $gender = $_POST['Gender'];  
    $quali = $_POST['quali'];  
    $address = $_POST['Address'];  
    $contact = $_POST['Contact'];  
    $date = $_POST['Date'];  
    $month = $_POST['Month'];  
    $year = $_POST['Year'];  
    $file = $_POST['Myfile'];  
    $agree = $_POST['Agree'];  
}
```

The first field includes the **Name Field**. The field cannot be left empty and must only include Letters and no Numbers.

```
<form name="myForm" method="POST" class="main_div">
  <table class="table_div">
    <tr>
      <th>First Name:</th>
      <td><input type="text" placeholder="First Name" name="Fname">
      <span class="error">*</span></td>
      <td class="error"><?php echo $error_fname;  ?></td>
    </tr>
    <tr></tr>
    <tr>
      <th>Last Name:</th>
      <td><input type="text" placeholder="Last Name" name="Lname">
      <span class="error">*</span></td>
      <td class="error"><?php echo $error_lname;  ?></td>
    </tr>
    <tr></tr>
  </table>
</form>
```

```
if(empty($fname)) {
  $error_fname = "First Name is Required!";
}
if (!preg_match("/^[a-zA-Z ]*$/",$fname)) {
  $error_fname = "Only Letters are Allowed!";
}

if(empty($lname)) {
  $error_lname = "Last Name is Required!";
}
if (!preg_match("/^[a-zA-Z ]*$/",$lname)) {
  $error_lname = "Only Letters are Allowed!";
}
```

First Name:

2334

*

Only Letters are
Allowed!

Last Name:

RR455

*

Only Letters are
Allowed!

Figure 1: Name Validation

The second field includes **Email Field**. Here, the data has to be filled correctly and that too in a proper format, else an ERROR MESSAGE will be displayed.

```
<tr>
  <th>Email:</th>
  <td><input type="email" placeholder="Email Id" name="Email">
  <span class="error">*</span></td>
  <td class="error"><?php echo $error_email;  ?></td>
</tr>
<tr></tr>
```

```
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
    $error_email = "Provide a Valid Email!";
}
```

Figure 2: Email Validation

The third field is the **Password & Confirm Password** field. Here, fields cannot be empty and the **Password** and **Confirm Password** Fields must match, else ERROR will be displayed.

```
<tr>
  <th>Password:</th>
  <td>
    <input type="password" placeholder="Password" name="Password">
    <span class="error">*</span>
  </td>
  <td class="error"><?php echo $password_error;  ?></td>
</tr>
<tr></tr>
<tr>
  <th>Confirm Password:</th>
  <td>
    <input type="password" placeholder="Confirm Password" name="Conpassword">
    <span class="error">*</span>
  </td>
  <td class="error"><?php echo $confirm_error;  ?></td>
</tr>
<tr></tr>
```

```
if(empty($password)) {
    $password_error = "Password is Required!";
}
if(empty($confirm)) {
    $confirm_error = "Confirm your Password!";
}
if($password !== $confirm) {
    $confirm_error = "Passwords do not Match!";
}
```


The next field is a “**Contact**” field the user cannot enter more than 10 numbers, else ERROR MESSAGE will be displayed.

```
<tr>
  <th>Contact Number:</th>
  <td><input type="number" name="Contact">
    <span class="error">*</span></td>
  <td class="error"><?php echo $error_contact; ?></td>
</tr>
<tr></tr>
```

```
if(empty($contact)) {  
    $error_contact = "Contact is Required!";  
}  
if(strlen($contact) < 10) {  
    $error_contact = "Contact must include 10 Digits!";  
}  
if(strlen($contact) > 10) {  
    $error_contact = "Contact must include 10 Digits!";  
}
```

Contact Number:

98745

Contact must
include 10 Digits!

Figure 6: Contact Validation

The next field is the **“Qualifications”** Field. It is basically an array as multiple data can be passed or checked.

[illegible]

```

}
if(empty($quali)) {
    $error_quali = "Give your Qualification!";
}

```

The next field is **“Upload Photo”** field. The Filed cannot be left Empty, else will show ERRORS.

```

<tr>
    <th>Upload your Photo:</th>
    <td><input type="file" name="Myfile">
    <span class="error">*</span></td>
    <td class="error"><?php echo $file_error;    ?></td>
</tr>
<tr></tr>

```

```

}
if(empty($file)) {
    $file_error = "Required!";
}

```

Figure 7: Upload File Validation

The next field is to **“Agree to the Terms and Conditions”**. It is of checkbox type.

```

<tr>
    <td>
        <input type="checkbox" id="Agree" name="Agree">&nbsp;
        <b>I agree the Company's Terms and Conditions</b>
        <span class="error">*</span>
    </td>
    <td class="error"><?php echo $agree_error;    ?></td>
</tr>

```

```

if(!isset($_POST['Agree'])) {
    $agree_error = "Agree to the Terms & Conditions!";
}

```

Figure 13: Terms & Conditions Validation

The Output if any field is left Unanswered: -

First Name:	<input type="text" value="First Name"/> *	First Name is Required!
Last Name:	<input type="text" value="Last Name"/> *	Last Name is Required!
Email:	<input type="text" value="Email Id"/> *	Provide a Invalid Email!
Password:	<input type="text" value="Password"/> *	Password is Required!
Confirm Password:	<input type="text" value="Confirm Password"/> *	Confirm your Password!
Gender:	<input type="radio"/> Male <input type="radio"/> Female *	Provide your Gender!
Qualification:	<input type="checkbox"/> X <input type="checkbox"/> XII *	Give your Qualification!
Address:	<input type="text" value=""/> *	Please Enter your Address!
Contact Number:	<input type="text" value=""/> *	Contact must include 10 Digits!
Date of Birth:	DD <input type="text" value=""/> MM <input type="text" value=""/> YYYY <input type="text" value=""/> *	
Upload your Photo:	<input type="button" value="Choose File"/> No file chosen *	Required!
<input type="checkbox"/> I agree the Company's Terms and Conditions *		Agree to the Terms & Conditions!
<input type="button" value="Submit"/> <input type="button" value="Reset"/>		

Figure 8: Final Output if the field remains Unanswered

SUBMITTING THE DATA TO SERVER

In order to submit the data to the server, MySQL is used.

The concept is to start by accessing the value of every field through their names given at the time of PHP Validation.

- The first and the foremost thing is to ***create a Database and establish the connection***. To create the database, I have used phpMyAdmin.
- ***To establish the connection, the following query is to be executed.***

However, if the connection is not established, an ***Error Message*** is thrown.

```
$host = "localhost";
$dbuser = "root";
$password = "";
$dbname = "registration_records";

$connection = mysqli_connect($host, $dbuser, $password, $dbname);

if(mysqli_connect_errno()) {
    die("Connection Failed!");
}
```

- To access the value of every field from its respective name, following command is to be executed. Here, the ***First Name*** field and the ***Last Name*** Field are concatenated to make the complete name of the user.
- Similarly, the date of birth is created by concatenating the ***Date, Month*** and the ***Year Fields***.
- Also, as the ***Qualification*** field is an array, ***implode()*** function is used to convert it into a string by a comma separator as shown in the figure.

```

if(isset($_POST['submit'])) {
    $fname = $_POST['Fname'];
    $lname = $_POST['Lname'];
    $email = $_POST['Email'];
    $password = $_POST['Password'];
    $confirm = $_POST['Conpassword'];
    $gender = $_POST['Gender'];
    $quali = $_POST['quali'];
    $address = $_POST['Address'];
    $contact = $_POST['Contact'];
    $date = $_POST['Date'];
    $month = $_POST['Month'];
    $year = $_POST['Year'];
    $file = $_POST['Myfile'];
    $agree = $_POST['Agree'];

    $name = $fname. " ".$lname;
    $quali_each = implode(" ", $quali);
    $dob = $date."th ".$month." ".$year;

```

For finally inserting the record into the table of the server, the following queries are to be executed. Here, the name of the table is – “data_entries”.

```

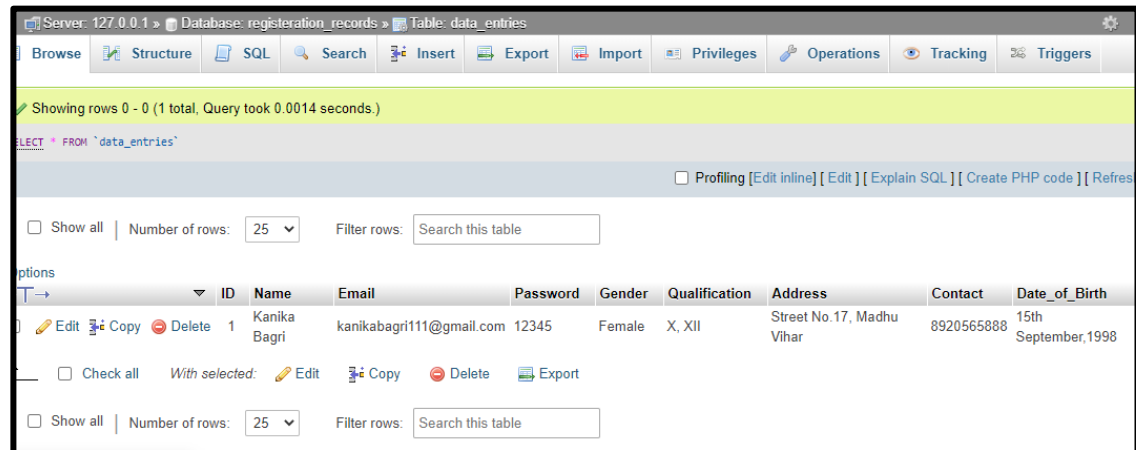
else {
    $sql = "INSERT INTO data_entries(Name, Email, Password, Gender, Qualification, Address, Contact,
    Date_of_Birth) VALUES('$name', '$email', '$password', '$gender', '$quali_each', '$address', '$contact', '$dob')
    ";

    $result = mysqli_query($connection, $sql);
    if(!$result) {
        echo "<script>alert(Data not Inserted!)</script>";
    }
    else {
        echo "<script>alert(Data Inserted Successfully!)</script>";
    }
}
?>

```

Figure 9: Execution of SQL Query

THE DATABASE



Server: 127.0.0.1 » Database: registration_records » Table: data_entries

Showing rows 0 - 0 (1 total, Query took 0.0014 seconds.)

SELECT * FROM `data_entries`

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

ID	Name	Email	Password	Gender	Qualification	Address	Contact	Date_of_Birth
1	Kanika Bagri	kanikabagri111@gmail.com	12345	Female	X, XII	Street No.17, Madhu Vihar	8920565888	15th September,1998

☐ Check all | With selected: Edit Copy Delete Export

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Note:

- 1. The Assignment will work if and only if saved to the htdocs folder of XAMPP i.e. on the Localhost.***
- 2. The Backend will work if and only if the database included with the folder is imported on the phpMyAdmin of the User.***