LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING
(AUTONOMOUS)

20CS63 – SERVER-SIDE SCRIPTING LAB

Department
of
COMPUTER SCIENCE & ENGINEERING

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1	Pursue higher education, entrepreneurship and research to compete at global level.
PEO2	Design and develop products innovatively in the area of computer science and engineering and in other allied fields.
PEO3	Function effectively as individuals and as members of a team in the conduct of interdisciplinary projects; and even at all the levels with ethics and necessary attitude.
PEO4	Serve ever-changing needs of the society with a pragmatic perception.

PROGRAMME OUTCOMES (POs):

PO1	**Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.
PO2	**Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of Mathematics, natural sciences and engineering sciences.
PO3	**Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and Environmental considerations.
PO4	**Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	**Modern tool usage:** Create, select and apply appropriate techniques, resources and modern engineering and IT tools including prediction and modeling to complex Engineering activities with an understanding of the limitations
PO6	**The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent Responsibilities relevant to the professional engineering practice
PO7	**Environment and sustainability:** Understand the impact of the professional Engineering solutions in societal and environmental contexts and demonstrate the knowledge of, and need for sustainable development.
PO8	**Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	**Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	**Communication:**Communicateeffectivelyoncomplexengineeringactivitieswiththe engineering community and with society at large, such as, being able to
PO11	**Project management and finance:** Demonstrate knowledge and understanding of the Engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	**Life-long learning:** Recognize the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAMME SPECIFIC OUTCOMES (PSOs):

PSO 1	The ability to apply Software Engineering practices and strategies in software project development using open source programming environment for the success of organization.
PSO 2	The ability to design and develop computer programs in networking, web applications and IoT as per the society needs.
PSO 3	To inculcate an ability to analyze, design and implement database applications.

VISION

The Computer Science & Engineering aims at providing continuously stimulating educational environment to its students for attaining their professional goals and meet the global challenges.

MISSION

1. To develop a strong theoretical and practical background across the computer science discipline with an emphasis on problem solving.
2. To inculcate professional behavior with strong ethical values, leadership qualities, innovative thinking and analytical abilities into the student.
3. Expose the students to cutting edge technologies which enhance their employability and knowledge.
4. Facilitate the faculty to keep track of latest developments in their research areas and encourage the faculty to foster the healthy interaction with industry.

B.Tech. (VI Sem.) **20CS63-SERVER-SIDE SCRIPTING LAB**

L	T	P	Cr.
0	0	3	1.5

Pre-requisite : JAVA Programming & Web Technologies

Course Educational Objective: Students will learn the importance of client-server architecture in the web application development and able to develop dynamic data driven (server-side) web applications by using advanced technologies (AJAX, PHP)

Course Outcomes (CO): *At the end of this course, the student will be able to:*

CO1: Understand the differences between server-side and client-side script, develop simple server-server-side web applications. **(Understand - L2)**

CO2: Identify the importance of AJAX, PHP programming constructs to design server-side web applications. **(Remember - L1)**

CO3: Develop Dynamic Data Driven (Server-side) Web Applications by using AJAX, PHP. **(Apply - L3)**

CO 4: Improve individual / teamwork skills, communication & report writing skills with ethical values.

Module 1:

a) Develop a static web application and deploy it in any one of the web servers (WAMP/Apache Tomcat/IIS).
b) Develop a JavaScript program to validate the client-side user input data (Example: username, password, email, phone number).

Module 2:

a) Create a XMLHttpRequest with a callback function and retrieve data from a TXT file by using AJAX.
b) Create an XMLHttpRequest to retrieve data from an XML file and display the data in an HTML table by using AJAX.

Module 3:

a) Develop AJAX application to demonstrate how a web page can communicate with a web server while a user type characters in an input field.
b) Develop a simple web application to demonstrate how a web page can fetch information from a database with AJAX.

Module 4:

a) Develop a PHP program to illustrate the PHP Form handling by using GET and POST methods.
b) Develop a PHP program to demonstrate the importance of include() and require() functions.

Module 5:

a) Develop a PHP program to manage the state information about the user by using PHP Cookie.
b) Develop a PHP program to manage the state information about the user by using PHP Session.

Module 6:

a) Develop a PHP program to perform various file handling operations.
b) Develop a PHP program to demonstrate the process of uploading and downloading the file.

Module 7:

a) Develop a PHP program to describe the importance of mail function to send the email.
b) Develop a PHP and AJAX program which demonstrates how a web page can communicate with a web server while a user types characters in an input filed.

Module 8:

a) Develop a PHP web application which demonstrates the process of storing HTML form data into MySQL database.
b) Develop a PHP web application which demonstrates the process of retrieving table data from the MySQL database and display it in the HTML table.

Module 9:

a) Develop a PHP web application which demonstrates the process of deleting a particular record in MySQL database table.
b) Develop a PHP web application to illustrate the process of updating MySQL database table data.

Module 10:

a) Develop a PHP web application which demonstrates how to upload image into database and display it.
b) Develop a PHP web application which demonstrates the process of encode and decode JSON by the help of json_encode() and json_decode() functions.

REFERENCE BOOKS:

1. Steven Holzner, "PHP: The Complete Reference", McGraw-Hill Education, 2007.
2. Kevin Tatroe, Peter MacIntyre, RasmusLerdorf, "Programming in PHP", O'Reilly, 3rd Edition, 2013.
3. Luke Welling, Laura Thomson, "PHP and MySQL Web Development",Pearson Education, 5th Edition, 2011
4. Lynn Beighley, Michael Morrison, "Head First PHP & MySQL: A Brain-Friendly Guide", O'Reilly, 1st Edition, 2007

Module 1:

a) Develop a static web application and deploy it in any one of the web servers (WAMP/Apache Tomcat/IIS).

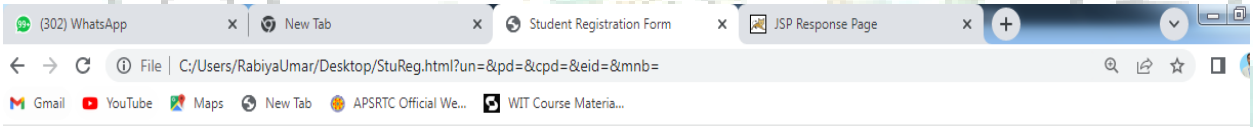
Step-1: Install WAMP/XAMP/Apache Tomcat/IIS
Step-2: Create a Static web page.

StuReg.html

```
<html>
<head>
<title>Student Registration Form</title>
</head>
<body>
<form name="stu" action="http://localhost:8080/examples/jsp/StuReg.jsp">
<h2><center>Student Registration Form Using Table in HTML</center></h2>
<table align="center" cellpadding = "10">

<!---------------------- User Name ------------------------------------------>
<tr>
<td>User Name</td>
<td><input type="text" name="un" id="t1" class="tb"  maxlength="25" placeholder="User Name" />
(Max 25 Characters Allowed)
</td>
</tr>

<!------------------------ Password ------------------------------------->
<tr>
<td>Password</td>
<td><input type="password" name="pd" maxlength="25" placeholder="Password"/>
(Max 25 Characters Allowed)
</td>
</tr>

<!------------------------ Re-Password ------------------------------------->
<tr>
<td>Confirm Password</td>
<td><input type="password" name="cpd" maxlength="25" placeholder="Confirm Password"/>
(Max 25 Characters Allowed)
</td>
</tr>

<!------------------------- Email ID ------------------------------------->
<tr>
<td>Email ID</td>
<td><input type="email" name="eid" maxlength="30" placeholder="Sample@gmail.com"/></td>
</tr>
```

```
<!------------------------- Mobile Number ------------------------------------>
<tr>
<td>Mobile Number</td>
<td>
<input type="text" name="mnb" maxlength="10" placeholder="1234567890"/>
(10 Digits Allowed)
</td>
</tr>
 <tr>
<td colspan="2" align="center">
<input type="reset" value="Clear Form" id="res" class="btn" />
   <input type="submit" value="Validate" class="btn"  />
</td>
</tr>
</table>

</form>

</body>
</html>
```

OUTPUT:



Student Registration Form Using Table in HTML

User Name	User Name	(Max 25 Characters Allowed)
Password	Password	(Max 25 Characters Allowed)
Confirm Password	Confirm Password	(Max 25 Characters Allowed)
Email ID	Sample@gmail.com	
Mobile Number	1234567890	(10 Digits Allowed)

Clear Form Validate

Step-3: Create a JSP Page and save at C:\Program Files\Apache Software Foundation\Tomcat 9.0\webapps\examples\jsp

StuReg.jsp

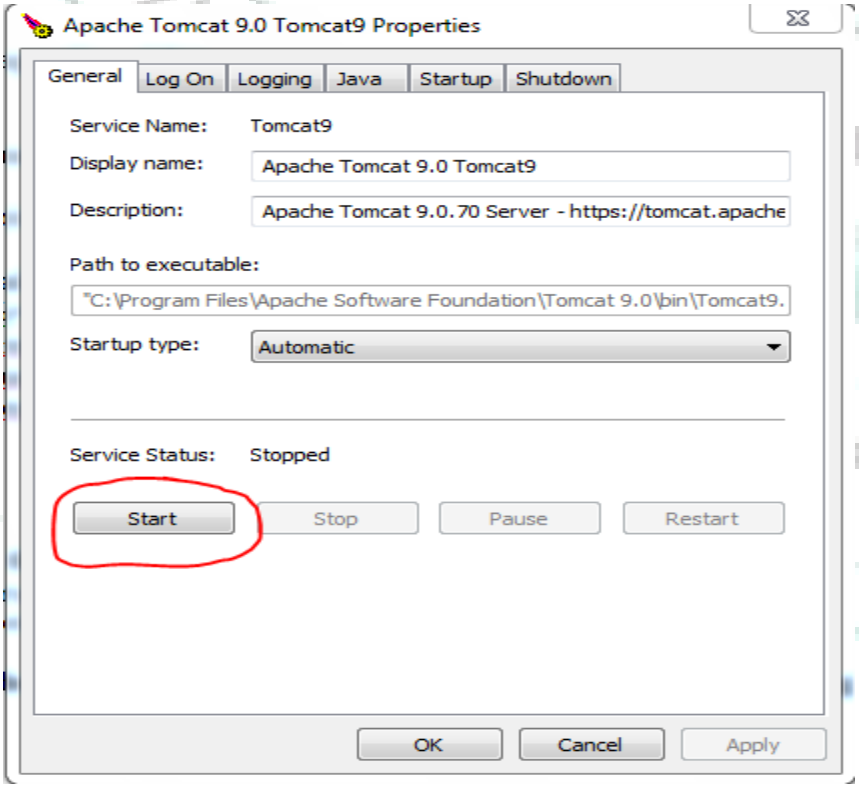
```
<%@ page language="java" %>
<html>
<head>
<title>JSP Response Page</title>
</head>
<body bgcolor="pink">
<center>
<h1>Student Details are</h1>
User Name is :<%=request.getParameter("un")%> </br>
Password is :<%=request.getParameter("pd")%></br>
Email is :<%=request.getParameter("eid")%></br>
Mobile No :<%=request.getParameter("mnb")%></br>
</center>
</body>
</html>
```

Step4: Create war file

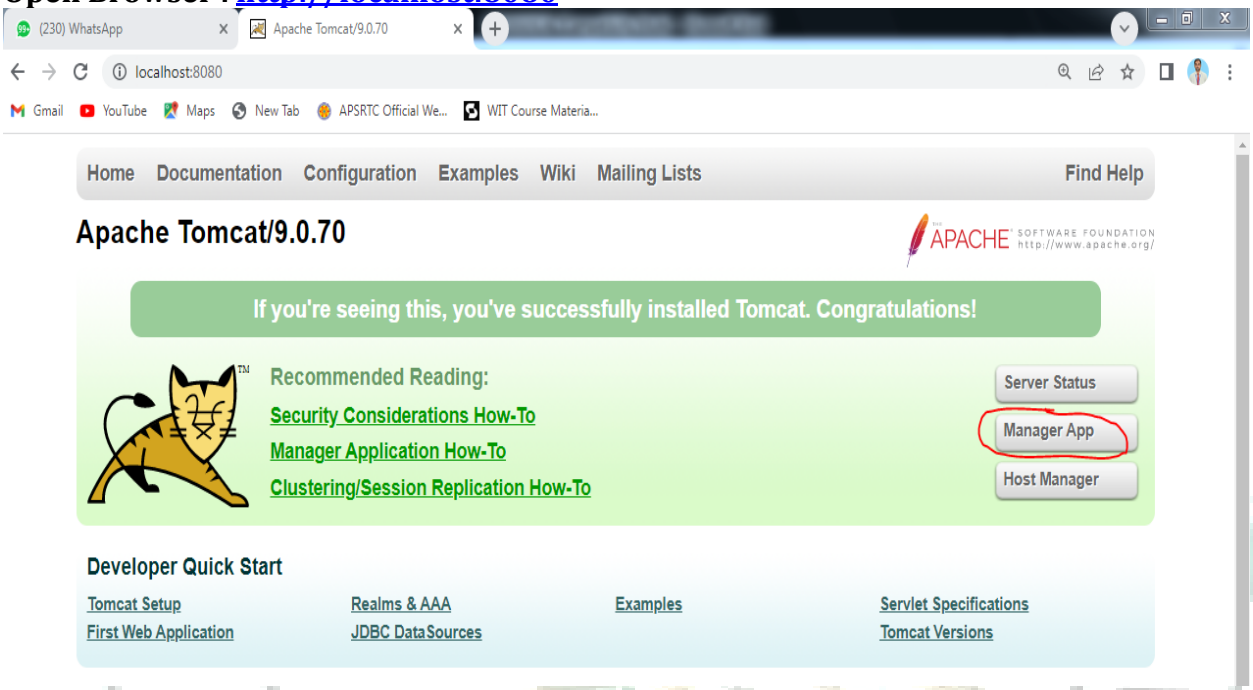
C:\Module-1>jar cvf Module-1.war .

Module-1.war created

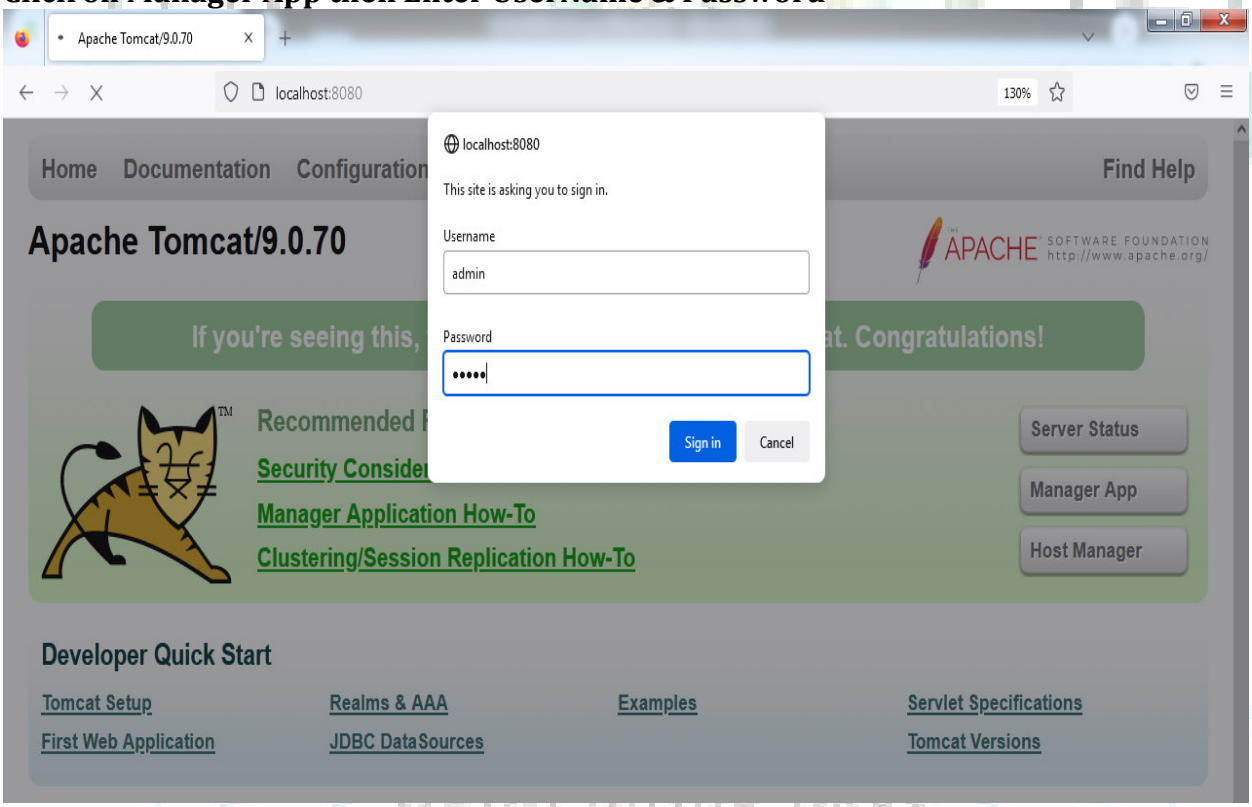
Step 5 : Deploy the Static web page using Tomcat Web Server by starting Tomcat Service



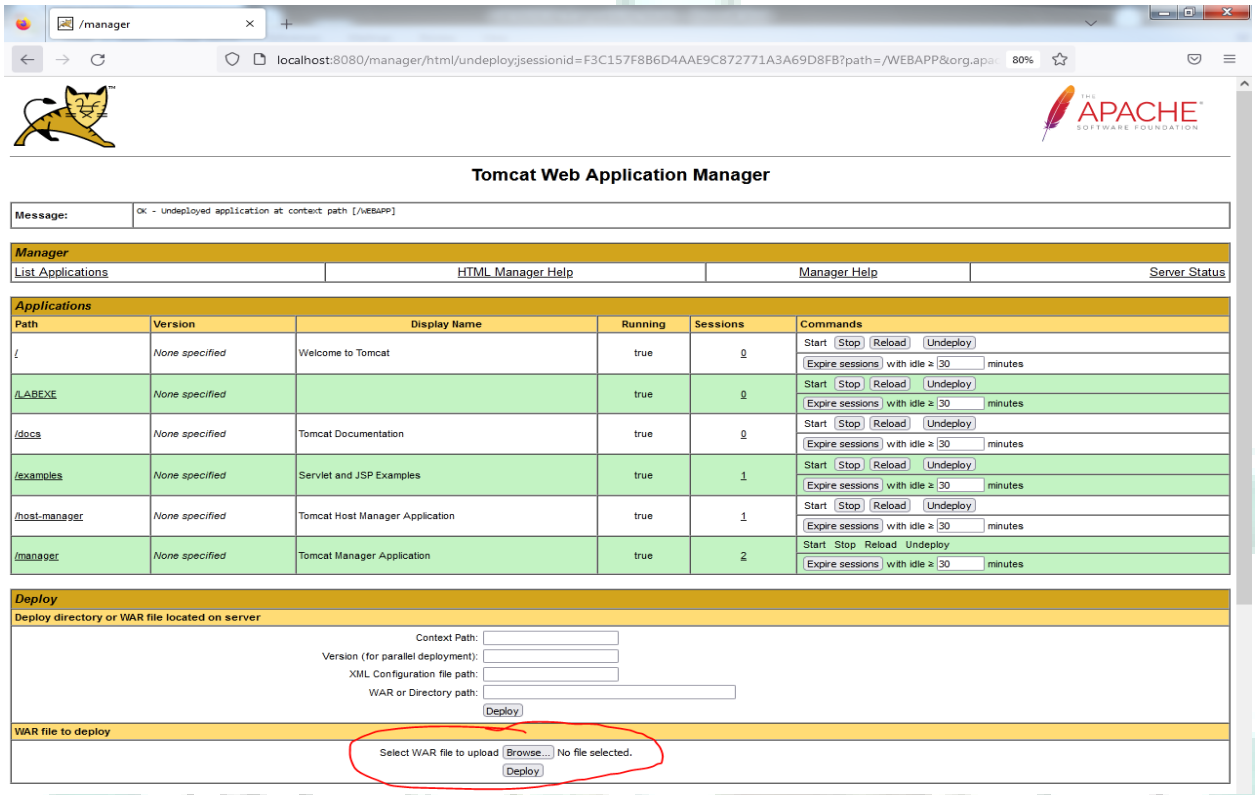
Open Browser : http://localhost:8080

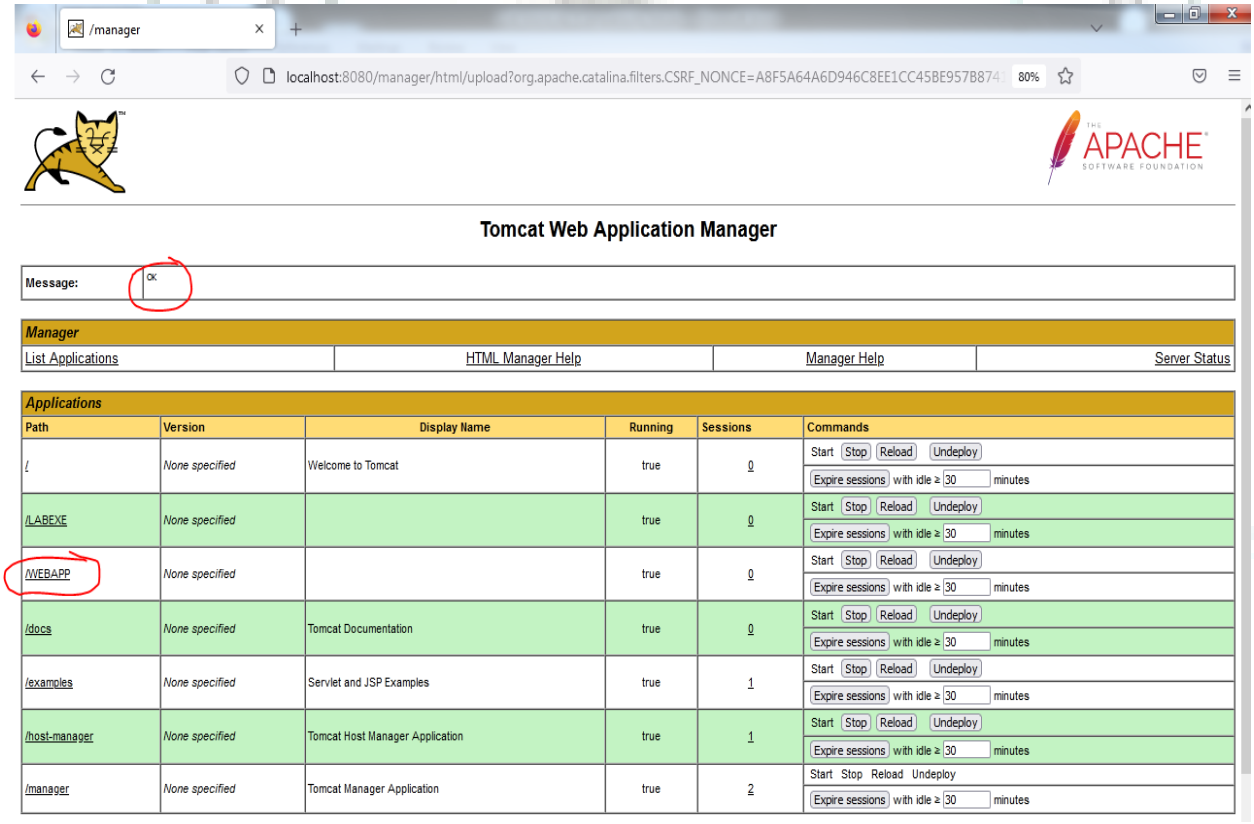


Click on Manager App then Enter UserName & Password

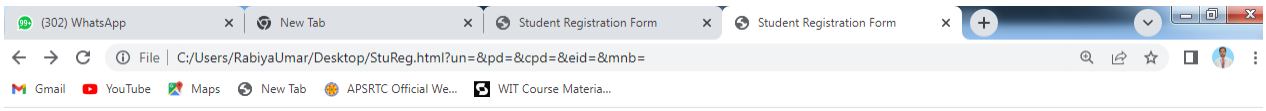


Browse the .war file & click on Deploy Button





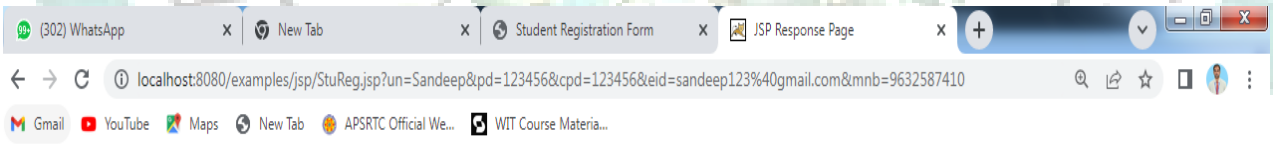
OPEN Browser & Enter URL: localhost:8080/WEBAPP/StuReg.html



Student Registration Form Using Table in HTML

User Name	Sandeep	(Max 25 Characters Allowed)
Password	••••••	(Max 25 Characters Allowed)
Confirm Password	••••••	(Max 25 Characters Allowed)
Email ID	sandeep123@gmail.com	
Mobile Number	9632587410	(10 Digits Allowed)

Clear Form Submit



Student Details are

User Name is :Sandeep
Password is :123456
Email is :sandeep123@gmail.com
Mobile No :9632587410

b) Develop a JavaScript program to validate the client-side user input data (Example: username, password, email, phone number).

Step-1: create StuReg.html

```
<html>
<head>
<title>Student Registration Form</title>
<link rel='stylesheet' href='stucss.css' type='text/css' />
<script src="stujs.js" type='text/javascript'></script>
</head>
<body>
<form name="stu">
<h2>Student Registration Form Using Table in HTML</h2>

<table align="center" cellpadding = "10">

<!---------------------- User Name ---------------------------------------->
<tr>
<td>User Name</td>
<td><input type="text" name="un" id="t1" class="tb"  maxlength="25" placeholder="User Name" />
(Max 25 Characters Allowed)
</td>
</tr>

<!------------------------ Password ------------------------------------->
<tr>
<td>Password</td>
<td><input type="password" name="pd" maxlength="25" placeholder="Password"/>
(Max 25 Characters Allowed)
</td>
</tr>

<!------------------------- Re-Password ------------------------------------>
<tr>
<td>Confirm Password</td>
<td><input type="password" name="cpd" maxlength="25" placeholder="Confirm Password"/>
(Max 25 Characters Allowed)
</td>
</tr>

<!-------------------------- Email ID ------------------------------------>
<tr>
<td>Email ID</td>
<td><input type="email" name="eid" maxlength="30" placeholder="Sample@gmail.com"/></td>
</tr>
```

```
<!-------------------------- Mobile Number ------------------------------------->
<tr>
<td>Mobile Number</td>
<td>
<input type="text" name="mnb" maxlength="10" placeholder="1234567890"/>
(10 Digits Allowed)
</td>
</tr>
 <tr>
<td colspan="2" align="center">
<input type="reset" value="Clear Form" id="res" class="btn" />
  <input type="submit" value="Validate" class="btn" onclick="registration()" />
</td>
</tr>
</table>

</form>

</body>
</html>
```

Step-2: create stujs.js

```
function registration()
  {
     var uname= stu.un.value;
     var pwd=stu.pd.value;
     var cpwd=stu.cpd.value;
     var email=stu.eid.value;
     var mno=stu.mnb.value;
     //email id expression code
     var pwd_expression = /^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-9])(?=.*?[#?!@$%^&*-])/;
     var letters = /^[A-Za-z]+$/;
     var filter = /^([a-zA-Z0-9_\.\-])+\@(([a-zA-Z0-9\-])+\.)+([a-zA-Z0-9]{2,4})+$/;
if(uname=='')
     {
       alert('Please enter your name');
     }
     else if(!letters.test(uname))
     {
       alert('Name field required only alphabet characters');
     }
else if(pwd=='')
     {
       alert('Please enter Password');
     }
     else if(cpwd=='')
     {
       alert('Enter Confirm Password');
     }
```

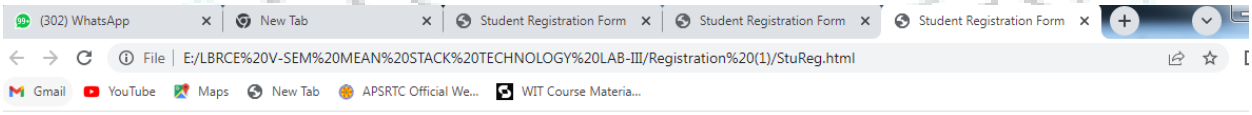
```
 else if(!pwd_expression.test(pwd))
      {
         alert ('Upper case, Lower case, Special character and Numeric letter are required in Password filed');
      }
      else if(pwd != cpwd)
      {
         alert ('Password not Matched');
      }
      else if(stu.pd.value.length < 6)
      {
         alert ('Password minimum length is 6');
      }
      else if(stu.pd.value.length > 12)
      {
         alert ('Password max length is 12');
      }
else if(email=='')
      {
         alert('Please enter your user email id');
      }
      else if (!filter.test(email))
      {
         alert('Invalid email');
      }
else if(mno=='')
      {
         alert('Please enter Mobile Number');
      }
else if(isNaN(mno))
{
 alert('Invalid Mobile Number');
}
else
{
alert('Thank You for Registration ');
}
}
```

Step-3: create stucss.css

```
h2{
  font-family: Sans-serif;
  font-size: 24px;
  font-style: normal;
  font-weight: bold;
  color: blue;
  text-align: center;
  text-decoration: underline
}
```

```
table{
  font-family: verdana;
  color:white;
  font-size: 16px;
  font-style: normal;
  font-weight: bold;
  background-color: #ff4242;
  border-collapse: collapse;
  border: 4px solid #000000;
  border-style: dashed;


}
table.inner{
  border: 50px
}

input[type=text], input[type=email], input[type=number]{
  width: 50%;
  padding: 6px 12px;
  margin: 5px 0;
  box-sizing: border-box;
}

input[type=submit], input[type=reset]{
  width: 15%;
  padding: 8px 12px;
  margin: 5px 0;
  box-sizing: border-box;
}
```

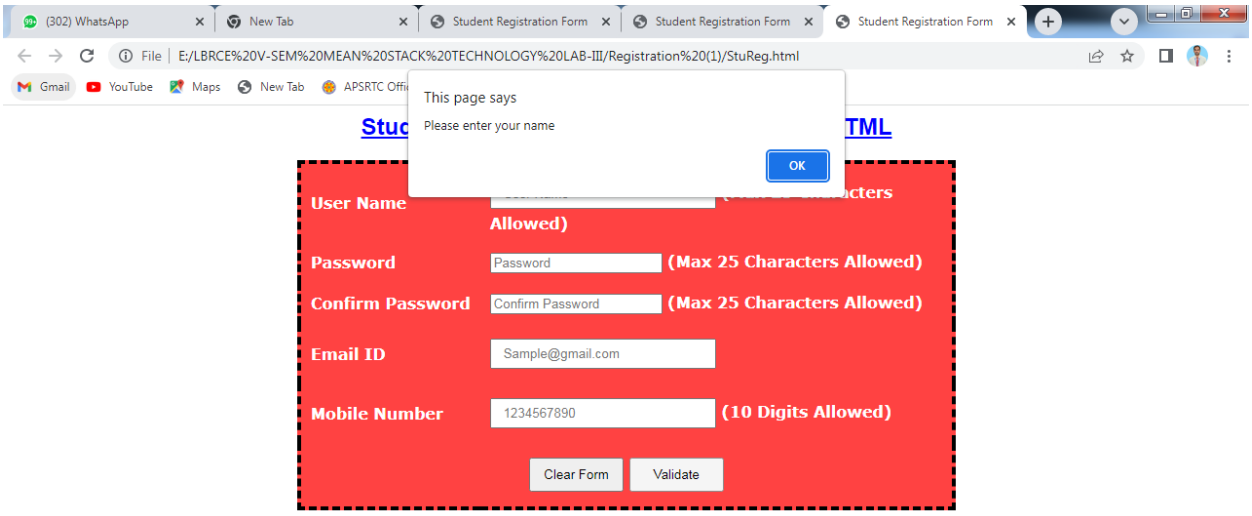
OUTPUT:

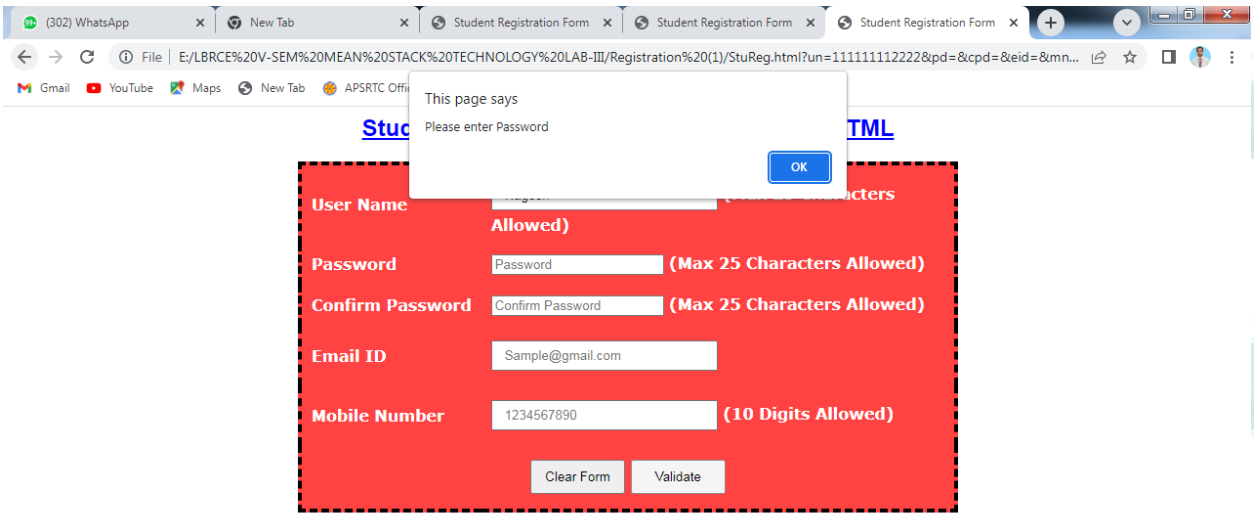


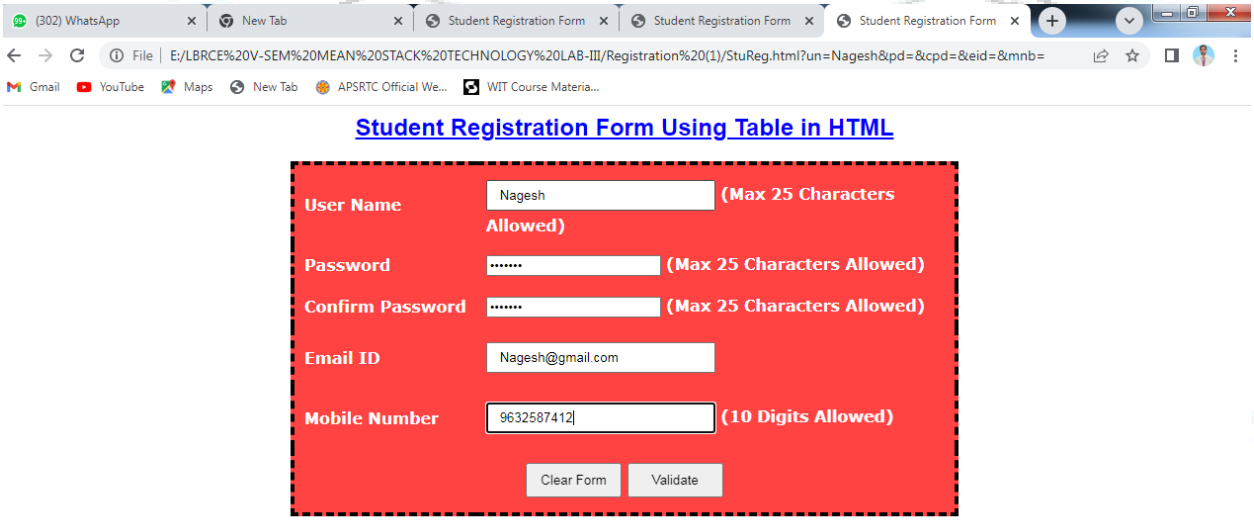
Student Registration Form Using Table in HTML

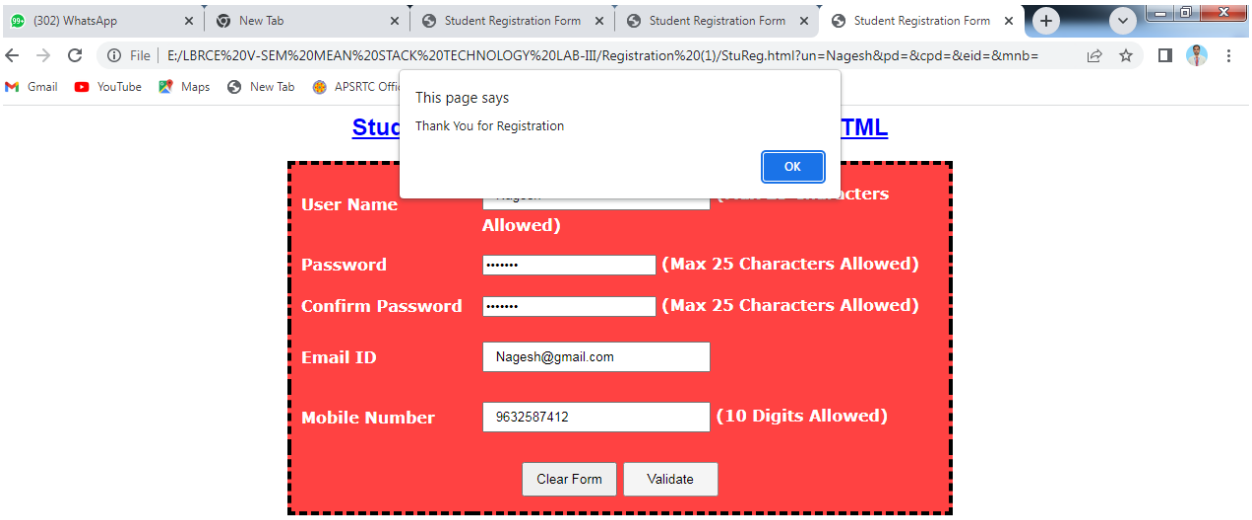
User Name	User Name	**(Max 25 Characters Allowed)**
Password	Password	**(Max 25 Characters Allowed)**
Confirm Password	Confirm Password	**(Max 25 Characters Allowed)**
Email ID	Sample@gmail.com	
Mobile Number	1234567890	**(10 Digits Allowed)**

Clear Form Validate


(302) WhatsApp
New Tab
Student Registration Form
Student Registration Form
Student Registration Form
File | E:/LBRCE%20V-SEM%20MEAN%20STACK%20TECHNOLOGY%20LAB-III/Registration%20(1)/StuReg.html
Gmail
YouTube
Maps
New Tab
This page says
Please enter your name
OK
TML
User Name
Allowed)
Password
Password
(Max 25 Characters Allowed)
Confirm Password
Confirm Password
(Max 25 Characters Allowed)
Email ID
Sample@gmail.com
Mobile Number
1234567890
(10 Digits Allowed)
Clear Form
Validate


(302) WhatsApp
New Tab
Student Registration Form
Student Registration Form
Student Registration Form
File | E:/LBRCE%20V-SEM%20MEAN%20STACK%20TECHNOLOGY%20LAB-III/Registration%20(1)/StuReg.html?un=111111112222&pd=&cpd=&eid=&mn...
Gmail
YouTube
Maps
New Tab
This page says
Please enter Password
OK
TML
User Name
Allowed)
Password
Password
(Max 25 Characters Allowed)
Confirm Password
Confirm Password
(Max 25 Characters Allowed)
Email ID
Sample@gmail.com
Mobile Number
1234567890
(10 Digits Allowed)
Clear Form
Validate


(302) WhatsApp
New Tab
Student Registration Form
Student Registration Form
Student Registration Form
File | E:/LBRCE%20V-SEM%20MEAN%20STACK%20TECHNOLOGY%20LAB-III/Registration%20(1)/StuReg.html?un=Nagesh&pd=&cpd=&eid=&mnb=
Gmail
YouTube
Maps
New Tab
APSRTC Official We...
WIT Course Materia...
Student Registration Form Using Table in HTML
User Name
Nagesh
(Max 25 Characters Allowed)
Password
(Max 25 Characters Allowed)
Confirm Password
(Max 25 Characters Allowed)
Email ID
Nagesh@gmail.com
Mobile Number
9632587412
(10 Digits Allowed)
Clear Form
Validate


(302) WhatsApp
New Tab
Student Registration Form
Student Registration Form
Student Registration Form
File | E:/LBRCE%20V-SEM%20MEAN%20STACK%20TECHNOLOGY%20LAB-III/Registration%20(1)/StuReg.html?un=Nagesh&pd=&cpd=&eid=&mnb=
Gmail
YouTube
Maps
New Tab
This page says
Thank You for Registration
OK
TML
User Name
Allowed)
Password
(Max 25 Characters Allowed)
Confirm Password
(Max 25 Characters Allowed)
Email ID
Nagesh@gmail.com
Mobile Number
9632587412
(10 Digits Allowed)
Clear Form
Validate

LAKIREDDY B
ENGINEERING
MYLAVARAM
TRUTH ALWAYS TRIUMPHS
HARD WORK PAYS

Module 2:

a) Create a XMLHttpRequest with a callback function and retrieve data from a TXT file by using AJAX.

Step 1: Create **sample.text** file

Welcome to Server Side Scripting Lab

AJAX Programs

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)

Step 2: Create **2a.html** file

```
<html>
<body>
<div id="disp">
<h2>The XMLHttpRequest Object</h2>
<button type="button" onclick="getTextFile()">Click Here to Get the Data From Text
File</button>
</div>
<script>
function getTextFile()
 {
// Creating the XMLHttpRequest object
  const xhttp = new XMLHttpRequest();
// Instantiating the request object
  xhttp.open("GET", "sample.txt");
xhttp.onreadystatechange = function()
      {
       // Check if the request is compete and was successful
     if(this.readyState === 4 && this.status === 200)
```

```
{
        // Inserting the response from server into an HTML element
  document.getElementById("disp").innerHTML = this.responseText;
    }
  };
  xhttp.send(); // Sending the request to the server
}
</script>
</body>
</html>
```

Step 3: Deploy these files into Apache Tomcat Server

Create war file to deploy

```
E:\VI-SEM SSS LAB\LABPROGRAMS>jar cvf LABPROGRAMS.WAR .
added manifest
adding: 2a.html(in = 761) (out= 404)(deflated 46%)
adding: sample.txt(in = 301) (out= 208)(deflated 30%)
```

Deploy this war file into Apache Tomcat Server

Step 4: Open browser then type URL

http://localhost:8080/LABPROGRAMS/s2.html

The XMLHttpRequest Object

Click Here to Get the Data From Text File

b) Create an XMLHttpRequest to retrieve data from an XML file and display the data in an HTML table by using AJAX.

Step-1: Create a data.xml file

data.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
        <book category="web">
            <title lang="en">Learning XML</title>
            <author>Erik T. Ray</author>
            <year>2008</year>
            <price>$30.00</price>
         </book>
        <book category="DataBase">
            <title lang="en">Data Base Management System</title>
            <author>Robert Sebesta</author>
```

```
        <year>2010</year>
          <price>$49.99</price>
    </book>
    <book category="Programming">
        <title lang="en">JAVA Programming</title>
         <author>James Gosling</author>
        <year>1992</year>
        <price>$52.95</price>
    </book>
</bookstore>
```

Step-2: Create a data.html file

data.html

```
<html>
<style>
table,th,td {
  border : 1px solid black;
  border-collapse: collapse;
}
th,td {
  padding: 5px;
}
</style>
<body>

<h1>The XMLHttpRequest Object of Books Details</h1>

<button type="button" onclick="loadDoc()">Get Books Details</button>
<br><br>
<table id="disp"></table>

<script>
```

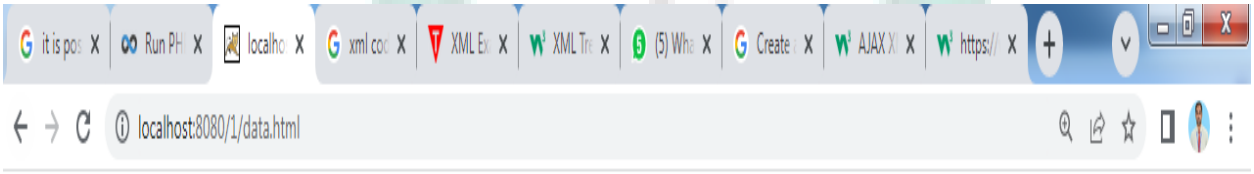
```
function loadDoc()
 {
 var xhttp = new XMLHttpRequest();
 xhttp.onreadystatechange = function()
  {
  if (this.readyState == 4 && this.status == 200)
  {
    myFunction(this);
  }
 };
 xhttp.open("GET", "data.xml", true);
 xhttp.send();
}
function myFunction(xml)
 {
 var i;
 var xmlDoc = xml.responseXML;
 var table="<tr><th>Title</th><th>Author</th><th>Year</th><th>Price</th></tr>";
 var x = xmlDoc.getElementsByTagName("book");
 for (i = 0; i <x.length; i++)
  {
  table += "<tr><td>" +
  x[i].getElementsByTagName("title")[0].childNodes[0].nodeValue +
  "</td><td>" +
  x[i].getElementsByTagName("author")[0].childNodes[0].nodeValue +
  "</td><td>" +
  x[i].getElementsByTagName("year")[0].childNodes[0].nodeValue +
  "</td><td>" +
  x[i].getElementsByTagName("price")[0].childNodes[0].nodeValue +
  "</td></tr>";
 }
 document.getElementById("disp").innerHTML = table;
}
</script>

</body>
</html>
```

Step-3: create war file

C:\2b>**jar cvf 2b.war .**

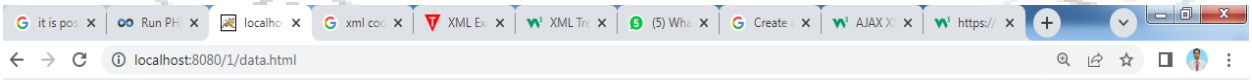
Step-4: Deploy the war file into the apache tomcat server

Step-5: enter URL: http://localhost:8080/2b/data.html



The XMLHttpRequest Object of Books Details

Get Books Details



The XMLHttpRequest Object of Books Details

Get Books Details

Title	Author	Year	Price
Learning XML	Erik T. Ray	2008	$30.00
Data Base Management System	Robert Sebesta	2010	$49.99
JAVA Programming	James Gosling	1992	$52.95

Module 3:

a) Develop AJAX application to demonstrate how a web page can communicate with a web server while a user type characters in an input field.

- AJAX stands for Asynchronous JavaScript and XML. AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS, and JavaScript.
- Conventional web applications transmit information to and from the server using synchronous requests. It means you fill out a form, hit submit, and get directed to a new page with new information from the server.
- With AJAX, when submit is pressed, JavaScript will make a request to the server, interpret the results and update the current screen. In the purest sense, the user would never know that anything was even transmitted to the server.

The following example will demonstrate how a web page can communicate with a web server while a user types characters in an input field:

Start typing a name in the input field below:

First name:

Suggestions:

Explanation

In the example above, when a user types a character in the input field, a function called "showing()" is executed.

The onkeyup event triggers the function.

Here is the HTML code:

```
<html>
<head>
<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>
</head>
<body>

<p><b>Start typing a name in the input field below:</b></p>
<form action="">
 <label for="fname">First name:</label>
 <input type="text" id="fname" name="fname" onkeyup="showHint(this.value)">
</form>
<p>Suggestions: <span id="txtHint"></span></p>
</body>
</html>
```

Code explanation:

First, check if the input field is empty (str.length == 0). If it is, clear the content of the txtHint placeholder and exit the function.

Do the following, if the input field is not empty:

- Create an XMLHttpRequest object
- Create the function to be executed when the server response is ready
- Send the request off to a PHP file (gethint.php) on the server
- Notice that the q parameter is added to the url (gethint.php?q="+str)
- And the str variable holds the content of the input field

The PHP File - "gethint.php."

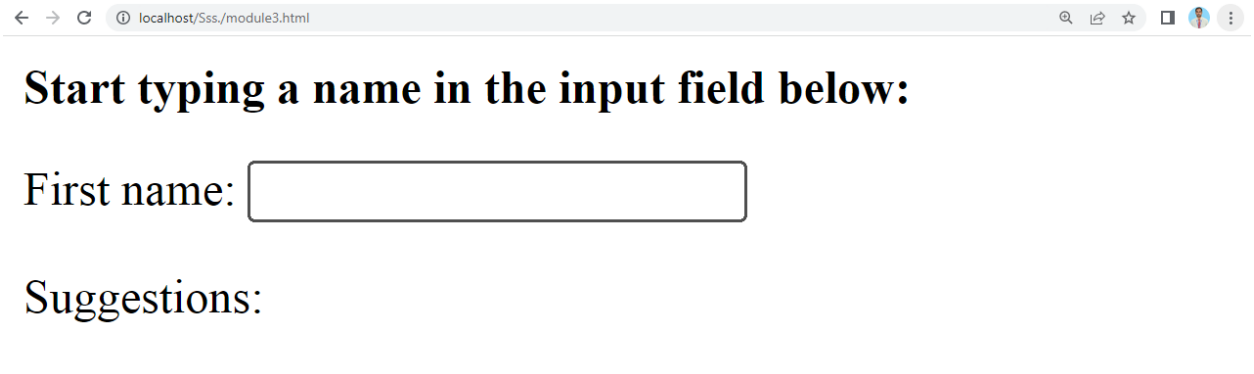
The PHP file checks an array of names, and returns the corresponding name(s) to the browser:

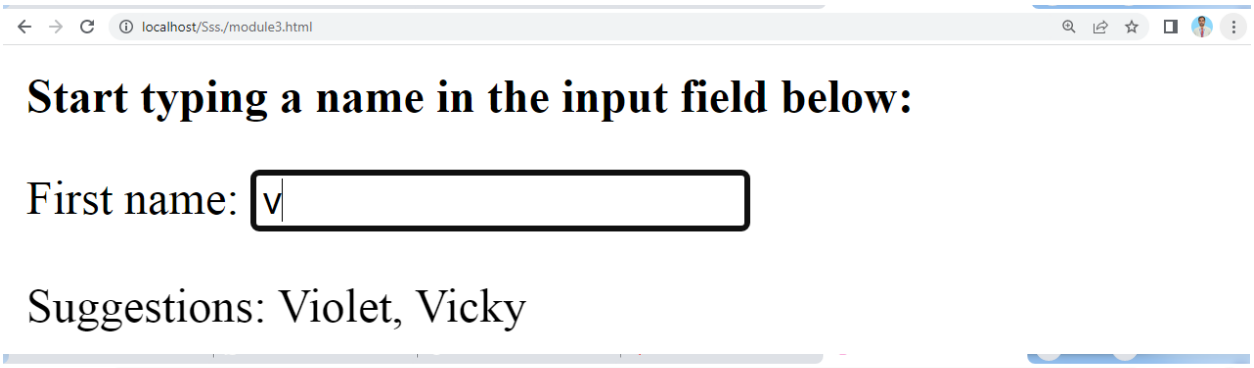
```php
<?php
// Array with names
$a[] = "Anna";
$a[] = "Brittany";
$a[] = "Cinderella";
$a[] = "Diana";
$a[] = "Eva";
$a[] = "Fiona";
$a[] = "Gunda";
$a[] = "Hege";
$a[] = "Inga";
$a[] = "Johanna";
$a[] = "Kitty";
$a[] = "Linda";
$a[] = "Nina";
$a[] = "Ophelia";
$a[] = "Petunia";
$a[] = "Amanda";
$a[] = "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";

// fetch q parameter from URL
$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";
      }
    }
  }
}

// It results in "no suggestion" if no hint was found or output correct values
echo $hint === "" ? "no suggestion" : $hint;
?>
```

Output:

both html & php files are placed into “www” folder and open html file through localhost





b) Develop a simple web application to demonstrate how a web page can fetch information from a database with AJAX.

3b.html

```
<!DOCTYPE html>
<html>
<head>
<title>AJAX Database Example</title>
<script

src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/jquery.min.js"></scrip

t>
<script>
$(document).ready(function(){
  $.ajax({
    url: "get_data.php", // Replace with your PHP file
    type: "GET",
    success: function(data){
      $("#result").html(data);
    }
  });
});
</script>
</head>
```

```
<body>

<h2>Data from Database</h2>

<div id="result"></div>

</body>
</html>
```

get_data.php

```
<?php

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "sss";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT * FROM stud";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
  // Output data in HTML format
  echo "<table><tr><th>ID</th><th>Name</th></tr>"; // Example table headers
  while($row = $result->fetch_assoc()) {
    echo "<tr><td>" . $row["regno"] . "</td><td>" . $row["name"] . "</td></tr>";
  }
  echo "</table>";
} else {
  echo "0 results";
}

$conn->close();
?>
```

OUTPUT:

localhost/3/3b.html

Data from Database

ID	Name
101	sandeep
101	sandeep
102	harish

Module 4:
a) Develop a PHP program to illustrate the PHP Form handling by using GET and POST methods.

- The PHP script needs following variables to read the form data and perform the operations on it.

Superglobal Array	Description
$_GET	Contains a list of all the field names and values sent by a form using the get method
$_POST	Contains a list of all the field names and values sent by a form using the post method
$_REQUEST	Contains the values of both the $_GET and $_POST arrays combined, along with the values of the $_COOKIE superglobal array

When to use method="get"?

- The variable names and values will be visible in URL if HTML forms submitted by the GET method.
- The GET method is restricted to send up to 2048 characters only.
- When you submit sensitive information like passwords then should not use this method.
- GET method can't be used, to send binary data like images and Word documents.
- GET method data can be accessed using PHP QUERY_STRING environment variable.PHP $_GET associative array is used to access all the sent information by GET method.

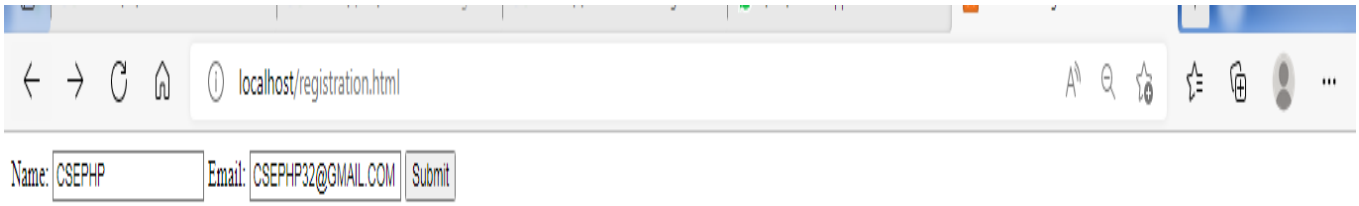
registration.html

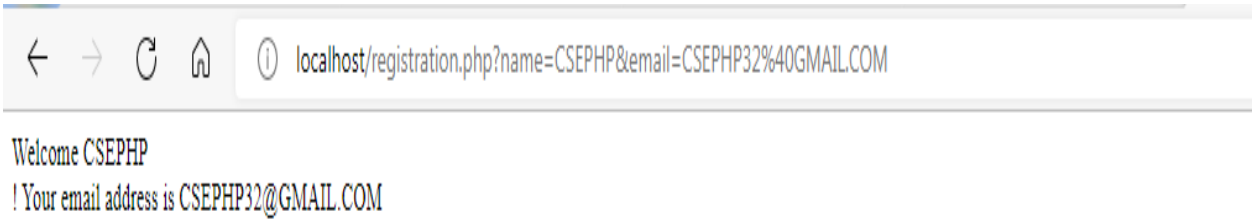
```
html>
 <body>
<form action=“get.php"method="get">
<h1>$_GET Variable Example Program</h1>
  Name: <input type="text" name="name">
</br>
  Email: <input type="text" name="email">
<input type="submit">
 </form>
 </body>
</html>
```

registration.php

```
<html>
<body>
Welcome <?php echo $_GET["name"]; ?>
</br>
! Your email address is <?php echo $_GET["email"]; ?>
</body>
</html>
```

OUTPUT:





When to use method="POST"?

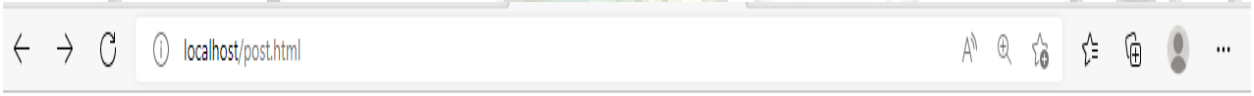
- **The POST method does not have any restriction on data size to be sent.**
- **The POST method can be used to send ASCII as well as binary data.**
- **The data sent by POST method goes through HTTP header, so security depends on HTTP protocol. By using Secure HTTP, you can make sure that your information is secure.**
- **PHP $_POST associative array is used to access all the sent information by POST method. Variables are not visible in the URL so users can't bookmark your page.**

post.html

```
<html>
 <body> <center>
<form action="post.php"method="post">
<h1>$_POST Variable Example Program</h1>
  Name: <input type="text" name="name">
</br>
  Email: <input type="text" name="email">
<input type="submit">
 </form>  </body> </center></html>
```

post.php

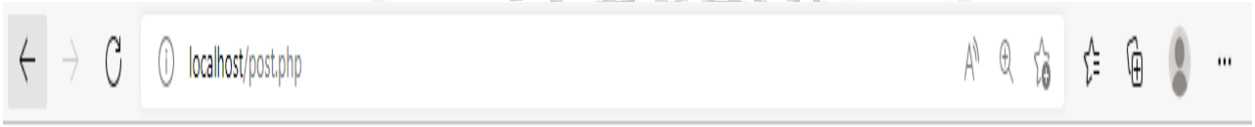
```
<html>
<body>
Welcome <?php echo $_POST["name"]; ?>
</br>
! Your email address is <?php echo $_POST["email"]; ?>
</body>
</html>
```



$_POST Variable Example Program

Name: LBRCECSE

Email: lbrecseinfo@lbrce.ac.in Submit



Welcome LBRCECSE

! Your email address is lbrecseinfo@lbrce.ac.in

b) Develop a PHP program to demonstrate the importance of include() and require() functions.

- **PHP allows us to create various elements and functions, which are used several times in many pages.**
- **It takes much time to script these functions in multiple pages.**
- **Therefore, use the concept of file inclusion that helps to include files in various programs and saves the effort of writing code multiple times.**
- **PHP allows you to include file so that a page content can be reused many times.**
- **It is very helpful to include files when you want to apply the same HTML or PHP code to multiple pages of a website.**

There are two ways to include file in PHP.

1. Include **2. require**

- **Both include and require are identical to each other, except failure.**
- **include only generates a warning, i.e., E_WARNING, and**

  continue the execution of the script.
- **require generates a fatal error, i.e., E_COMPILE_ERROR, and**

  stop the execution of the script.

<u>Advantage</u>

- **Code Reusability: By the help of include and require construct, we can reuse HTML code or PHP script in many PHP scripts.**
- **Easy editable: If we want to change anything in webpages, edit the source file included in all webpage rather than editing in all the files separately.**

PHP include:

- **PHP include is used to include a file on the basis of given path.**

Syntax

- **There are two syntaxes available for include:**

include 'filename '; Or include ('filename');

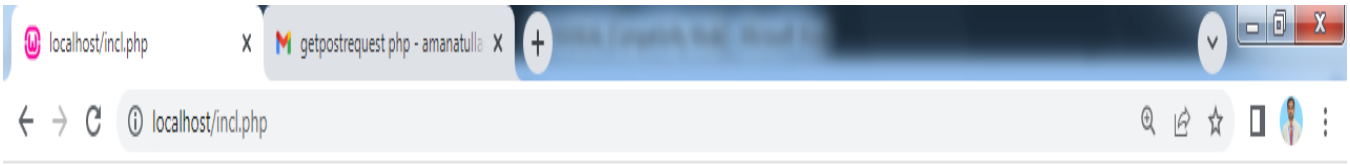
urls.html

```
<html>
<a href="http://www.google.com">Google</a> |
<a href="http://www.yahoo.com">Yahoo</a> |
<a href="https://www.flipkart.com/">FlipKart</a> |
<a href="https://www.amazon.in/">Amazon</a> |
<a href="https://www.bigbasket.com/">BigBasket</a> |
</html>
```

Incl.php

```
<h1> The Following URLs are Reglarling using for searching & Buying  </h1>
<?php  include("urls.html");
echo "Sample include" ?>
</br>
<?php  require("urls.html");
echo "Sample require" ?>
```

OUTPUT:



The Following URLs are Reglarling using for searching & Buying

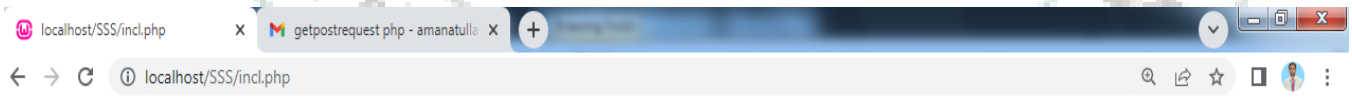
Google | Yahoo | FlipKart | Amazon | BigBasket | Sample include
Google | Yahoo | FlipKart | Amazon | BigBasket | Sample require

details.php

```
<?php
$year="III";
$branch="CSE";
$colg="LBRCE";
?>
```

incl.php

```
<?php include "details.php";
echo "I am a student of $year , $branch , $colg .";
?>
<?php require "details.php";
echo "I am a student of $year , $branch , $colg .";
?>
```

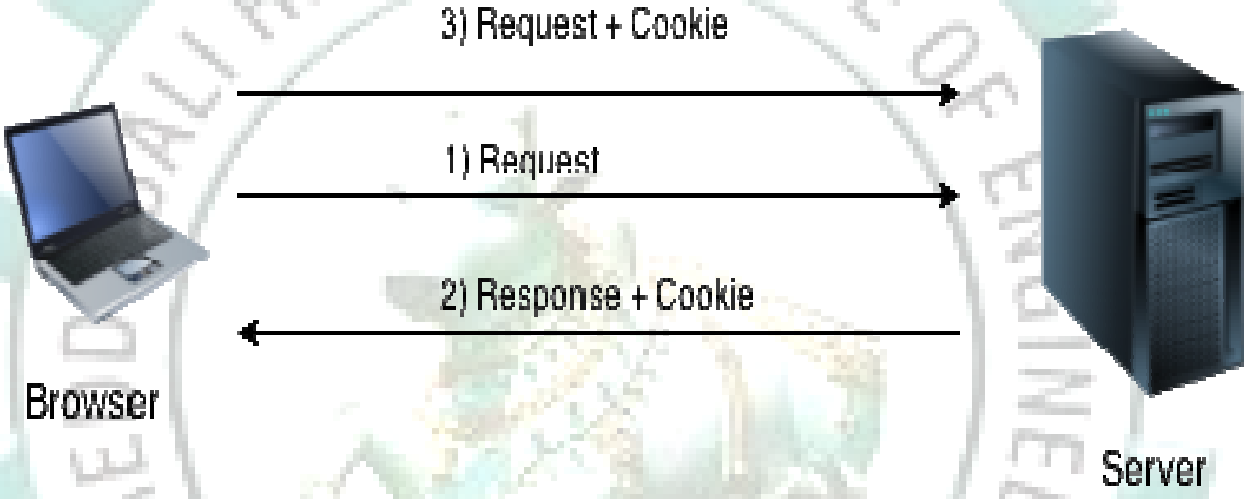


I am a student of III , CSE , LBRCE .I am a student of III , CSE , LBRCE .

Module 5:

a) **Develop a PHP program to manage the state information about the user by using PHP Cookie.**

- PHP cookie is a small piece of information which is stored at client browser. It is used to recognize the user.
- Cookie is created at server side and saved to client browser. Each time when client sends request to the server, cookie is embedded with request. Such way, cookie can be received at the server side.



Create Cookies with PHP

- The **setcookie()** function used to create a cookie.
- A single domain has maximum 20 cookies.
- A single cookie can not exceed 4 kilobytes in size.
- PHP **setcookie()** function must be called before <html> tag.
- The **setcookie()** function has up to six arguments.

setcookie() function

syntax:

setcookie(name, value, expire, path, domain, security);

Name : This is name of the cookie and it is used to retrieve cookie.
Value : information or value to be stored in cookie.
Expire : used to set time of expiry of cookie. If not set expiry the cookie automatically expire when browser closed.
Path : "/" means the cookie available in entire website or all directories. or we can specify the directories for valid cookie.
Domain : used to specify the domain name.

Security : used to security purpose. set 1 or 0, 1 for secure transmission using HTTPS and 0 for sent by simple HTTP.

Example of create cookies

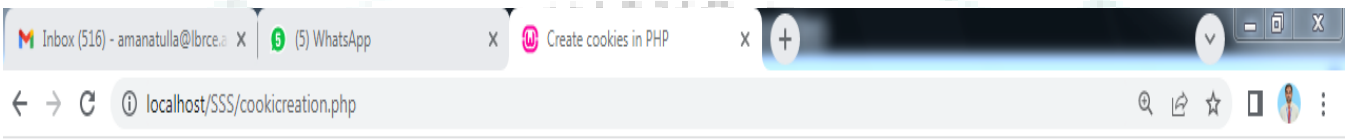
setcookie("college", "lbrce", time() + 3600, "/", "", 0);

Here, we create cookie named "**college**" with value "**lbrce**". The cookie will expire after 1 hour. The "/" means cookie available in all directories.

Example:

```
<?php
setcookie("branch","CSE", time() + 3600, "/", "", 0);
setcookie("code", "05", time() + 3600, "/", "", 0); ?>
<HTML>
 <head>
<title>Create cookies in PHP</title>
 </head>
<body>
<?php echo "The cookies created for branch and code."; ?>
</body>
 </HTML>
```

OUTPUT:



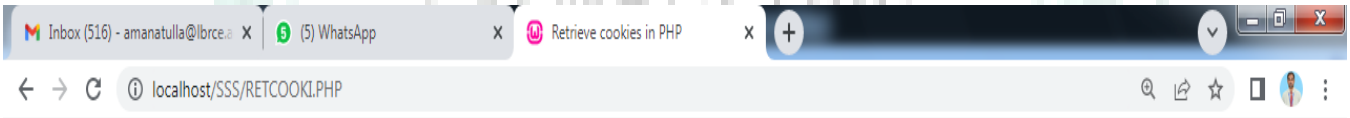
The cookies created for branch and code.

Retrieve / Access cookies :

The cookies retrieve using the $_COOKIE[] global variable.
The isset() function used to check if a cookie is set or not.

```
<HTML>
<head>
<title>Retrieve cookies in PHP</title>
</head>
<body>
<?php
if(isset($_COOKIE["branch"]) && isset($_COOKIE["code"]))
{
echo " The Branch name = " .$_COOKIE["branch"]. "<br/>";
echo "The Branch code = ". $_COOKIE["code"];
}
else
{
echo "Sorry !! cookies is not set.";
}
?>
</body>
</HTML>
```

OUTPUT:



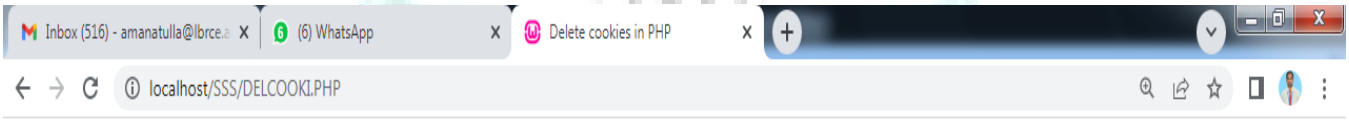
The Branch name = CSE
The Branch code = 05

Delete / Destroy cookies:

A cookie deleted by calling the same setcookie() function with the cookie name and any value (such as an empty string) however this time you need the set the expiration date in the past.

```
<?php
setcookie("branch","", time() - 3600, "/", "", 0);
setcookie("code", "", time() - 3600, "/", "", 0);
?>
<HTML>
<head>
<title>Delete cookies in PHP</title>
</head>
<body>
<?php
echo "The cookies Deleted for".$_COOKIE['branch'];
?>
</body>
</HTML>
```

OUTPUT:



The cookies Deleted forCSE

Example:

links.html

```
<html>
  <body>
                <a href="login.php">Login</a> |
                <a href="profile.php">View Profile</a> |
                <a href="logout.php">Logout</a>
                <hr>

  </body>
  </html>
```

login.php

```
<html>
      <head>
                <script>
                        function myfun()
                        {
                                var un = document.login.uname.value;
                                var ps = document.login.pwd.value;

                                if(un=="")
                                {
                                        alert("please enter username");
                                        document.login.uname.focus();
                                        return false;
                                }
                                if(ps=="")
                                {
                                        alert("please enter password");
                                        document.login.pwd.focus();
                                        return false;
                                }

                                return true;
                        }
                </script>

      </head>

      <body>

            <?php
                        include("links.html");
            ?>
```

```
                <form name="login" method="post" onsubmit="return myfun()"
action="welcome.php">

                        Enter User Name: <input type="text" name="uname"/><br>
                        Enter Password: <input type="password" name="pwd"/><br>

                        <input type="submit"/><input type="reset"/>
                </form>

        </body> </html>
```

welcome.php

```
<?php
        include("links.html");

        $un = $_POST["uname"];

        $ps = $_POST["pwd"];


        if($un=="cse" && $ps=="cse05")
        {

                setcookie("mycookie", $un, time()+3600*24, "/");

                echo "Welcome to ".$un;
        }
        else
        {
                ?>

                <script>
                        alert("Invalid credentials");
                </script>

                <?php
                include("login1.php");
        }

        ?>
```

profile.php

```
<?php


  if(isset($_COOKIE['mycookie'])) {
```

```
    include("links.html");

    echo "this is your profile, your name is:".$_COOKIE['mycookie'];
  }
  else
  {
                ?>

                <script>
                        alert("please login first");
                </script>

                <?php
                include("login.php");
  }

?>
```

login1.php

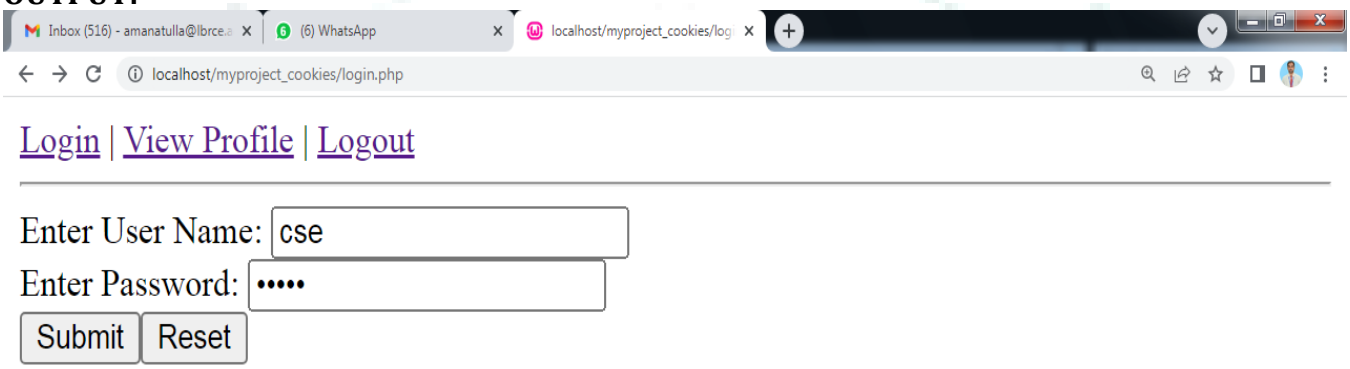
```
<html>
        <head>
                <script>
                        function myfun()
                        {
                                var un = document.login.uname.value;
                                var ps = document.login.pwd.value;

                                if(un=="")
                                {
                                        alert("please enter username");
                                        document.login.uname.focus();
                                        return false;
                                }
                                if(ps=="")
                                {
                                        alert("please enter password");
                                        document.login.pwd.focus();
                                        return false;
                                }
                                return true;
                        }
                </script>
```

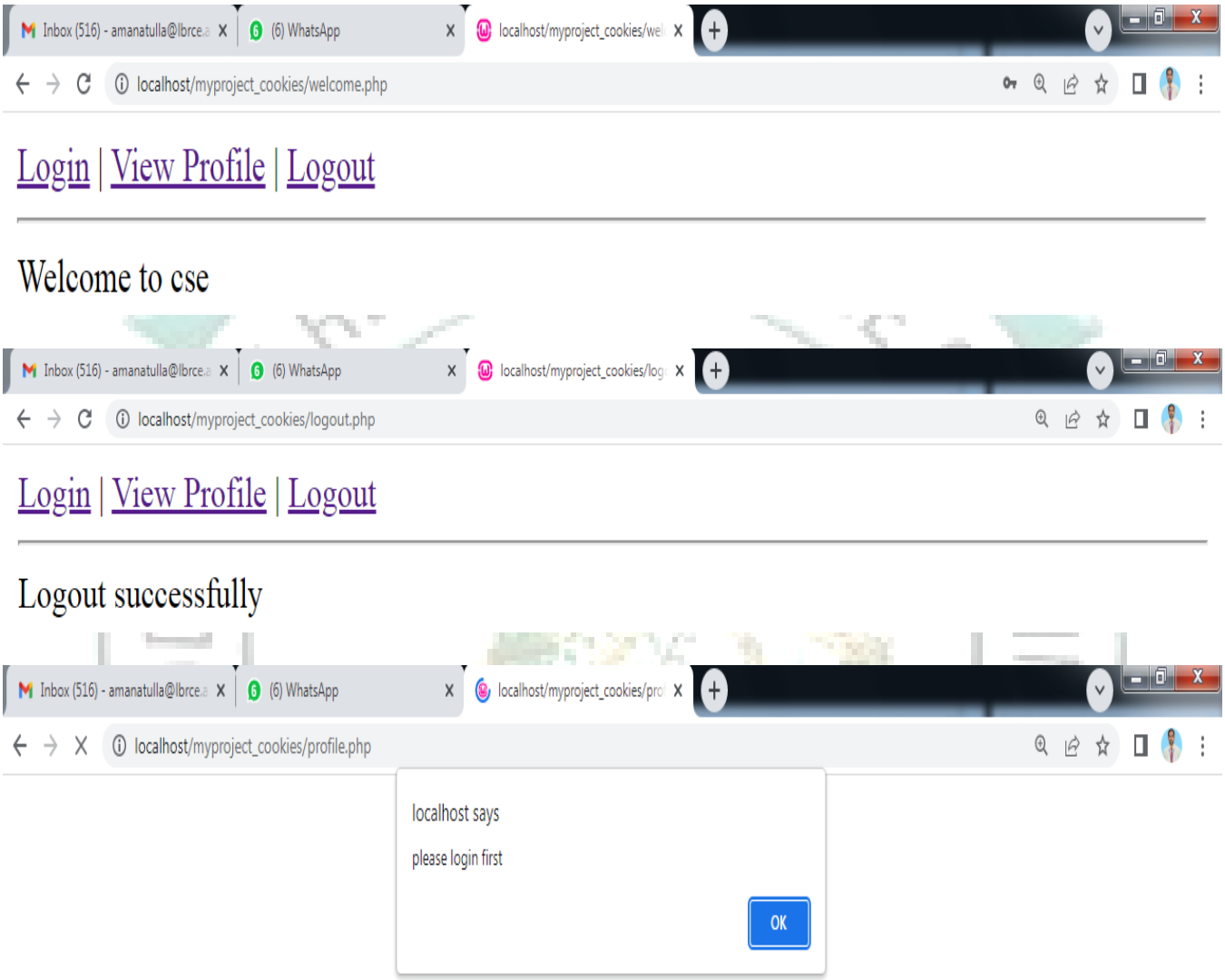
```
        </head>

        <body>

                <form name="login" method="post" onsubmit="return myfun()"
action="welcome.php">

                        Enter User Name: <input type="text" name="uname"/><br>
                        Enter Password: <input type="password" name="pwd"/><br>

                        <input type="submit"/><input type="reset"/>
                </form>

        </body>

</html>
```

logout.php

```
<?php

include("links.html");

setcookie("mycookie", "", time()-3600, "/");

echo "Logout successfully";

?>
```

OUTPUT:





b) Develop a PHP program to manage the state information about the user by using PHP Session.

To store information accessible across web pages, we use sessions. **Session** is not stored on the user browser like Cookies, hence it is a more secure option.

As we know HTTP is a stateless protocol, if a user visits a webpage and perform some action, there is no way to remember what he did when the user navigates to the next webpage.

Let's take a practical example, when you log into your facebook account, by providing your email address and password, until and unless you logout, the web application remembers who you are and display what your friends are posting and liking on your News Feed, you can update your profile, send someone message, join a group etc, this is accomplished by **Session**.

When a user logs into their account on any web application, a session is created for them, and in the session their *username* or *userid* or some other unique identifier is stored, which is then used on the consecutive webpages to show information specific to that user. On logout, the session is destroyed.

Session is not limited by any size limit, you can store any information in the session, irrespective of its size.

Real world Use of Session

1. Web applications which require a user to login, use session to store user information, so that on every webpage related information can be displayed to the user.
2. In eCommerce websotes, shopping cart is geberally saved as part of session.

Start a Session in PHP

In PHP we can start a session by using the session_start() function. And data is stored in the session using session variable, which can be assigned different values using global variable $_SESSION

In simpler words, using the function session_start() we initialize the session, in which we can store information using the session variable $_SESSION.

Let's take an example; below we have a webpage with Php file named

start_session_page.php

```
<?php
session_start();      // start the session

$_SESSION["username"] = "cse";          // set the session variable

$_SESSION["userid"] = "501";  // set the session variable

?>
<html>
  <body>
   <?php
          echo "Session variable is set.";
  ?>
 <center>
<h1>
      <a href="second_page.php">Go to Second Page</a>
 </h1>
</center>
  </body> </html>
```

NOTE: The function session_start() should be the first statement of the page, before any HTML tag.

Getting PHP Session Variable Values

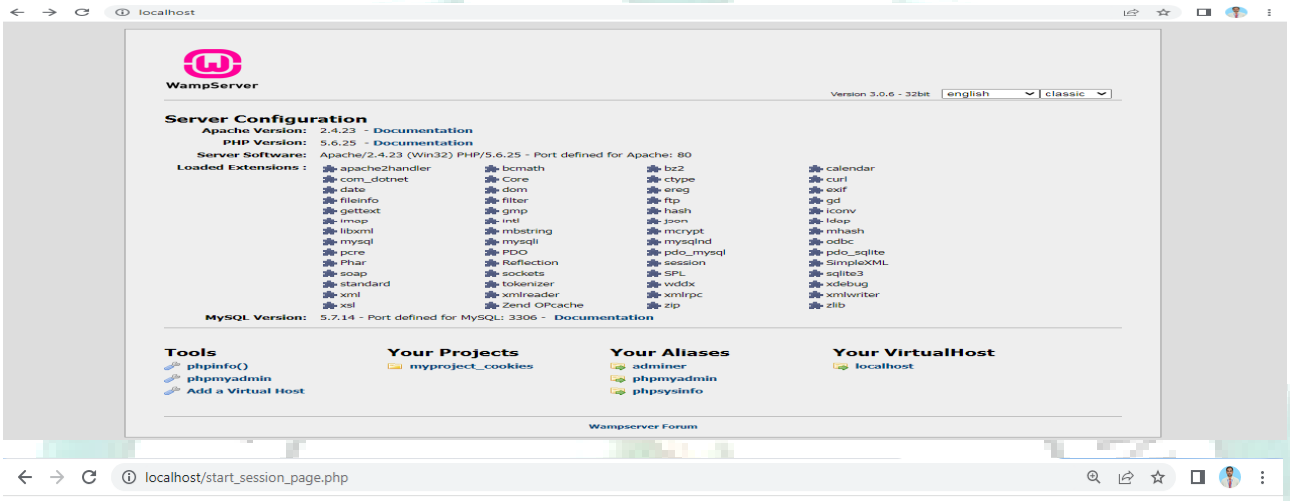
In the code above, we have started a session and set two session variables. Above webpage will also have a link to navigate to Second page **second_page.php**.

Below is the code for **second_page.php**, in which we fetch values from the session variable which are set in the **start_session_page.php**.

second_page.php

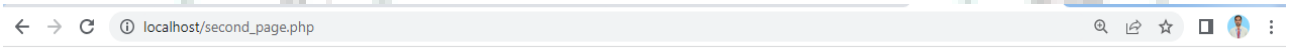
```
<?php
session_start(); // start the session
$username = $_SESSION["username"]; // get the session variable values
$userid = $_SESSION["userid"]; // get the session variable values
?>
<html>
   <body>
<center>
 <h1>
   <?php
   echo "User Name is: ".$username."<br/>";
   echo "User Id is: ".$userid;
   ?>
</h1>
 </center>
   </body>
</html>
```

OUTPUT:



Session variable is set.

Go to Second Page



User Name is: cse
User Id is: 501

You must be thinking, why we used session_start() in "**second_page.php**" although we did not set any new values in the session variable.

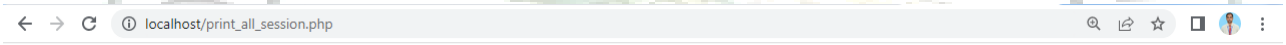
session_start() function is used to initialize a new session and to fetch the ongoing session(if already started), and then, using the $_SESSION global variable, we can either set new values into the session or get the saved values.

If there are too many values stored in the session, and you don't know which one do you want to get, you can use the below code to print all the current session variable data.

print_all_session.php

```
<?php
session_start();// start the session
?>
<html>
  <body>
 <?php
  print_r($_SESSION);
  ?>
    </body>  </html>
```

OUTPUT:



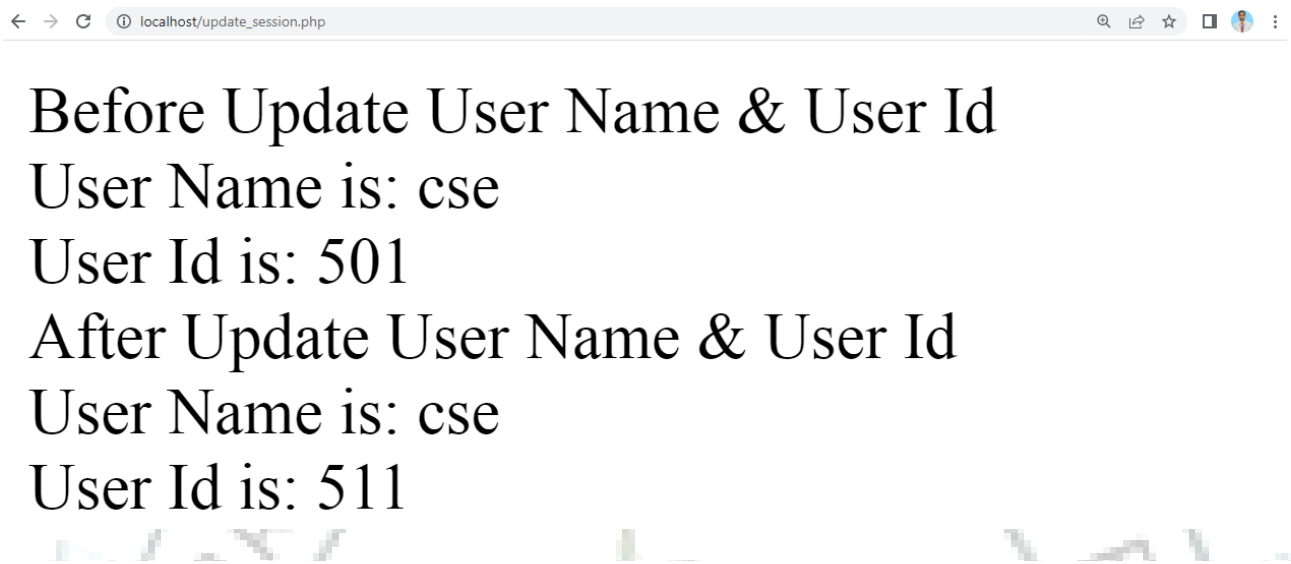
Array ([username] => cse [userid] => 501)

Update Session Variable in PHP

To update any value stored in the session variable, start the session by calling session_start() function and then simply overwrite the value to update session variable.

Update_session.php

```
<?php
session_start(); // start the session
echo "Before Update User Name & User Id"."<br/>";
echo "User Name is: ".$_SESSION["username"]."<br/>";
echo "User Id is: ".$_SESSION["userid"]."<br/>";
$_SESSION["userid"] = "511"; // update the session variable values
?>
<html>
  <body>
  <?php
  echo "After Update User Name & User Id"."<br/>";
  echo "User Name is: ".$_SESSION["username"]."<br/>";
  echo "User Id is: ".$_SESSION["userid"];
  ?>      </body> </html>
```



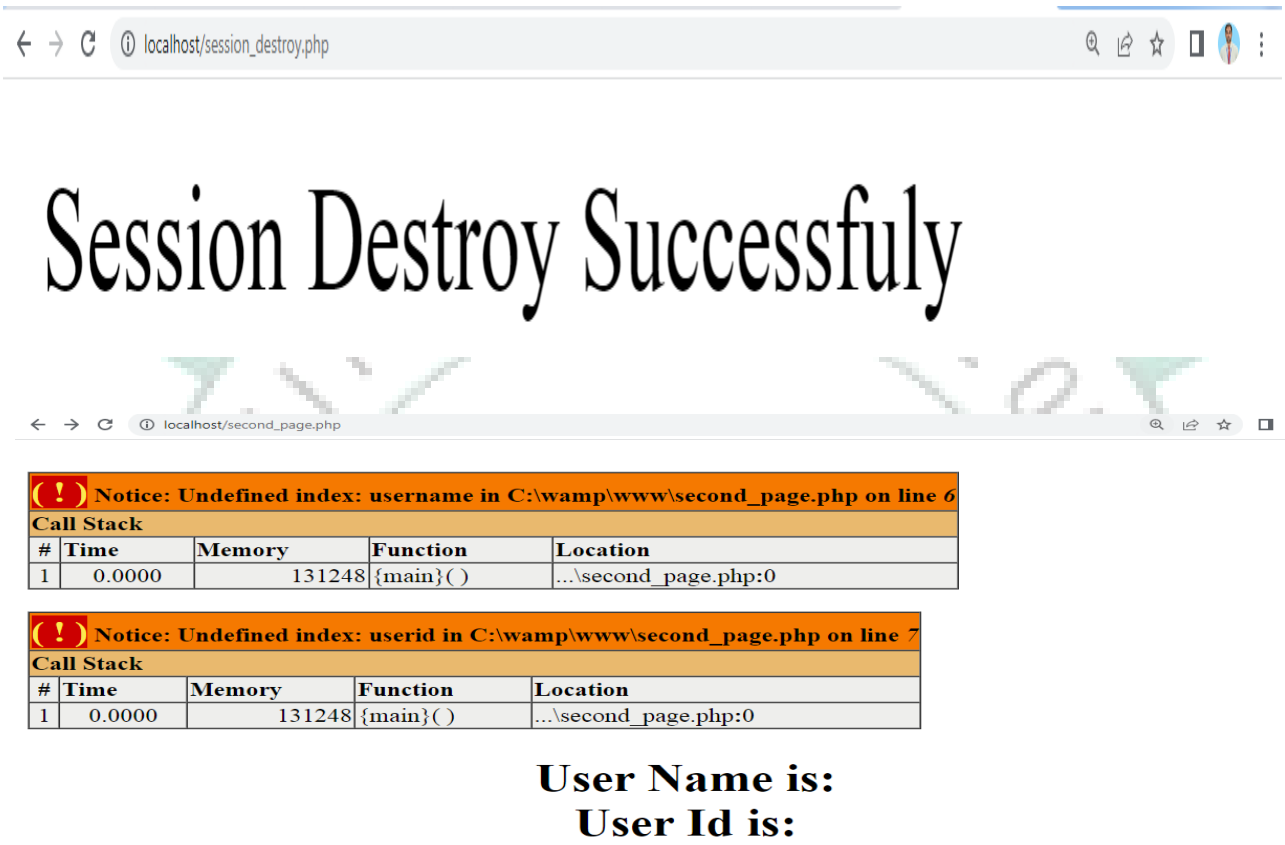
Destroy a Session in PHP

To clean the session variable or to remove all the stored values from the session variable we can use the function session_unset() and to detroy the session, we use session_destroy() function.

session_destroy.php

```
<?php
session_start(); // start the session
?>
<html>
   <body>
   <?php
   session_unset();    // clean the session variable
      session_destroy();// destroy the session
   echo "Session Destroy Successfuly";
   ?>
   </body>
</html>
```

OUTPUT:



Module 6:

File Handling in PHP

When we develop a web application using PHP, quite often we need to work with external files, like reading data from a file or maybe writing user data into file etc. So it's important to know how files are handled while working on any web application.

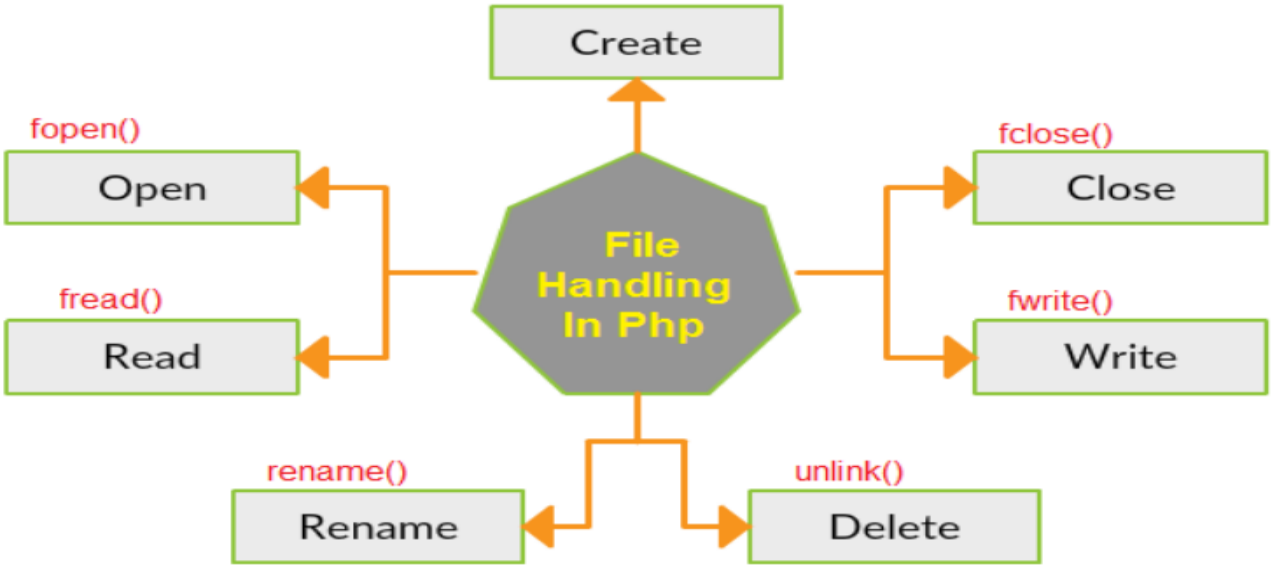
File Handling Operations

File handling starts with creating a file, reading its content, writing into a file to appending data into an existing file and finally closing the file. Php provides pre-defined functions for all these operations, so let's start by knowing these functions.

1. **Create a File**: fopen()
2. **Open a File**: fopen()
3. **Read a File**: fread()
4. **Write to a File**: fwrite()
5. **Append to a File**: fwrite()

6. **Close a File**: fclose()
7. **Delete a File**: unlink()

You must be wondering that why we have specified same functions for multiple file operations, well, that is because just by changing one or more arguments, same function can be used to perform multiple operations on file.



PHP Open File - fopen()

The PHP fopen() function is used to open a file.

Syntax

fopen(filename,mode,include_path,context)

Where:

filename is the name of the file or URL to open

mode is the mode in which to open the file. This can be one of the following:

'r': read only

'w': write only (truncates the file to zero length or creates a new file)

'a': append-only (opens the file for writing at the end of the file)

'x': exclusive write (creates a new file and opens it for writing only if it doesn't already exist)

'b': binary mode (used in conjunction with the above modes to indicate that the file should be opened in binary mode)

't': text mode (used in conjunction with the above modes to indicate that the file should be opened in text mode)

include_path is a boolean parameter that indicates whether to search for the file in the include path (if set)

context is an optional parameter that allows you to specify a context for the file stream (e.g. HTTP headers, SSL settings, etc.)

PHP Close File - fclose()

The PHP fclose() function is used to close an open file pointer.

Syntax

fclose(file)

sample.txt

WELCOME TO SERVERSIDE SCRIPTING LAB

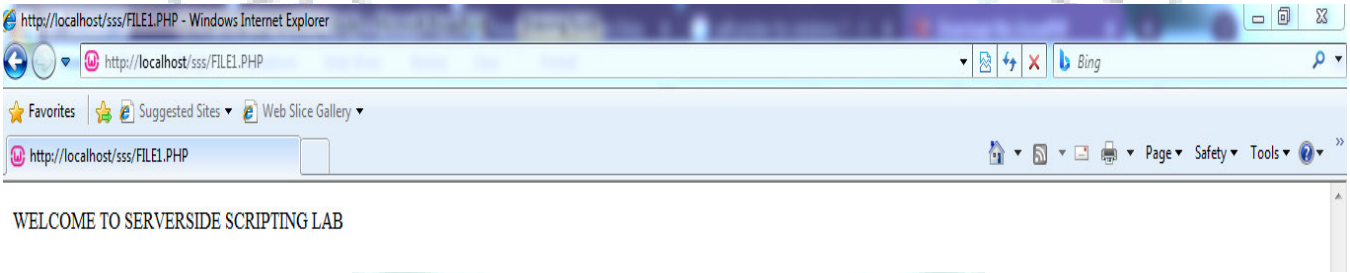
WELCOME TO SERVERSIDE SCRIPTING LAB

WELCOME TO SERVERSIDE SCRIPTING LAB

Example1:

```
<?php
$fp= fopen('sample.txt', 'r');
        echo fgets($fp);
        fclose($fp);
?>
```

OUTPUT:

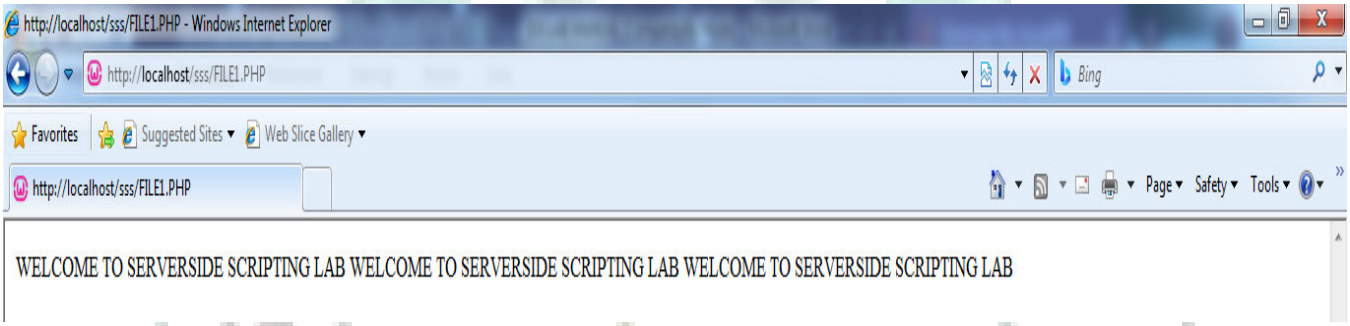


Example2:

```
<?php
        $fp= fopen('sample.txt', 'r');
        while(!feof($fp))
        {
         echo fgetc($fp);
        }
```

```
fclose($fp);
?>
```

OUTPUT:



PHP Read File - fread()

The PHP fread() function is used to read the content of the file. It accepts two arguments: resource and file size.

Syntax

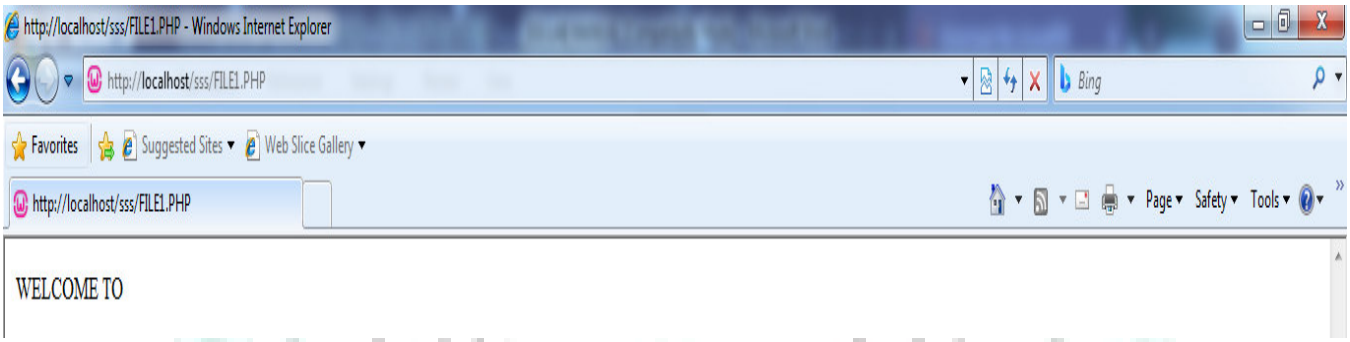
fread(file,length)

Parameter	Description
file	Required. Specifies the open file to read from
length	Required. Specifies the maximum number of bytes to read

Example 1: Read 10 bytes from file:

```
<?php
$file = fopen("sample.txt","r");
$fileContents =fread($file,10);
echo $fileContents;
fclose($file);
?>
```

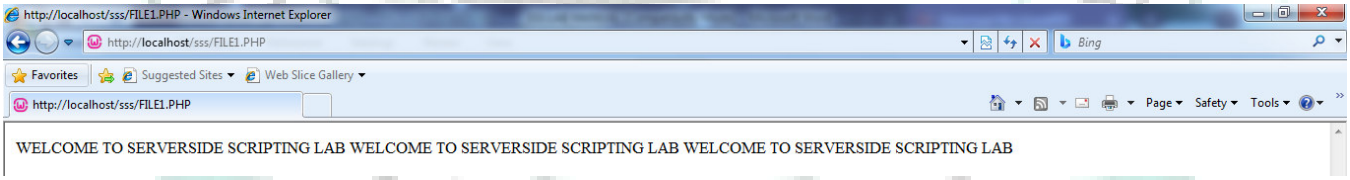
OUTPUT:



Example 2: Read Entire file:

```
<?php
    $file = fopen("sample.txt","r");
    $fileContents =fread($file,
                filesize("Sample.txt"));
   echo $fileContents;
   fclose($file);
?>
```

OUTPUT:



Write Files In PHP

PHP fwrite() and fputs() functions are used to write data into file. To write data into a file, you need to use w, r+, w+, x, x+, c or c+ mode.

PHP Write File - fwrite()

The PHP fwrite() function is used to write content of the string into file.

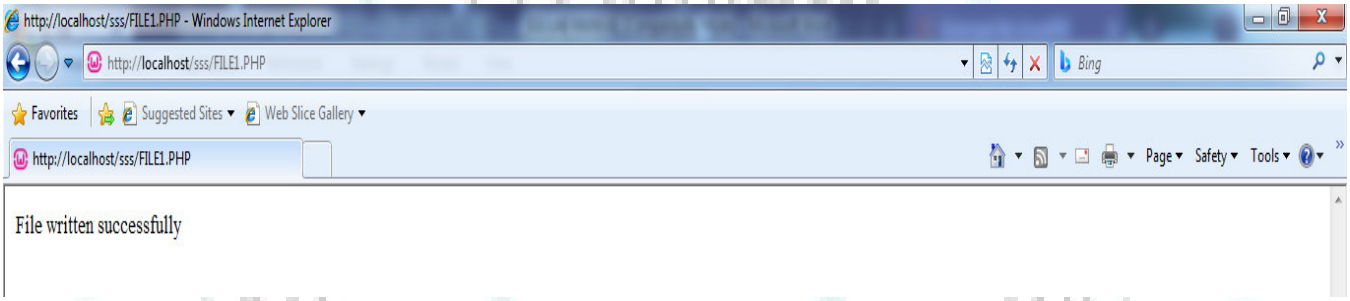
Syntax

fwrite(file,string,length)

PHP Write File - fwrite() Example

```
<?php
$fp = fopen("cse.txt", "w");
fwrite($fp, "welcome");
fwrite($fp, "to CSE 6th SEM SSS Lab");
fclose($fp);
echo "File written successfully";
?>
```

OUTPUT:

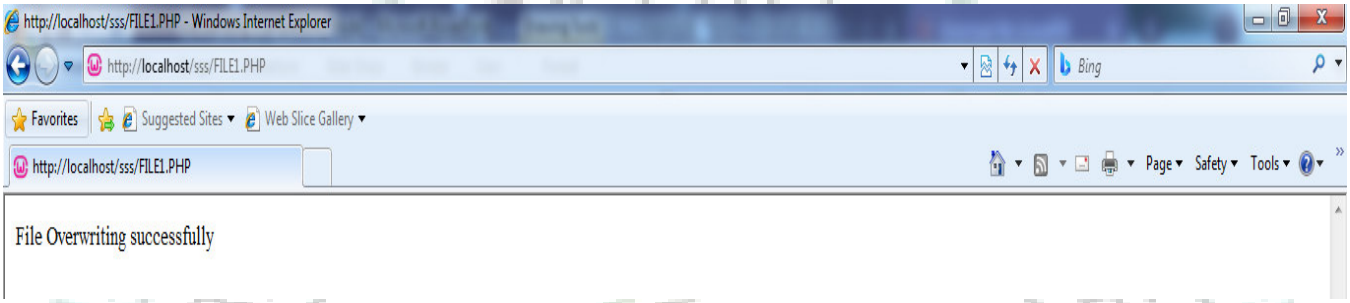


PHP Overwriting File

```
<?php
$fp = fopen("cse.txt", "w");
fwrite($fp, "PHP File HANDLING OPERATIONS");
```

```
fclose($fp);

echo "File Overwriting successfully";

?>
```

OUTPUT:



PHP rename() Function

The rename() function renames a file or directory.

This function returns TRUE on success or FALSE on failure.

Syntax

rename(oldname,newname,context)

Parameter Description

oldname Required. Specifies the file or directory to be renamed

newname Required. Specifies the new name of the file or directory

context Optional. Specifies the context of the filehandle. Context is a set of options that can modify the behavior of a stream

Example:

```
<?php

rename("cse.txt","aiml.txt");

echo "File Rename Successful";
```

?>

Delete Files In PHP

- we can delete any file using unlink() function.
- The unlink() function accepts one argument only: file name.
- It is similar to UNIX C unlink() function.

PHP unlink() generates E_WARNING level error if file is not deleted.

It returns TRUE if file is deleted successfully otherwise FALSE.

Syntax

bool unlink (string $filename [, resource $context])

unlink ("file name with directory path")

$filename represents the name of the file to be deleted.

```
//PHP Delete File Example
<?php
$status=unlink("dataFile.txt");
if($status)
{
echo "File deleted successfully";
}
else
{
echo "Opps Your File Not Delete !";
}
?>
```

b) Develop a PHP program to demonstrate the process of uploading and downloading the file.

File Upload In PHP

PHP file upload features allow you to upload binary and text files both.

PHP $_FILES

- The PHP global $_FILES contains all the information of the file.
- With the help of $_FILES global, we can get the file name, file type, file size, temp file name and errors associated with the file.

$_FILES["filename"]["name"] :- returns file name.

$_FILES["filename"]["type"] :- returns MIME type of the file.

$_FILES["filename"]["size"] :- returns size of the file (in bytes).

$_FILES["filename"]["tmp_name"] :- returns temporary file name of the file which was stored on the server.

$_FILES["filename"]["error"] :- returns error code associated with this file.

move_uploaded_file() function / Saving the Uploaded File

- The move_uploaded_file() function moves the uploaded file to a new location.
- The move_uploaded_file() function checks internally if the file is uploaded thorough the POST request.
- It moves the file if it is uploaded through the POST request.

Syntax:

- bool move_uploaded_file (string $filename , string $destination) ;
- move_uploaded_file ("Your File Name With Temp Name","Your Directory Path With Image Name")

fUpload.html

```
<html>
<head>
<title>File Upload Sample Program</title>
</head>
<body>
<form action="fUpload.php" method="post" enctype="multipart/form-data">
<center>
<h1>File Upload Sample Program</h1>
  Select File:
  <input type="file" name="fileUpload" />
  <input type="submit" value="Click here to Upload " name="submit" />
</center>
</form>
</body>
</html>
```

fUpload.php

```php
<?php
$target_dir = "C:/sss/";

$target_path = $target_dir.basename( $_FILES["fileUpload"]["name"]);

if(move_uploaded_file($_FILES["fileUpload"]["tmp_name"],$target_path))
{
  echo "File uploaded successfully!";
}
else
{
  echo "Sorry, file not uploaded, please try again!";
}
?>
```

Download File In PHP

PHP enables you to download the file easily using a built-in readfile() function. The readfile() function reads a file and writes it to the output buffer.

PHP readfile() function

Syntax

int readfile (string $filename [, bool $use_include_path = false [, resource $context]])

$filename: represents the file name

$use_include_path: it is the optional parameter. It is by default false. You can set it to true to the search the file in the included_path.

$context: represents the context stream resource.

int: it returns the number of bytes read from the file.

PHP Download File Example: Text File

fDownload1.php

```
<?php
$file_url = "C:/sss/file.txt";
header("Content-Type: application/octet-stream");
header("Content-Transfer-Encoding: utf-8");
header("Content-disposition: attachment; filename=\"".basename ($file_url)."\"");
readfile($file_url);
?>
```

PHP Download File Example: Binary File

fDownload2.php

```
<?php
$file_url = "C:/sss/file.txt";
header("Content-Type: application/octet-stream");
header("Content-Transfer-Encoding: Binary");
header("Content-disposition: attachment; filename=\"".basename ($file_url)."\"");
readfile($file_url);
?>
```

Module 7:

a) Develop a PHP program to describe the importance of mail function to send the email.

b) Develop a PHP and AJAX program which demonstrates how a web page can communicate with a web server while a user types characters in an input filed.

- AJAX stands for Asynchronous JavaScript and XML. AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS, and JavaScript.
- Conventional web applications transmit information to and from the server using synchronous requests. It means you fill out a form, hit submit, and get directed to a new page with new information from the server.
- With AJAX, when submit is pressed, JavaScript will make a request to the server, interpret the results and update the current screen. In the purest sense, the user would never know that anything was even transmitted to the server.

The following example will demonstrate how a web page can communicate with a web server while a user types characters in an input field:

Start typing a name in the input field below:

First name:

Suggestions:

Explanation

In the example above, when a user types a character in the input field, a function called "showing()" is executed.

The onkeyup event triggers the function.

Here is the HTML code:

```
<html>
<head>
<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>
</head>
<body>

<p><b>Start typing a name in the input field below:</b></p>
<form action="">
 <label for="fname">First name:</label>
 <input type="text" id="fname" name="fname" onkeyup="showHint(this.value)">
</form>
<p>Suggestions: <span id="txtHint"></span></p>
</body>
</HTML>
```

Code explanation:

First, check if the input field is empty (str.length == 0). If it is, clear the content of the txtHint placeholder and exit the function.

Do the following, if the input field is not empty:

- Create an XMLHttpRequest object
- Create the function to be executed when the server response is ready
- Send the request off to a PHP file (gethint.php) on the server
- Notice that the q parameter is added to the url (gethint.php?q="+str)
- And the str variable holds the content of the input field

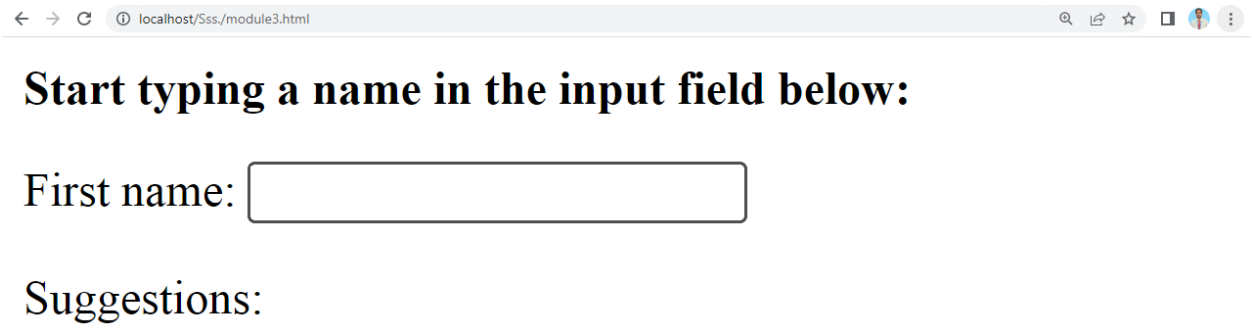
The PHP File - "gethint.php."

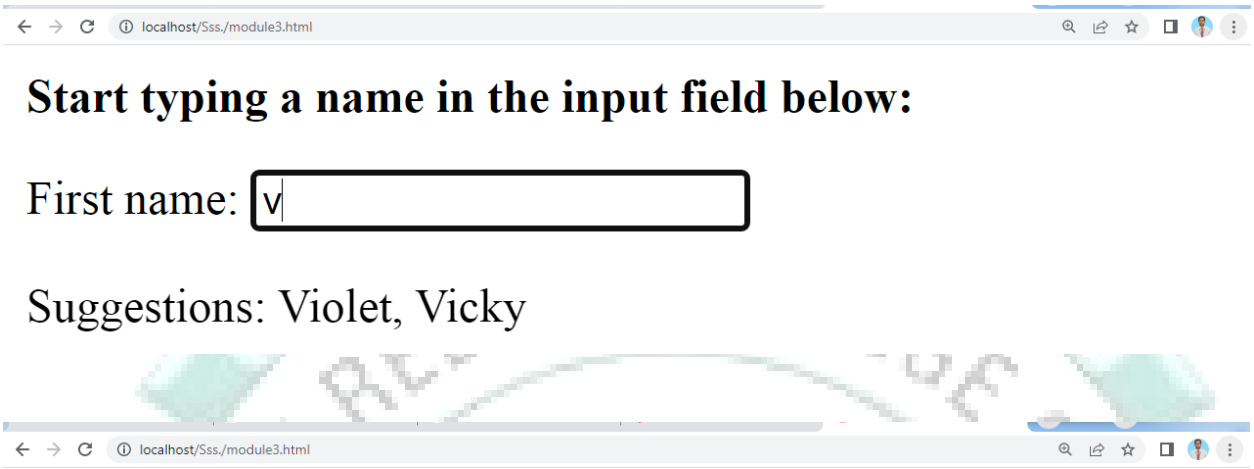
The PHP file checks an array of names, and returns the corresponding name(s) to the browser:

```
<?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";

// fetch q parameter from URL
$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";
      }
    }
  }
}

// It results in "no suggestion" if no hint was found or output correct values
echo $hint === "" ? "no suggestion" : $hint;
?>
```

Output: both html & php files are placed into “www” folder and open html file through localhost





Module 8:

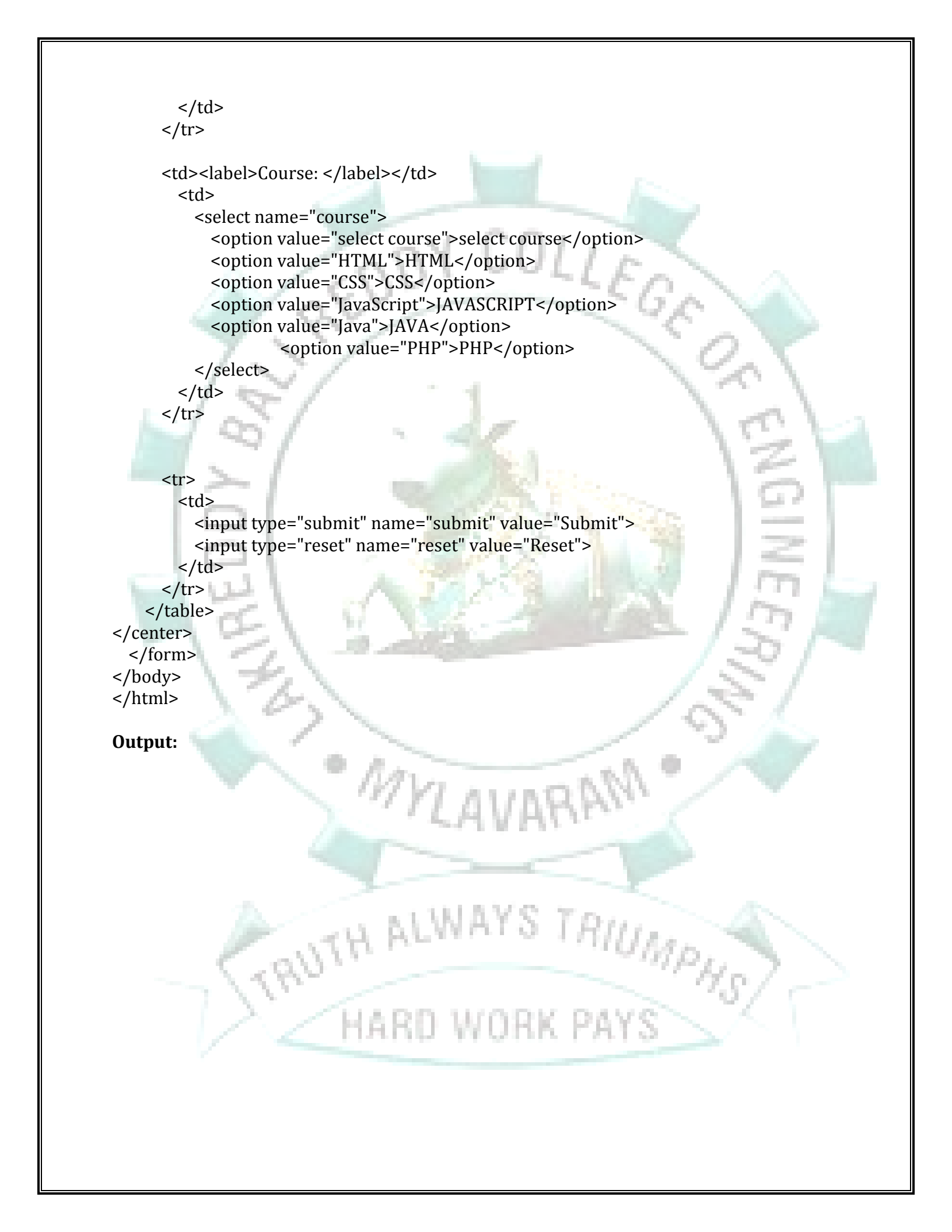
a) Develop a PHP web application which demonstrates the process of storing HTML form data into MySQL database.

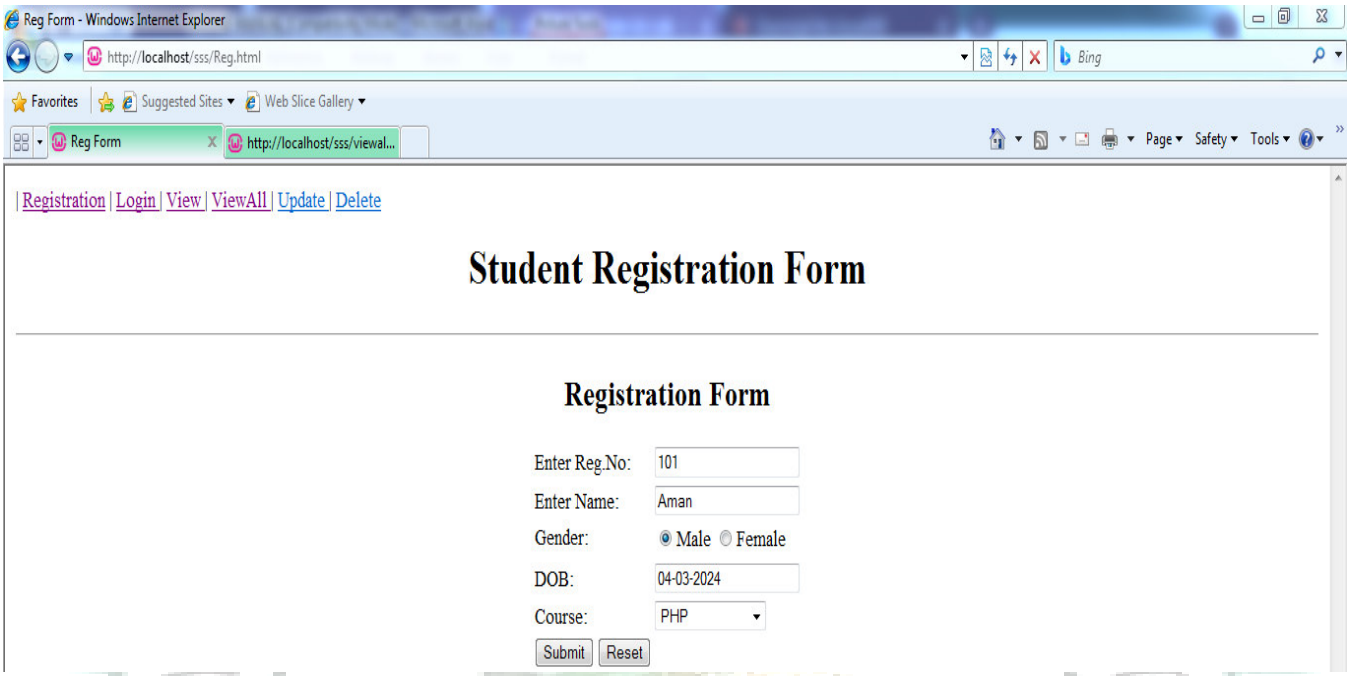
REGISTRATION.HTML

```
<html>
<head>
  <title>Reg Form</title>
</head>
<body>
| <a href="Reg.html">Registration</a> | <a href="login.html"> Login </a> | <a
href="view.html"> View </a> | <a href="viewall.php"> ViewAll </a> | <a href="update.html">
Update </a> | <a href="delete.html"> Delete </a>


  <center><h1>Student Registration Form</h1></center>
  <hr>
  <form method="post" action="Reg.php" name="reg_form">
       <center>
    <h2>Registration Form</h2>
    <table>
      <tr>
        <td><label>Enter Reg.No: </label></td>
        <td>
          <input type="text" name="regno" placeholder="Enter Reg.NO">
        </td>
      </tr>
      <tr>
        <td><label>Enter Name: </label></td>
        <td>
          <input type="text" name="sname" placeholder="Enter Name">
        </td>
      </tr>

      <tr>
        <td><label>Gender: </label></td>
        <td>
          <input type="radio" name="gender" value="male">Male
          <input type="radio" name="gender" value="femele">Female
        </td>
      </tr>
      <tr>
<tr>
        <td><label>DOB: </label></td>
        <td>
          <input type="date" name="dt">
```

```
        </td>
      </tr>

      <td><label>Course: </label></td>
        <td>
          <select name="course">
            <option value="select course">select course</option>
            <option value="HTML">HTML</option>
            <option value="CSS">CSS</option>
            <option value="JavaScript">JAVASCRIPT</option>
            <option value="Java">JAVA</option>
                    <option value="PHP">PHP</option>
          </select>
        </td>
      </tr>


      <tr>
        <td>
          <input type="submit" name="submit" value="Submit">
          <input type="reset" name="reset" value="Reset">
        </td>
      </tr>
    </table>
</center>
  </form>
</body>
</html>
```

Output:



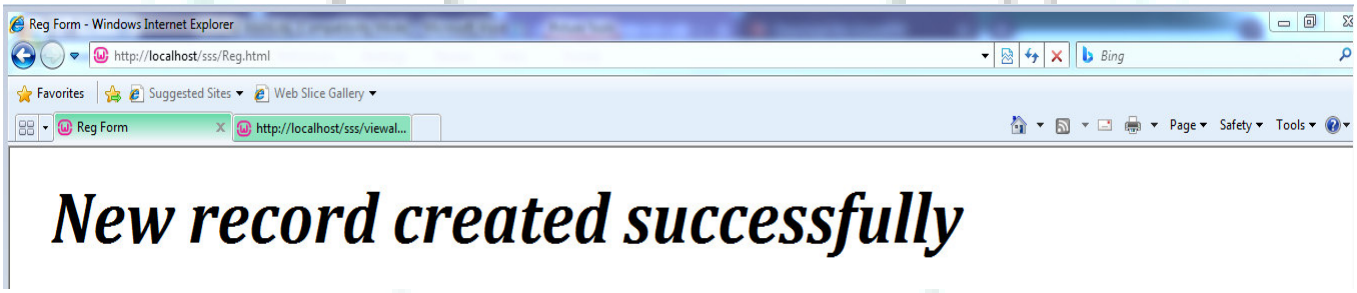


Reg.php

```
<?php
include("links.html");
$rno=$_POST['regno'];
$name=$_POST['sname'];
$gen=$_POST['gender'];
$dob=$_POST['dt'];
$crs=$_POST['course'];

$servername="localhost"; //local server name default localhost
$username="root"; //mysql username default is root.
$password=""; //blank if no password is set for mysql.
$dbname = "lbrce";
// Create connection
$conn = mysqli_connect($servername, $username, $password,$dbname);// Check connection
if(!$conn)
```

```
{
die("Connection failed:".mysqli_connect_error());
}
$sql = "INSERT INTO student_info (regno,sname,gender,dob,course)VALUES
('$rno','$name','$gen','$dob','$crs')";
if (mysqli_query($conn, $sql))
{
 echo "<center><h1>New record created successfully </center></h1>";
} else
 {
 echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}
mysqli_close($conn);
?>
```

OUTPUT:



b) Develop a PHP web application which demonstrates the process of retrieving table data from the MySQL database and display it in the HTML table.

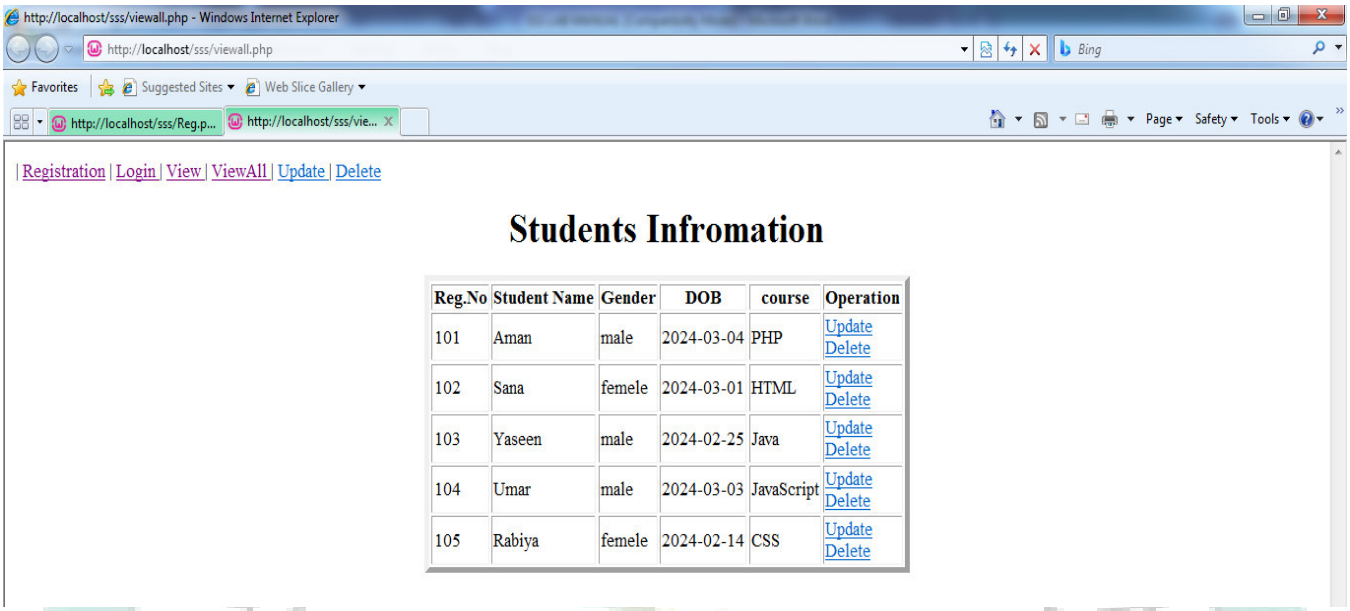
Viewall.php

```
<?php
include("links.html");
```

```
$servername="localhost"; //local server name default localhost
$username="root"; //mysql username default is root.
$password=""; //blank if no password is set for mysql.
$dbname = "lbrce";
// Create connection
$conn = mysqli_connect($servername, $username, $password,$dbname);// Check connection
if(!$conn)
{
die("Connection failed:".mysqli_connect_error());
}
$sql = "SELECT * FROM student_info";
$result = mysqli_query($conn, $sql);
 ?>
<h1 align="center">Students Infromation</h1>
<table align="center" border="5">
<tr>
<th>Reg.No</th>
<th>Student Name</th>
<th>Gender</th>
<th>DOB</th>
<th>course</th>
<th>Operation</th>
</tr>
<?php
  while($row = mysqli_fetch_assoc($result))
{
```

```
?>

<tr>
<td><?php echo $row["regno"]?> </td>

<td><?php echo $row["sname"]?> </td>

<td><?php echo $row["gender"]?> </td>

<td><?php echo $row["dob"]?> </td>

<td><?php echo $row["course"]?> </td>

<td><a href="update.php">Update </br> <a href="delete.php">Delete</td>
</tr>
<?php
 }
?>
</table>
<?php
mysqli_close($conn);
?>
```

OUTPUT:

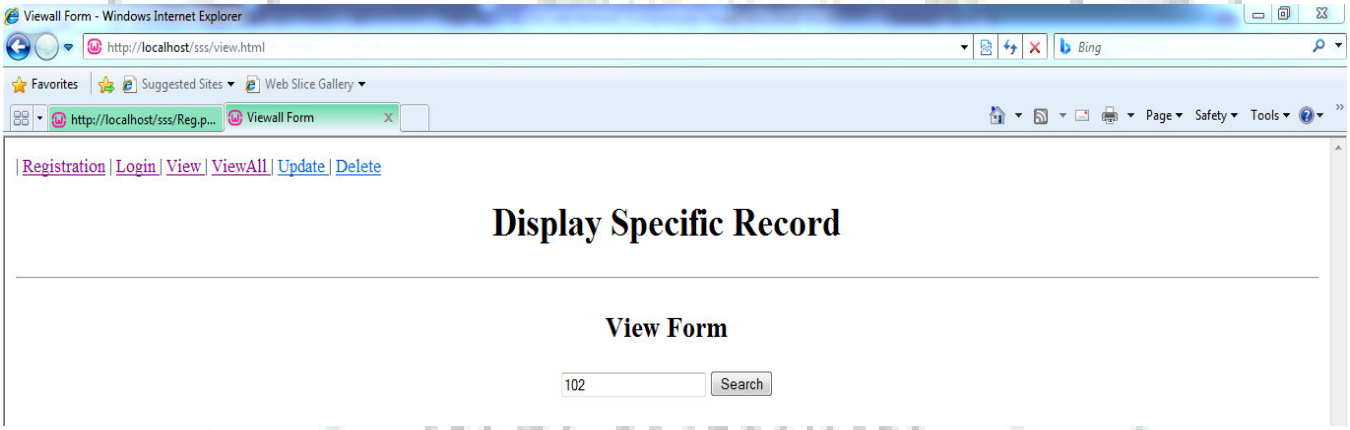


View.html

```
<html>
<head>
   <title>Viewall Form</title>
</head>
<body>
| <a href="Reg.html">Registration</a> | <a href="login.html"> Login </a> | <a
href="view.html"> View </a> | <a href="viewall.php"> ViewAll </a> | <a href="update.html">
Update </a> | <a href="delete.html"> Delete </a>
   <center><h1>Display Specific Record</h1></center>
   <hr>
   <form method="post" action="view.php" name="view_form">
        <center>
     <h2>View Form</h2>
     <table>
        <tr>
```

```
        <td>
            <input type="text" name="regno" value="Enter Roll.No" >
                <input type="submit" name="submit" value="Search">
            </td>
        </tr>
    </table>
</center>
  </form>
</body>
</html>
```

OUTPUT:



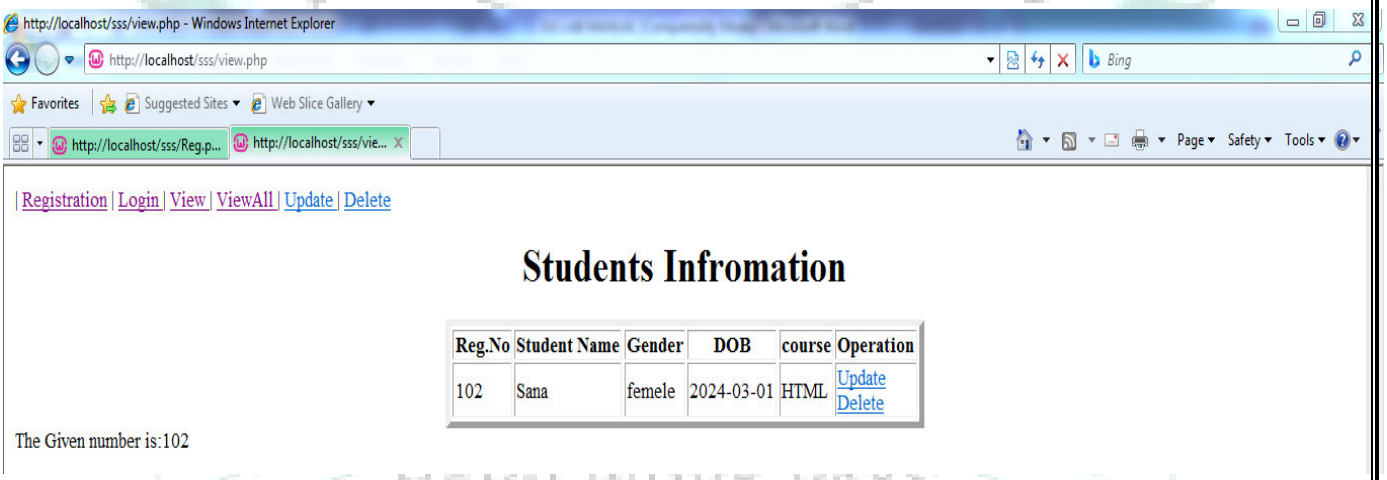
View.php

```
<?php
include("links.html");
$rno=$_POST['regno'];
$servername="localhost"; //local server name default localhost
$username="root"; //mysql username default is root.
$password=""; //blank if no password is set for mysql.
```

```
$dbname = "lbrce";
// Create connection
$conn = mysqli_connect($servername, $username, $password,$dbname);// Check connection
if(!$conn)
{
die("Connection failed:".mysqli_connect_error());
}
$sql = "SELECT * FROM student_info where regno='+$rno+'";
$result = mysqli_query($conn,$sql);
 ?>
<h1 align="center">Students Infromation</h1>
<table align="center" border="5">
<tr>
<th>Reg.No</th>
<th>Student Name</th>
<th>Gender</th>
<th>DOB</th>
<th>course</th>
<th>Operation</th>
</tr>
<?php
  while($row = mysqli_fetch_assoc($result))
{
?>

<tr>
```

```
<td><?php echo $row["regno"]?> </td>
<td><?php echo $row["sname"]?> </td>
<td><?php echo $row["gender"]?> </td>
<td><?php echo $row["dob"]?> </td>
<td><?php echo $row["course"]?> </td>
<td><a href="update.php">Update </br> <a href="delete.php">Delete</td>
</tr>
<?php
 }
?>
</table>
<?php
echo "The Given number is:".$rno;
mysqli_close($conn);
?>
```

OUTPUT:



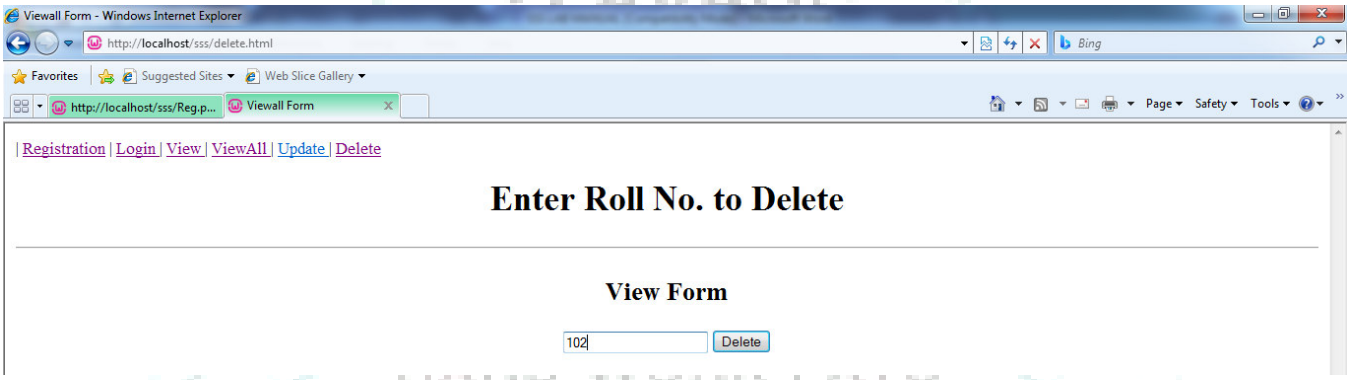
Module 9:

a) Develop a PHP web application which demonstrates the process of deleting a particular record in MySQL database table.

delete.html

```
<html>
<head>
  <title>Viewall Form</title>
</head>
<body>
| <a href="Reg.html">Registration</a> | <a href="login.html"> Login </a> | <a
href="view.html"> View </a> | <a href="viewall.php"> ViewAll </a> | <a href="update.html">
Update </a> | <a href="delete.html"> Delete </a>
  <center><h1>Enter Roll No. to Delete</h1></center>
  <hr>
  <form method="post" action="delete.php" name="view_form">
       <center>
    <h2>View Form</h2>
    <table>
      <tr>
        <td>
              <input type="text" name="regno" value="Enter Roll.No" >
                    <input type="submit" name="Delete" value="Delete">
              </td>
      </tr>
    </table>
</center>
  </form>
</body>
</html>
```

OUTPUT:

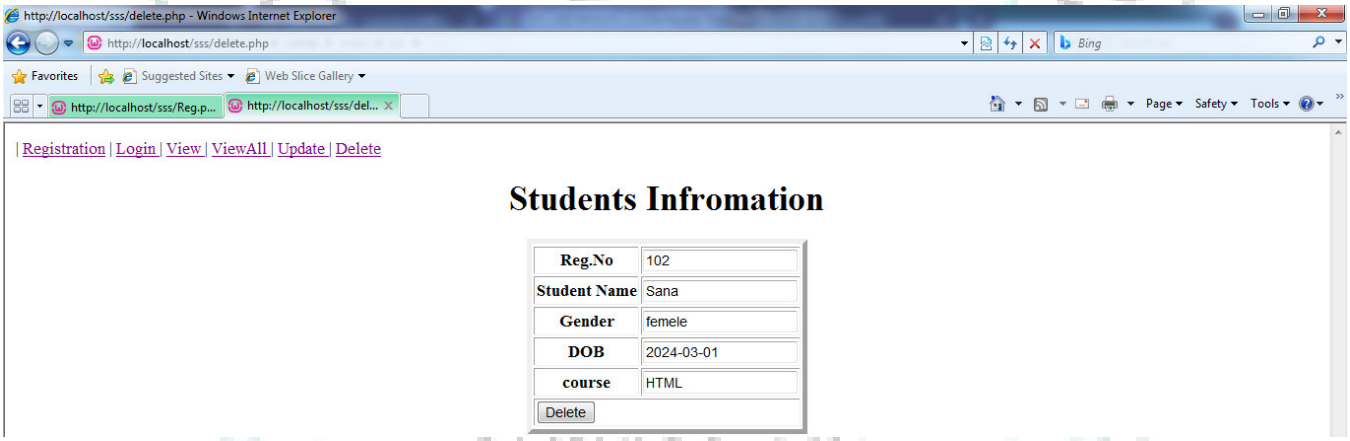


delete.php

```
<?php
include("links.html");
$rno=$_POST['regno'];
$servername="localhost"; //local server name default localhost
$username="root"; //mysql username default is root.
$password=""; //blank if no password is set for mysql.
$dbname = "lbrce";
// Create connection
$conn = mysqli_connect($servername, $username, $password,$dbname);// Check connection
if(!$conn)
{
die("Connection failed:".mysqli_connect_error());
}
$sql = "SELECT * FROM student_info where regno='+$rno+'";
$result = mysqli_query($conn,$sql);
 ?>
<form name="update_form" action="<?php $_PHP_SELF?>" method="post">
<h1 align="center">Students Infromation</h1>
<table align="center" border="5">
<?php
  while($row = mysqli_fetch_assoc($result))
{
?>

<tr>
<th>Reg.No</th>
<td><input type="text" name="rn" value=<?php echo $row["regno"] ?> /></td>
</tr>

<tr>
<th>Student Name</th>
<td><input type="text" name="sn" value=<?php echo $row["sname"]?> /></td>
</tr>

<tr>
<th>Gender</th>
<td><input type="text" name="g" value=<?php echo $row["gender"]?> /></td>
</tr>

<tr>
<th>DOB</th>
<td><input type="text" name="d" value=<?php echo $row["dob"]?> /></td>
</tr>

<tr>
```

```
<th>course</th>
<td><input type="text" name="c" value=<?php echo $row["course"]?> /></td>
</tr>

<tr>
<td colspan="2"><input type="submit" value="Update" align="center"/></td>
</tr>
</table>
</form>
<?php
}
$rno=$_POST["rn"];

$sql = "DELETE * from student_info where regno='+$rno+'";
$result = mysqli_query($conn,$sql);
if(!$result)
{
echo "Roll No ".$rno." Record Deleted Successfull";
}
else
{
echo "NOT Deleted";
}
?>
```

OUTPUT:



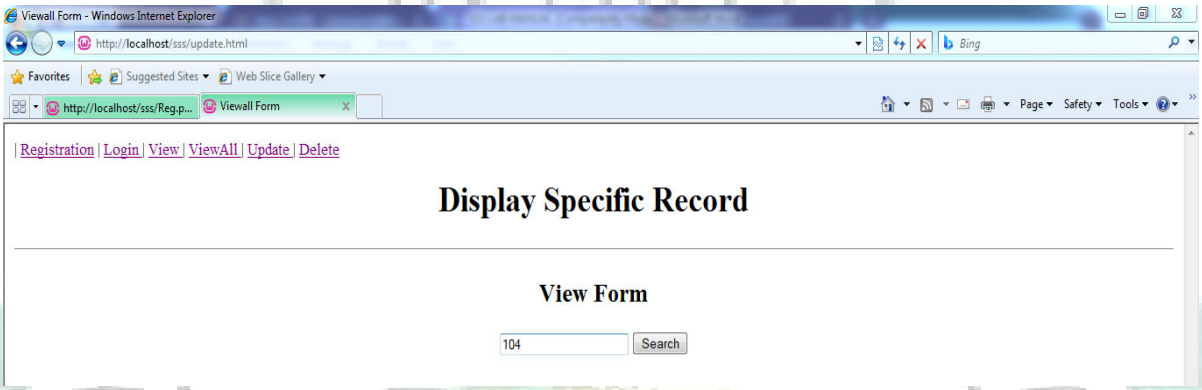
b) Develop a PHP web application to illustrate the process of updating MySQL database table data.

update.hml

```
<html>
<head>
  <title>Viewall Form</title>
</head>
<body>
| <a href="Reg.html">Registration</a> | <a href="login.html"> Login </a> | <a
href="view.html"> View </a> | <a href="viewall.php"> ViewAll </a> | <a
href="update.html">
Update </a> | <a href="delete.html"> Delete </a>
  <center><h1>Display Specific Record</h1></center>
  <hr>
  <form method="post" action="update.php" name="view_form">
      <center>
    <h2>View Form</h2>
    <table>
      <tr>
        <td>
            <input type="text" name="regno" value="Enter Roll.No" >
                <input type="submit" name="submit" value="Search">
            </td>
      </tr>
    </table>
</center>
```

```
  </form>
</body>
</html>
```

OUTPUT:



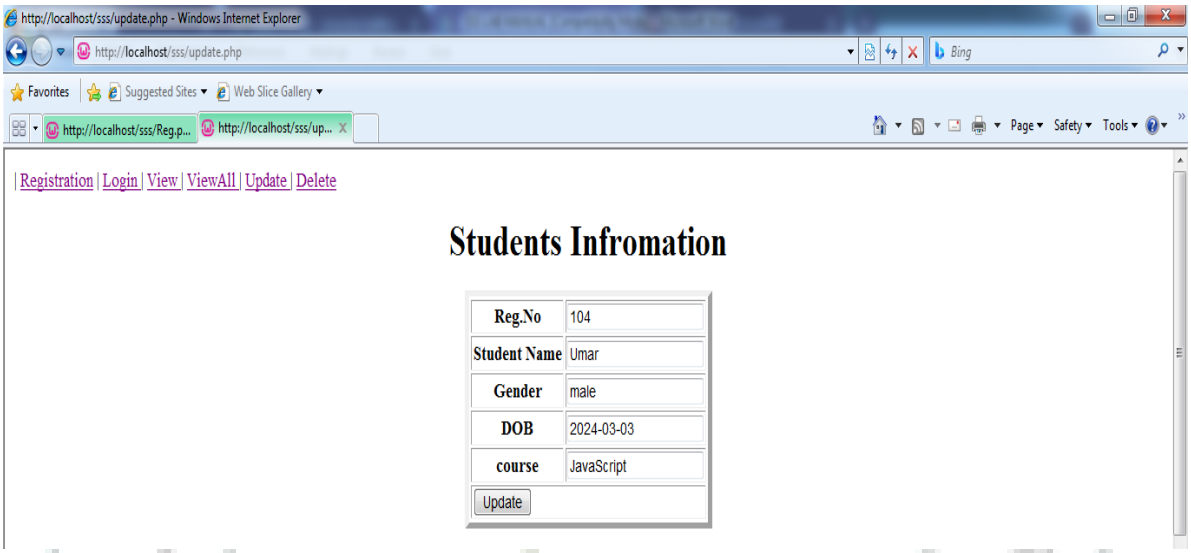
Update.php

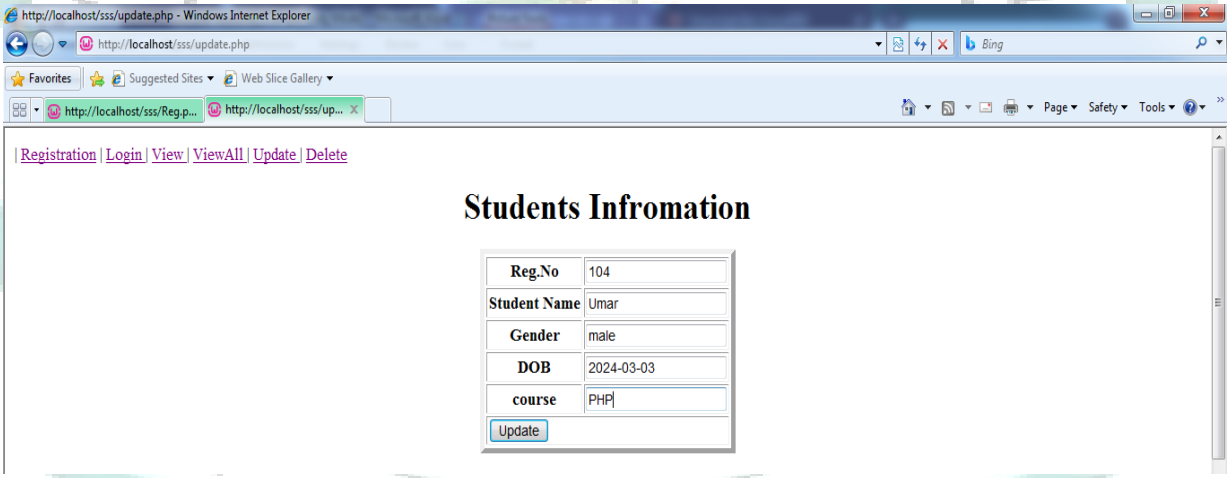
```
<?php
include("links.html");
$rno=$_POST['regno'];
$servername="localhost"; //local server name default localhost
$username="root"; //mysql username default is root.
$password=""; //blank if no password is set for mysql.
$dbname = "lbrce";
// Create connection
$conn = mysqli_connect($servername, $username, $password,$dbname);// Check
connection
if(!$conn)
{
die("Connection failed:".mysqli_connect_error());
}
$sql = "SELECT * FROM student_info where regno='+$rno+'";
$result = mysqli_query($conn,$sql);
 ?>
```

```
<form name="update_form" action="<?php $_PHP_SELF?>" method="post">
<h1 align="center">Students Infromation</h1>
<table align="center" border="5">
<?php
 while($row = mysqli_fetch_assoc($result))
{
?>

<tr>
<th>Reg.No</th>
<td><input type="text" name="rn" value=<?php echo $row["regno"] ?> /></td>
</tr>

<tr>
<th>Student Name</th>
<td><input type="text" name="sn" value=<?php echo $row["sname"]?> /></td>
</tr>

<tr>
<th>Gender</th>
<td><input type="text" name="g" value=<?php echo $row["gender"]?> /></td>
</tr>

<tr>
<th>DOB</th>
<td><input type="text" name="d" value=<?php echo $row["dob"]?> /></td>
</tr>

<tr>
<th>course</th>
<td><input type="text" name="c" value=<?php echo $row["course"]?> /></td>
</tr>
```

```
<tr>
<td colspan="2"><input type="submit" value="Update" align="center"/></td>
</tr>
</table>
</form>
<?php
}
$rno=$_POST["rn"];
$sname=$_POST["sn"];
$gen=$_POST["g"];
$dob=$_POST["d"];
$cs=$_POST["c"];
echo "The Given number is:".$rno;
echo "The Given number is:".$sname;
echo "The Given number is:".$gen;
echo "The Given number is:".$dob;
echo "The Given number is:".$cs;
$sql = "UPDATE student_info set sname='+sname+', gender='+gen' where regno='+$rno+'";
$result = mysqli_query($conn,$sql);
if(!$result)
{
echo "UPdate Successfull";
}
else
{
echo "NOT UPDATED";
}
?>
```

OUTPUT:





updt.php

```
<?php
include("links.html");
include("update.php");
$servername="localhost"; //local server name default localhost
$username="root"; //mysql username default is root.
$password=""; //blank if no password is set for mysql.
$dbname = "lbrce";
// Create connection
$conn = mysqli_connect($servername, $username, $password,$dbname);// Check connection
```

```
if(!$conn)
{
die("Connection failed:".mysqli_connect_error());
}
$sql = "UPDATE student_info set sname=+$sname+, gender=+$gen+ where regno='+$rno+'";
$result = mysqli_query($conn,$sql);
if(!$result)
{
?>
<h1>Record Updated Succesfully</h1>
<?php
}
else
{
?>
<h1>Record NOT Updated</h1>
<?php
mysqli_close($conn);
 ?>
```

OUTPUT:

Students Infromation

The Given number is:104The Given number is:UmarThe Given number is:maleThe Given number is:2024-03-03The Given number is:PHPUPdate Successfull

Module 10:
a) Develop a PHP web application which demonstrates how to upload image into database and display it.

1. Database Setup

- **Create a database:**
  - Use a tool like phpMyAdmin or MySQL Workbench to create a new database (e.g., "image_upload").
- **Create a table:**
  - Create a table named "images" with the following structure:

SQL

```
CREATE TABLE images (
    id INT AUTO_INCREMENT PRIMARY KEY,
    image_name VARCHAR(255) NOT NULL,
    image_path VARCHAR(255) NOT NULL
);
```

2. PHP Code (upload_image.php)

```
<?php

// Database credentials
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "sss";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}

if ($_SERVER["REQUEST_METHOD"] == "POST") {
   // Get image file details
   $target_dir = "uploads/"; // Directory to save uploaded files
   $target_file = $target_dir . basename($_FILES["image"]["name"]);
   $uploadOk = 1;
   $imageFileType = strtolower(pathinfo($target_file,PATHINFO_EXTENSION));

   // Check if image file is a actual image or fake image
   $check = getimagesize($_FILES["image"]["tmp_name"]);
   if($check !== false) {
      $uploadOk = 1;
   } else {
      echo "File is not an image.";
      $uploadOk = 0;
```

```
    }

    // Check if file already exists
    if (file_exists($target_file)) {
        echo "Sorry, file already exists.";
        $uploadOk = 0;
    }

    // Check file size
    if ($_FILES["image"]["size"] > 500000) { // 500KB limit
        echo "Sorry, your file is too large.";
        $uploadOk = 0;
    }

    // Allow certain file formats
    if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg"
    && $imageFileType != "gif" ) {
        echo "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";
        $uploadOk = 0;
    }

    // Check if $uploadOk is set to 0 by an error
    if ($uploadOk == 0) {
        echo "Sorry, your file was not uploaded.";
    // if everything is ok, try to upload file
    } else {
        if (move_uploaded_file($_FILES["image"]["tmp_name"], $target_file)) {
            // Insert image path into database
            $image_name = basename($_FILES["image"]["name"]);
            $image_path = $target_file;
            $sql = "INSERT INTO images (image_name, image_path) VALUES ('$image_name', '$image_path')";
            if ($conn->query($sql) === TRUE) {
                echo "New record created successfully";
            } else {
                echo "Error: " . $sql . "<br>" . $conn->error;
            }
        } else {
            echo "Sorry, there was an error uploading your file.";
        }
    }
}

?>

<!DOCTYPE html>
<html>
<head>
<title>Image Upload</title>
</head>
<body>

<h2>Upload Image</h2>

<form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>" method="post" enctype="multipart/form-
data">
    Select image to upload:
    <input type="file" name="image" id="image">
```

```
  <input type="submit" value="Upload Image" name="submit">
</form>

<h2>Uploaded Images</h2>

<?php
// Retrieve images from database
$sql = "SELECT * FROM images";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
  // Output data of each row
  while($row = $result->fetch_assoc()) {
    echo "<img src='" . $row["image_path"] . "' width='200' height='200'><br>";
  }
} else {
  echo "0 results";
}

$conn->close();
?>

</body>
</html>
```

3. Output

- **Upload Image Form:**
  - A form will be displayed where you can select an image file to upload.
- **Upload Success/Failure:**
  - Upon clicking "Upload Image," the script will process the upload.
  - If successful, it will display a success message.
  - If there are errors (e.g., invalid file type, file size limit exceeded), it will display appropriate error messages.
- **Uploaded Images:**
  - Below the upload form, the script will display thumbnails of the successfully uploaded images.