Terna Engineering College
Computer Engineering Department

Program: Sem V

Course: Web Technology Laboratory (CSL504)

Faculty: Mrs Reshma Koli

LAB Manual

PART A

(PART A: TO BE REFERRED BY STUDENTS)

Experiment No.09

A.1 Aim:

Dynamic Web page using PHP and AJAX

Create a simple application that allows users to upload the image dynamically or provide suggestions to the user based on data entered or based on data entered by the user to hide or explore certain page content.

A.2 Prerequisite:

1. Knowledge of PHP,AJAX.
2. Web Browser, Web Server (PHP, MySql), Notepad.
3. Installation of XAMPP or WAMP Server.

A.3 Outcome:

After successful completion of this experiment students will be able to - Evaluate client and server side technologies and create Interactive web pages using PHP and AJAX with database connectivity using MySQL (XAMPP/WAMP).

A.4 Theory:

- ➔ AJAX is an acronym standing for Asynchronous JavaScript and XML and this technology helps us to load data from the server without a browser page refresh.
- ➔ If you are new with AJAX, I would recommend you go through our Ajax Tutorial before proceeding further.
- ➔ JQuery is a great tool which provides a rich set of AJAX methods to develop next generation web application

Loading Simple Data

- ➔ This is very easy to load any static or dynamic data using JQuery AJAX. JQuery provides load() method to do the job –

Syntax

➔ Here is the simple syntax for load() method –
[selector].load(URL, [data], [callback]);

Getting JSON Data

➔ There would be a situation when the server would return JSON string against your request. JQuery utility function getJSON() parses the returned JSON string and makes the resulting string available to the callback function as the first parameter to take further action.

Syntax

➔ Here is the simple syntax for getJSON() method –
[selector].getJSON(URL, [data], [callback]);

Passing Data to the Server

➔ Many times you collect input from the user and you pass that input to the server for further processing. JQuery AJAX made it easy enough to pass collected data to the server using the data parameter of any available Ajax method.

Difference in AJAX and Conventional CGI Program

➔ Try these two examples one by one and you will feel the difference. While trying AJAX example, there is not discontinuity and you get the response very quickly, but when you try the standard GCI example, you would have to wait for the response and your page also gets refreshed

Here is a list of some famous web applications that make use of AJAX.

1. Google Maps

A user can drag an entire map by using the mouse, rather than clicking on a button.

- http://maps.google.com/

2. Google Suggest

As you type, Google will offer suggestions. Use the arrow keys to navigate the results.

- http://www.google.com/webhp?complete=1&hl=en

3. Gmail

Gmail is a webmail, built on the idea that email can be more intuitive, efficient and useful.

- http://gmail.com/

4. Yahoo Maps (new)

Now it's even easier and more fun to get where you're going!

- http://maps.yahoo.com/

AJAX = Asynchronous JavaScript And XML. AJAX is not a programming language. AJAX just uses a combination of: A browser built-in XMLHttpRequest object (to request data from a web server) JavaScript and HTML DOM (to display or use the data)

AJAX is a technique for creating fast and dynamic web pages. AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

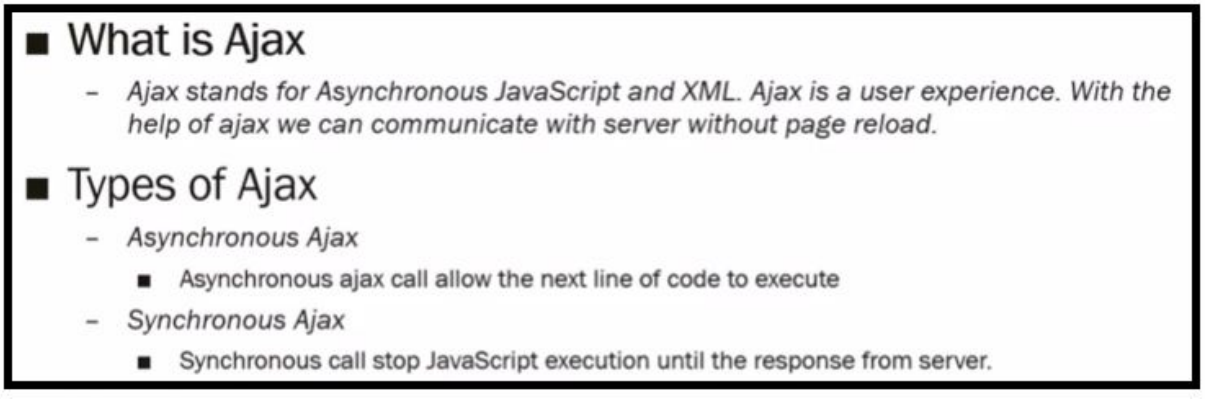
- ➔ Read data from a web server - after the page has loaded
- ➔ Update a web page without reloading the page
- ➔ Send data to a web server - in the background

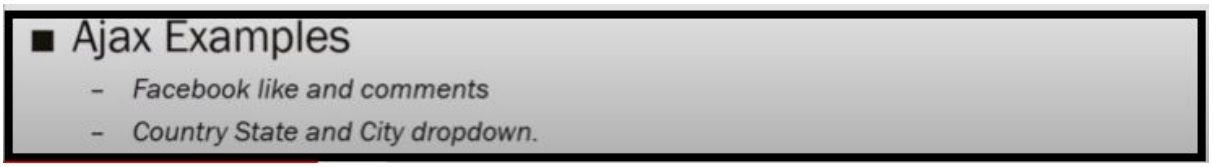
Advantages:

- ➔ Reduced Bandwidth Usage and Increased speed — AJAX uses client-side scripting to communicate with the web server and exchange data using JavaScript.
- ➔ Using AJAX, you can cut down on network load and bandwidth usage and retrieve only the data that is required to give you faster interfaces and better responsive times.

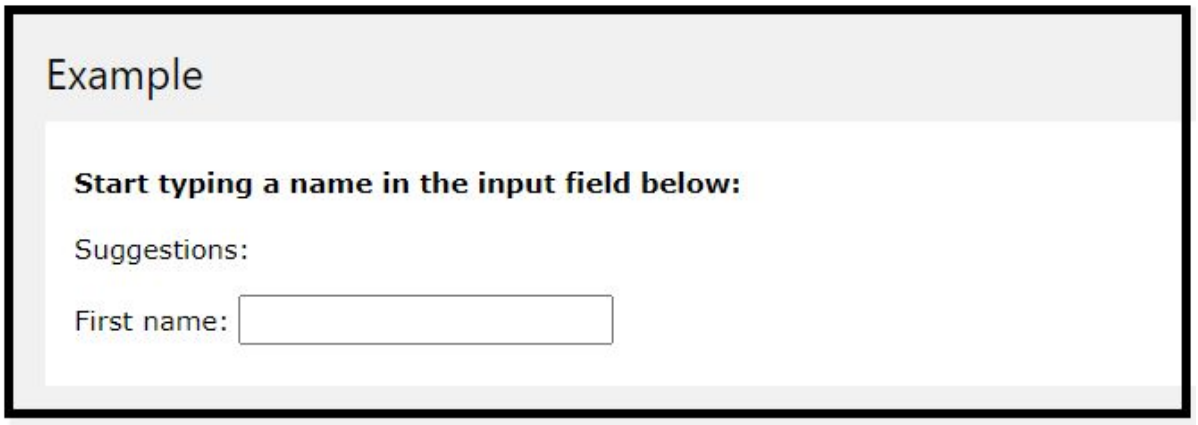
Disadvantages:

- ➔ AJAX application would be a mistake because search engines would not be able to index an AJAX application





AJAX and PHP Example:
The following example demonstrates how a web page can communicate with a web server while a user types characters in an input field:



Example Explained
In the example above, when a user types a character in the input field, a function called showHint() is executed.
The function is triggered by the onkeyup event.

Here is the HTML and PHP code:

Hint.html

```
<html>
<body>
<p><b>Start typing a name in the input field below:</b></p>
<p>Suggestions: <span id="txtHint"></span></p>
<form>
First name: <input type="text" onkeyup="showHint(this.value)">
</form>
<script>
function showHint(str) {
  if (str.length == 0) {
    document.getElementById("txtHint").innerHTML = "";
    return;
  } else {
    var xmlhttp = new XMLHttpRequest();
    xmlhttp.onreadystatechange = function() {
      if (this.readyState == 4 && this.status == 200) {
        document.getElementById("txtHint").innerHTML = this.responseText;
      }
    };
    xmlhttp.open("GET", "gethint.php?q=" + str, true);
    xmlhttp.send();
  }
}
```

```
</script>
</body>
</html>
```

Gethint.php

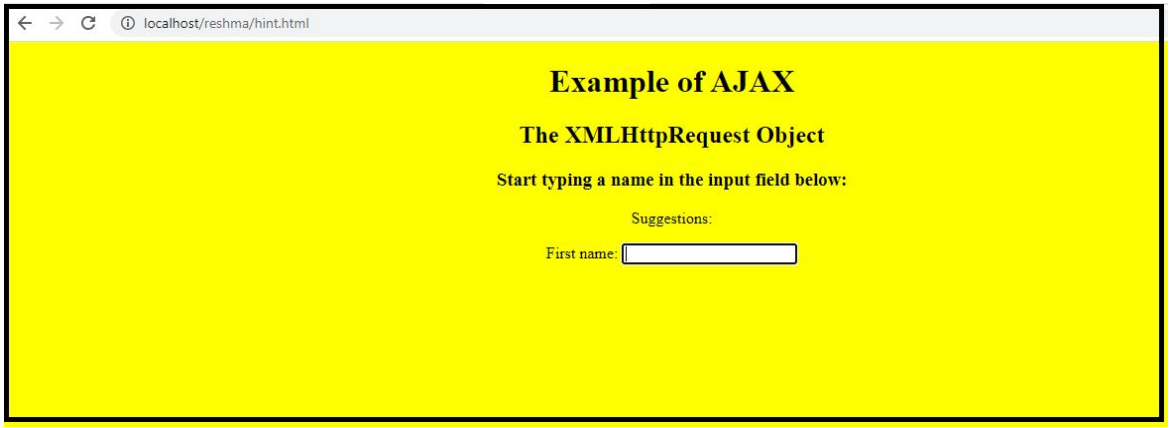
The PHP file checks an array of names, and returns the corresponding name(s) to the browser:

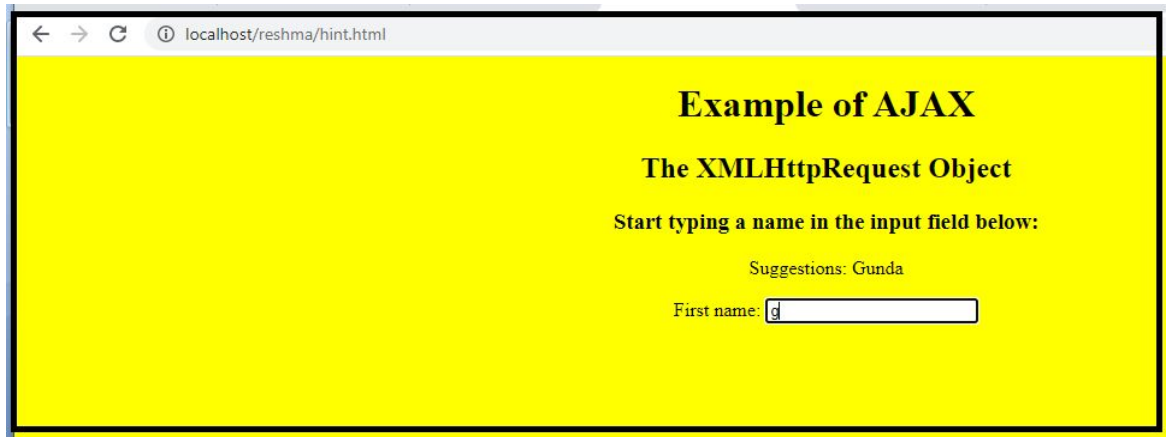
```
<?php
// Array with names
$a[] = "Anna";
$a[] = "Brittany";
$a[] = "Cinderella";
$a[] = "Diana";
$a[] = "Eva";
$a[] = "Fiona";
$a[] = "Gunda";
$a[] = "Hege";
$a[] = "Inga";
$a[] = "Johanna";
$a[] = "Kitty";
$a[] = "Linda";
$a[] = "Nina";
$a[] = "Ophelia";
$a[] = "Petunia";
$a[] = "Amanda";
$a[] = "Raquel";
$a[] = "Cindy";
$a[] = "Doris";
$a[] = "Eve";
$a[] = "Evita";
$a[] = "Sunniva";
$a[] = "Tove";
$a[] = "Unni";
$a[] = "Violet";
$a[] = "Liza";
$a[] = "Elizabeth";
$a[] = "Ellen";
$a[] = "Wenche";
$a[] = "Vicky";

// get the q parameter from URL
```

```php
$q = $_REQUEST["q"];
$hint = "";
// lookup all hints from array if $q is different from ""
if ($q !== "") {
  $q = strtolower($q);
  $len=strlen($q);
  foreach($a as $name) {
    if (stristr($q, substr($name, 0, $len))) {
      if ($hint === "") {
        $hint = $name;
      } else {
        $hint .= ", $name";
      }
    }
  }
}
// Output "no suggestion" if no hint was found or output correct values
echo $hint === "" ? "no suggestion" : $hint;
?>
```

Output





PART B

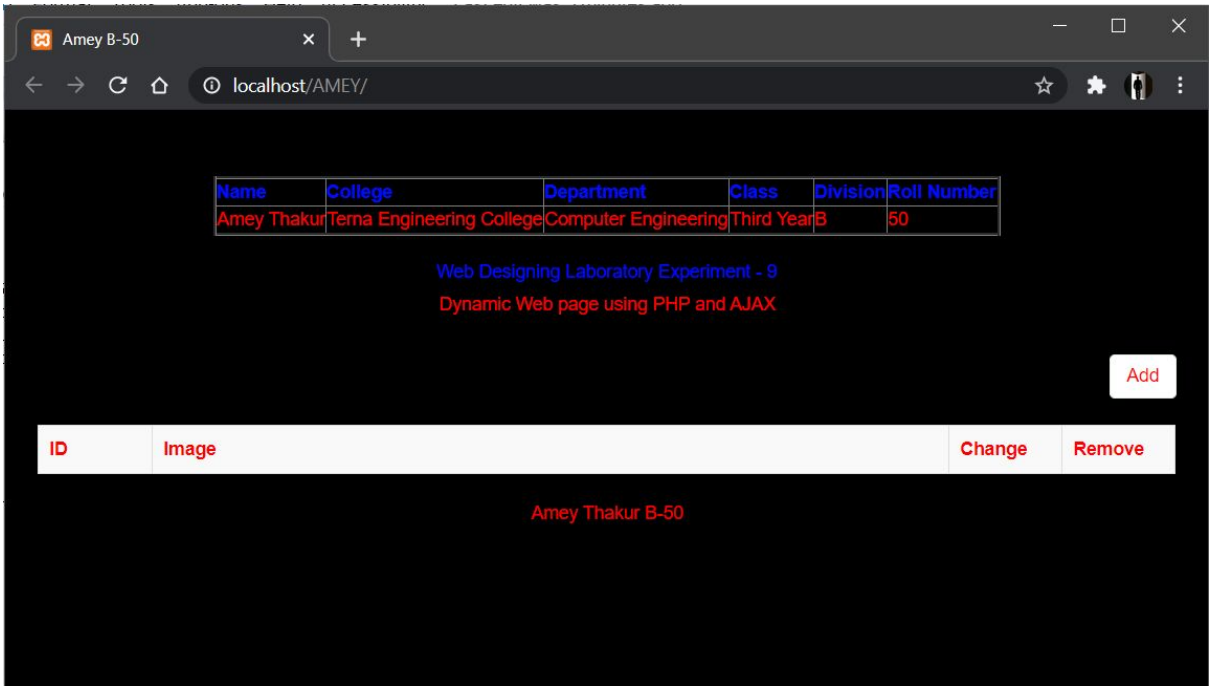
(PART B: TO BE COMPLETED BY STUDENTS)

(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 Blackboard access available)

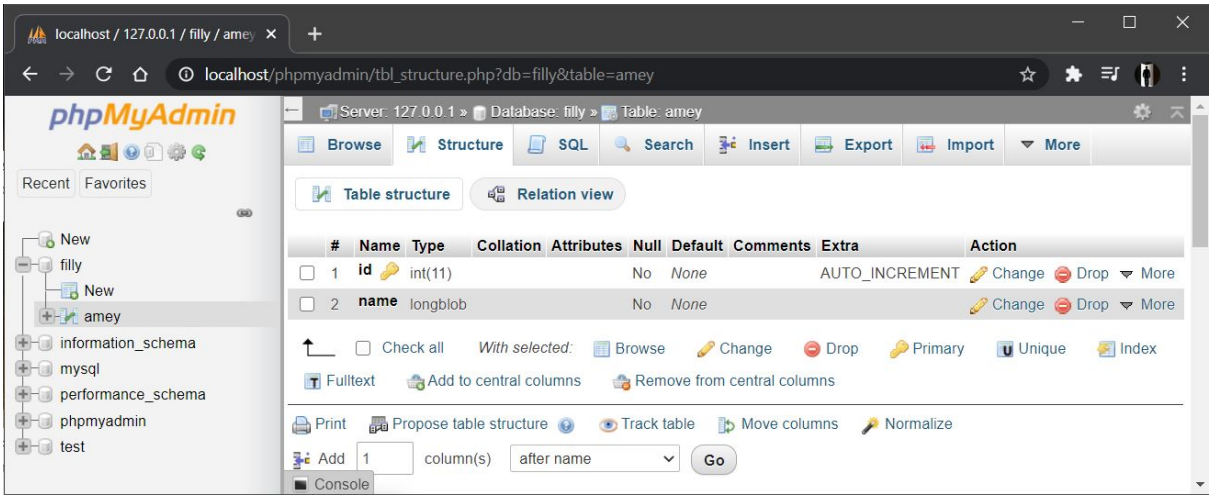
Roll No. 50	Name: Amey Thakur
Class: TE-Comps B	Batch: B3
Date of Experiment: 15/09/2020	Date of Submission: 15/09/2020
Grade :	

B.1 Web page Snapshot: (Add all snapshots of output.)

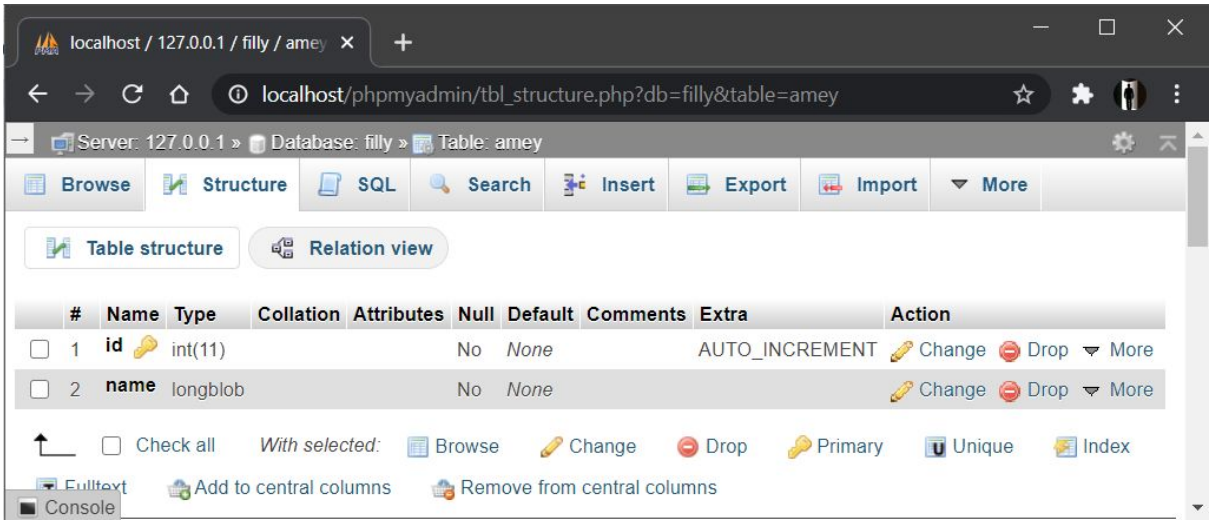
- Dynamic Web page using PHP and AJAX



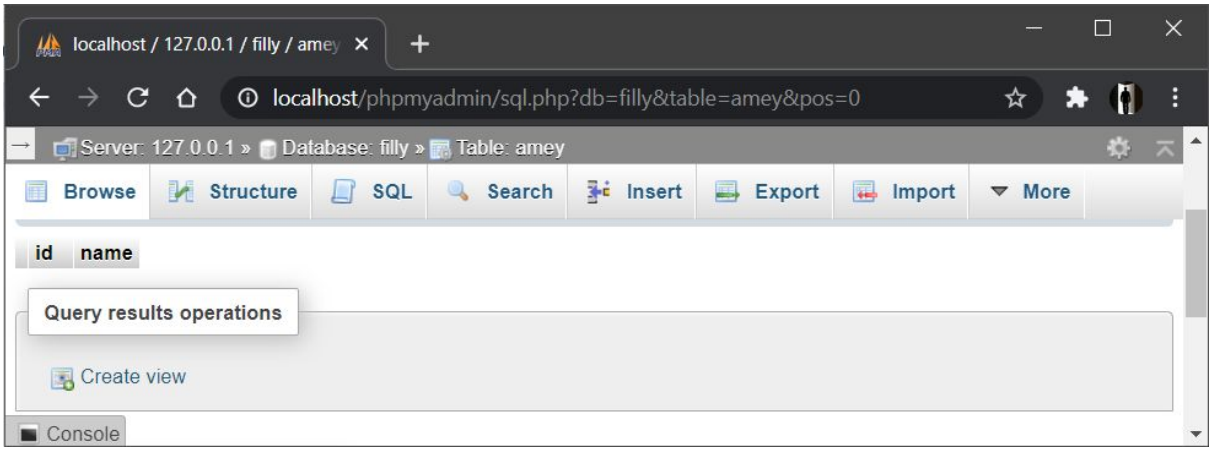
- Database: filly >> Table: amey



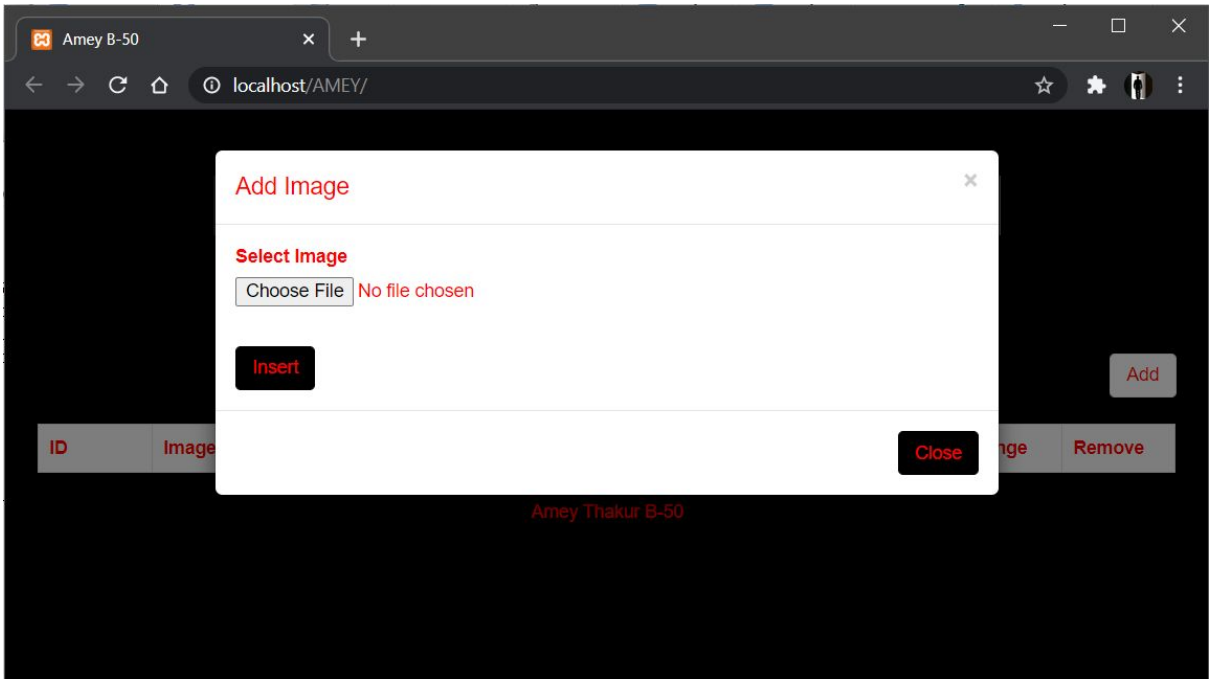
- Database > Table > Structure



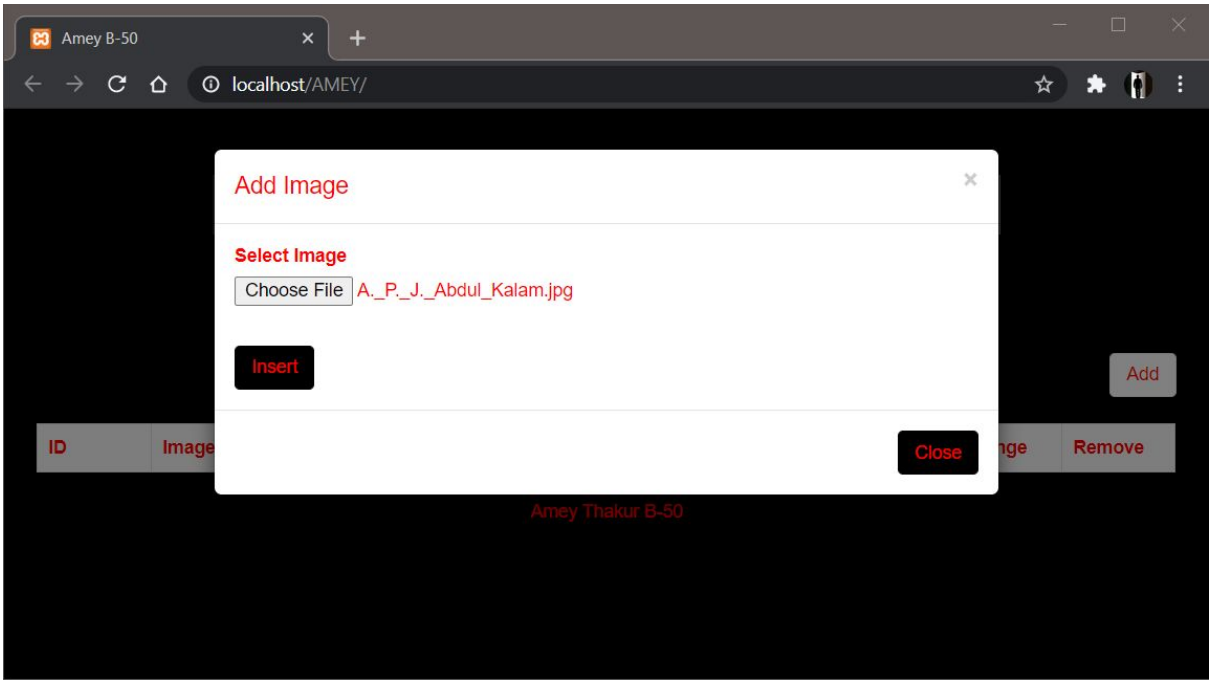
- Database > Table > Browse



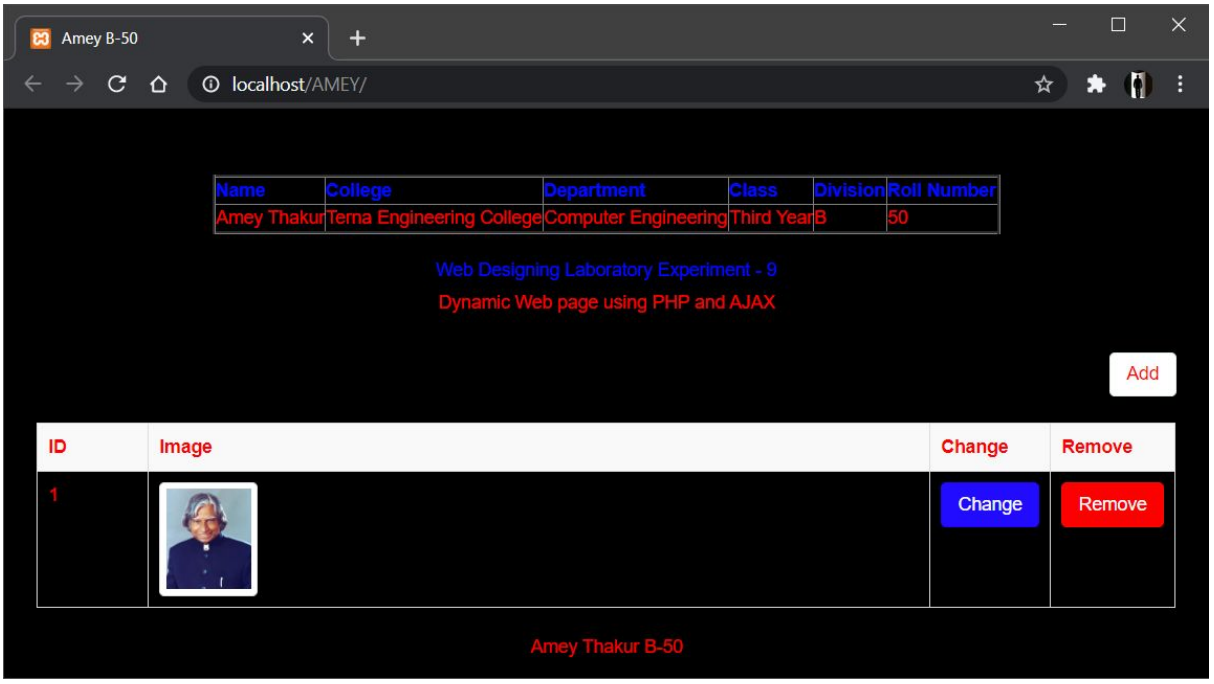
- Insert Image



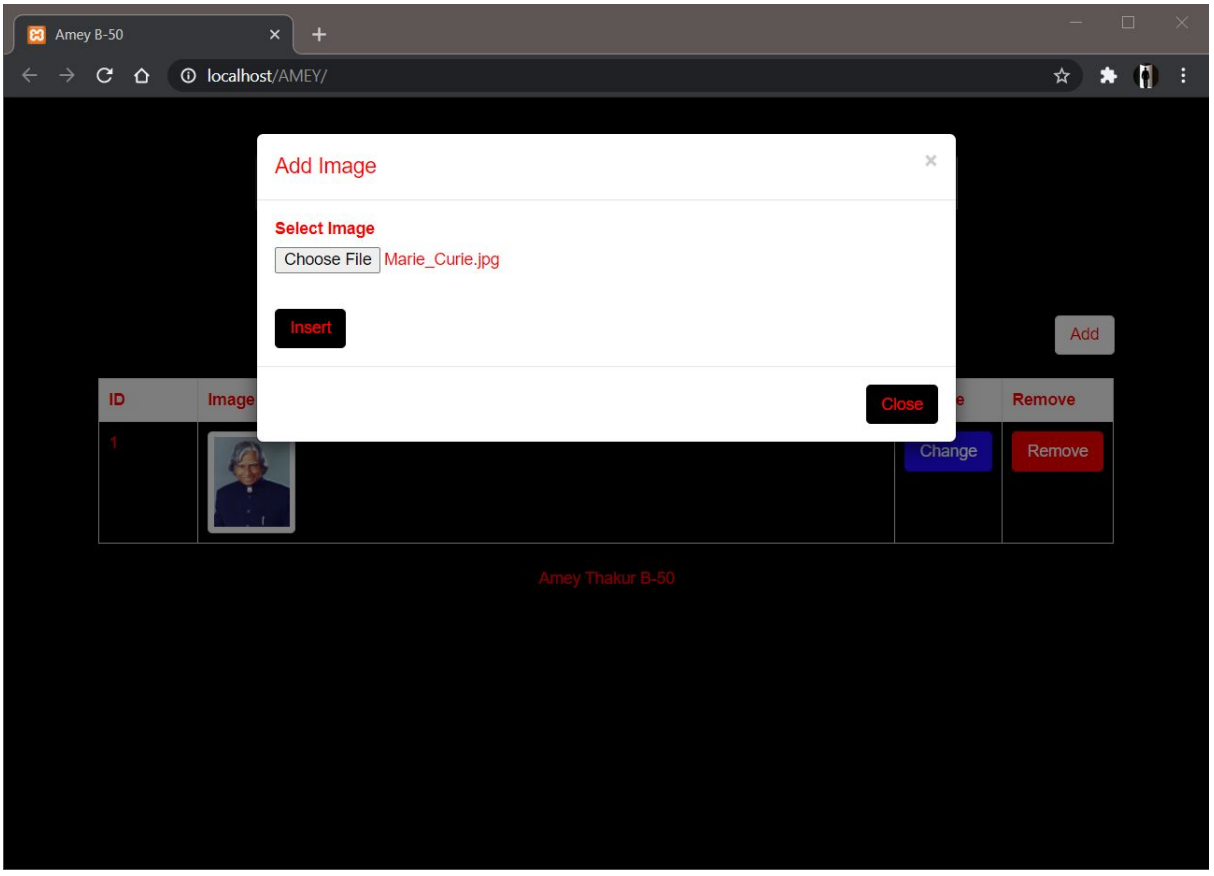
- Select Image



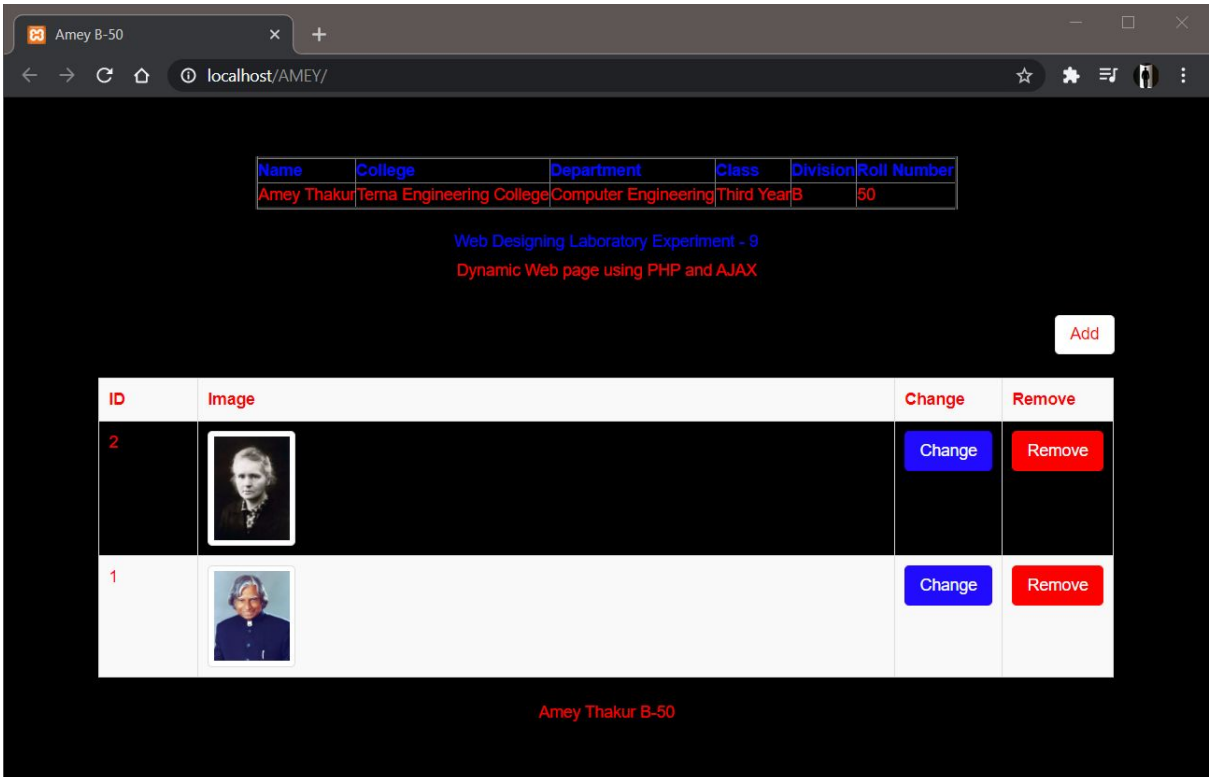
- Image Inserted



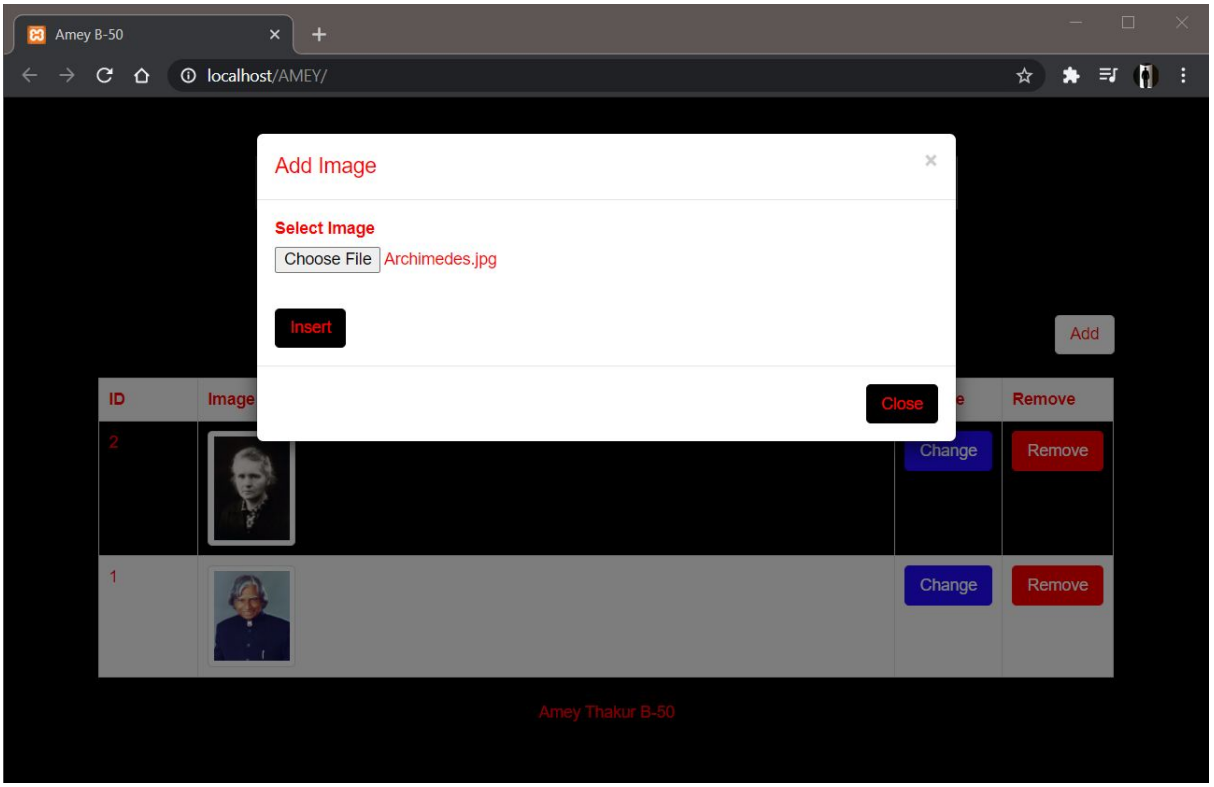
- Insert Image



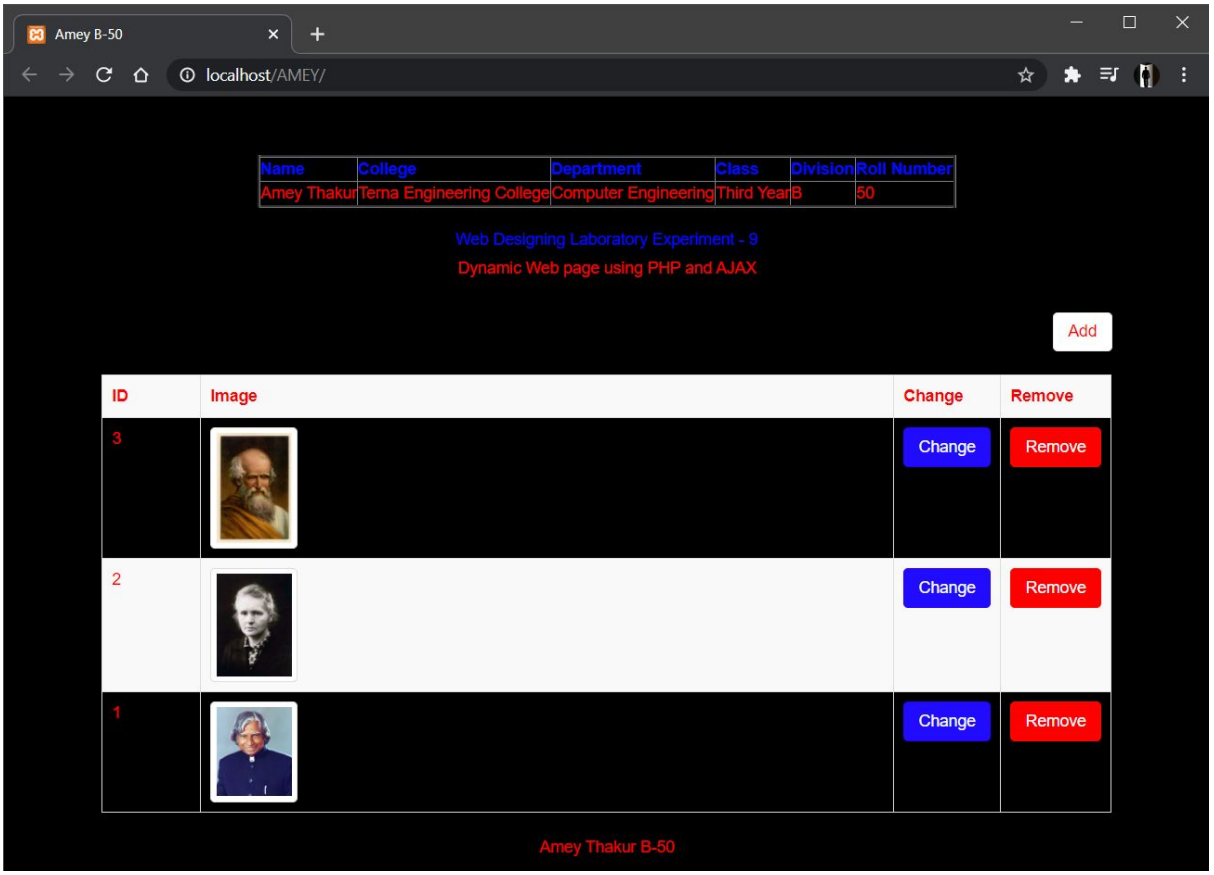
- Image Inserted



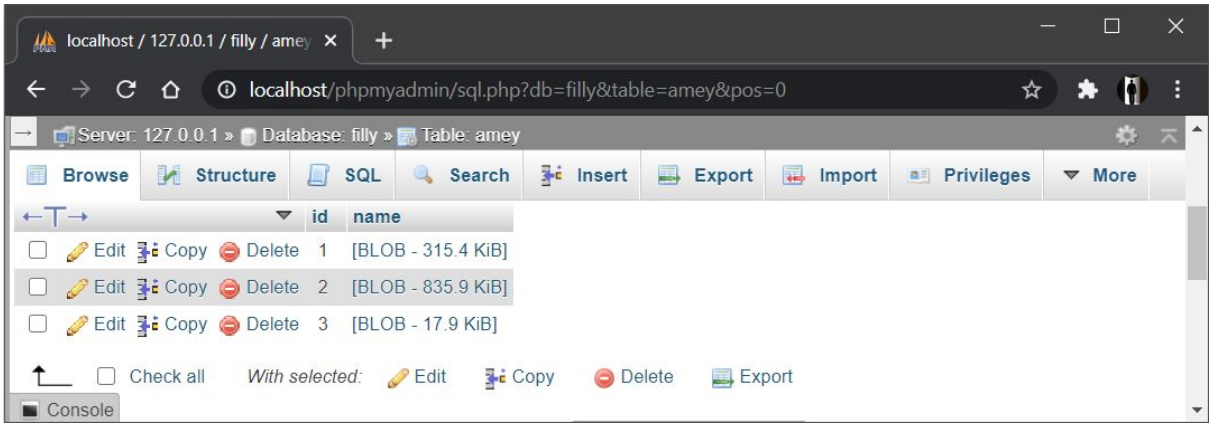
- Insert Image



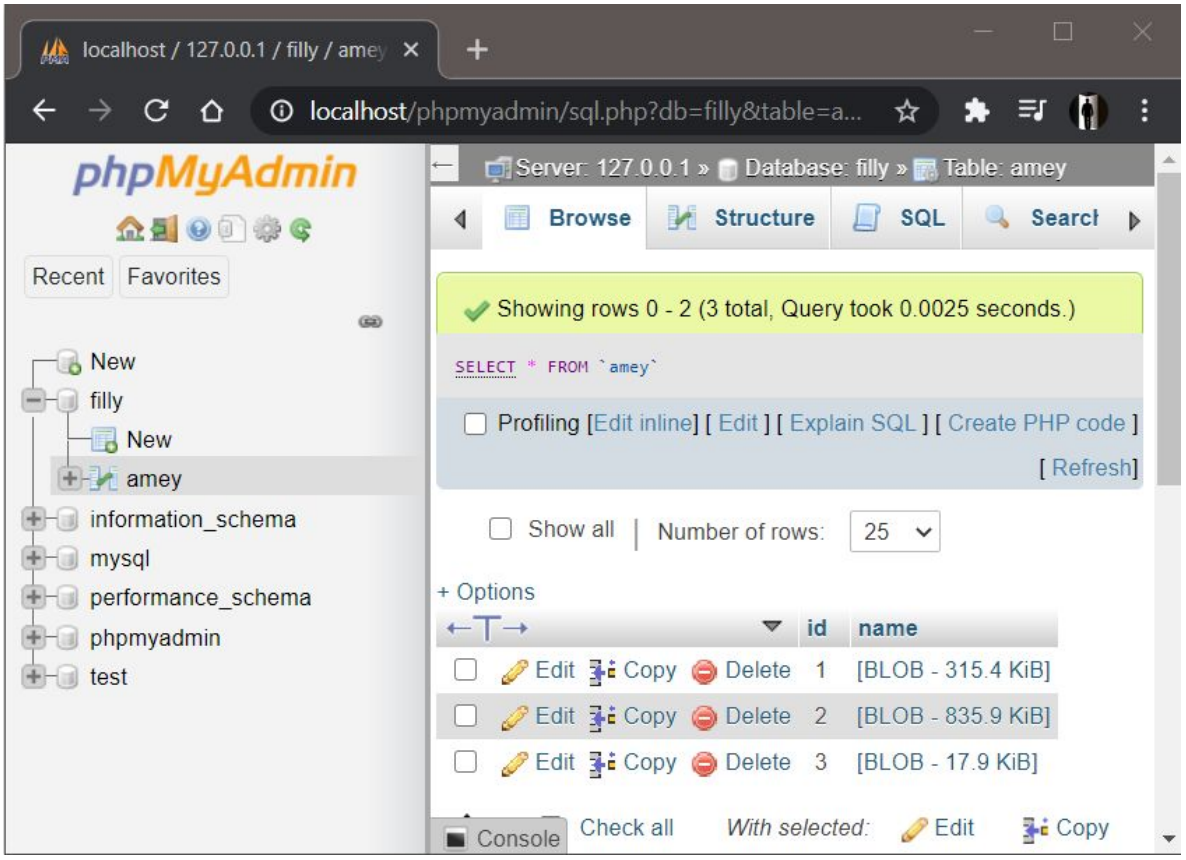
- Image Inserted



- Recorded data



- Form inputs > Database > Table > Browse



- Database > Table > Inputs

```
SELECT * FROM `amey`
```

id	name
1	[BLOB - 315.4 KiB]
2	[BLOB - 835.9 KiB]
3	[BLOB - 17.9 KiB]

B.2 Web page source code:

(Add source code of web pages)

- **index.php**

```
<!DOCTYPE html>
<html>
	<head>
	<title>Amey B-50</title>

<script	src="https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/jquery.min.js">
</script>

<link	rel="stylesheet"	href=
"https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" />

<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js">
</script>
</head>

	<body style="background-color:black;">
		<br />
		<br />

<aside id="sidebar">
		<p>
		<table border="3" bgcolor="#000000" align="center" cellspacing= "20">
		<tr style="color:#000dff">
		<th >Name</th>
		<th>College</th>
		<th>Department</th>
		<th>Class</th>
		<th>Division</th>
		<th>Roll Number</th>
		</tr>

		<tr style="color:#ff0000">
		<td>Amey Thakur</td>
		<td>Terna Engineering College</td>
		<td>Computer Engineering</td>
		<td>Third Year</td>
		<td>B</td>
		<td>50</td>
		</tr>
		</table>
		</p>
</aside>
```

```
<div class="container" style="width:900px;">
        <h5 align="center" style="color:#000dff" >Web Designing Laboratory
        Experiment - 9 </h5>
        <h5 align="center" style="color:#ff0000">Dynamic Web page using PHP and
        AJAX </h5>
                        <br />
        <div align="right">
                <button style="color:#ffffff" type="button" name="add" id="add"
                class="btn btn- success">Add</button>
        </div>
                        <br />
          <div id="image_data">
          </div>
 </div>

        <footer>
                 <center><p style="color:#f70000">Amey Thakur B-50</p></center>
        </footer>
    </body>
</html>

<div id="imageModal" class="modal fade" role="dialog">
        <div class="modal-dialog">
        <div class="modal-content">
        <div class="modal-header">
                <button type="button" class="close" data-dismiss="modal"> &times;
                </button>
                <h4 class="modal-title">Add Image</h4>
        </div>
        <div class="modal-body">
        <form id="image_form" method="post" enctype="multipart/form-data">
                <p><label>Select Image</label>
                <input type="file" name="image" id="image" /></p><br />
                <input type="hidden" name="action" id="action" value="insert" />
                <input type="hidden" name="image_id" id="image_id" />
                <input type="submit" name="insert" id="insert" value="Insert"
                class="btn btn-info" />
        </form>
        </div>
        <div class="modal-footer">
        <button type="button" class="btn btn-default" data-dismiss="modal"> Close
        </button>
        </div>
        </div>
        </div>
</div>
```

```
<script>
$(document).ready(function()
{
    fetch_data();
    function fetch_data()
    {
        var action = "fetch";
        $.ajax
        ({
            url:"filly.php",
            method:"POST",
            data:{action:action},
            success:function(data)
            {
                $('#image_data').html(data);
            }
        })
    }
    $('#add').click(function()
    {
        $('#imageModal').modal('show');
        $('#image_form')[0].reset();
        $('.modal-title').text("Add Image");
        $('#image_id').val('');
        $('#action').val('insert');
        $('#insert').val("Insert");
    });
    $('#image_form').submit(function(event)
    {
        event.preventDefault();
        var image_name = $('#image').val();
        if(image_name == '')
        {
            alert("Please Select Image");
            return false;
        }
        else
            {
                var extension = $('#image').val().split('.').pop().toLowerCase();
                if(jQuery.inArray(extension, ['gif','png','jpg','jpeg']) == -1)
                {
                    alert("Invalid Image File");
                    $('#image').val('');
                    return false;
                }
                else
                    {
                        $.ajax
```

```
            ({
                url:"filly.php",
                method:"POST",
                data:new FormData(this),
                contentType:false,
                processData:false,
                success:function(data)
                {
                    alert(data);
                    fetch_data();
                    $('#image_form')[0].reset();
                    $('#imageModal').modal('hide');
                }
            });
        }
    }
});
$(document).on('click', '.update', function()
{
    $('#image_id').val($(this).attr("id"));
    $('#action').val("update");
    $('.modal-title').text("Update Image");
    $('#insert').val("Update");
    $('#imageModal').modal("show");
});
$(document).on('click', '.delete', function()
{
    var image_id = $(this).attr("id");
    var action = "delete";
  if(confirm("Are you sure you want to remove this image from database?"))
    {
        $.ajax
        ({
            url:"filly.php",
            method:"POST",
            data:{image_id:image_id, action:action},
            success:function(data)
            {
                alert(data);
                fetch_data();
            }
        })
    }
    else
        {
            return false;
        }
    });
```

```
        });
</script>
```

- **filly.php**

```
<?php
if (isset($_POST["action"]))
{
    $connect = mysqli_connect("localhost", "root", "", "filly");
    if ($_POST["action"] == "fetch")
    {
        $query = "SELECT * FROM amey ORDER BY id DESC";
        $result = mysqli_query($connect, $query);
        $output = '
        <table class="table table-bordered table-striped">
        <tr>
            <th width="10%">ID</th>
            <th width="70%">Image</th>
            <th width="10%">Change</th>
            <th width="10%">Remove</th>
        </tr>
        ';
        while ($row = mysqli_fetch_array($result))
        {
            $output.= '
            <tr>
            <td>' . $row["id"] . '</td>
            <td>
                <img src="' . base64_encode($row['name']) . '" height="60" width="75" class="img-thumbnail" />
            </td>
            <td><button type="button" name="update" class="btn btn-warning bt-xs update" id="' . $row["id"] . '">Change</button></td>
        <td><button type="button" name="delete" class="btn btn-danger bt-xs delete" id="' . $row["id"] . '">Remove</button></td>
            </tr>
            ';
        }
        $output.= '</table>';
        echo $output;
    }
    if ($_POST["action"] == "insert")
```

```
    {
        $file = addslashes(file_get_contents($_FILES["image"]["tmp_name"]));
        $query = "INSERT INTO amey(name) VALUES ('$file')";
        if (mysqli_query($connect, $query))
        {
            echo 'Image Inserted into Database';
        }
    }
    if ($_POST["action"] == "update")
    {
        $file = addslashes(file_get_contents($_FILES["image"]["tmp_name"]));
       $query = "UPDATE amey SET name = '$file' WHERE id = '" . $_POST["image_id"]
. "'";
         if (mysqli_query($connect, $query))
         {
             echo 'Image Updated into Database';
         }
    }
    if ($_POST["action"] == "delete")
    {
        $query = "DELETE FROM amey WHERE id = '" . $_POST["image_id"] . "'";
        if (mysqli_query($connect, $query))
        {
            echo 'Image Deleted from Database';
        }
    }
}
?>
```

B.3 Conclusion:

- AJAX is about updating parts of a web page, without reloading the whole page. AJAX is a technique for creating fast and dynamic web pages.
- AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.
- Classic web pages, (which do not use AJAX) must reload the entire page if the content should change.
- We learnt how to create a simple application that allows users to upload the image dynamically.

B.4 Question of Curiosity:

Q.1 Write features of AJAX.
Ans:
Features of AJAX -

1. User-Friendly.
2. It makes a web page faster.
3. Free from server technology.
4. Boosts the performance of a web page.
5. Support for Live data binding.
6. Assist the Data View control.
7. Assist Client-side template.
8. Responsive user interfaces.
9. Usage of server resources is less.
10. JavaScript is used and so processing is the same for all browser types.
11. Ajax helps to develop faster and more interactive web applications.
12. Not required to completely reload a page due to this server using less bandwidth.

Q.2 How AJAX is different from conventional programs is explained with an example.
Ans:
Let's take one example: we have a website for a user management system.

- ➔ Without Ajax, Whenever we want to access an account, we press the login link. New login page opens and enters the username, password, and login successfully. It's a time consuming process, inefficient for only small amounts of data(Login page) exchange required.
- ➔ Now With Ajax, We want to access the account, we press the login link. Background Javascript call to a server for fetching login portion and update/add to current web page. Now we can enter username, password and login successfully. It takes a small bit of time for this process.