

LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING (AUTONOMOUS)



Department of Computer Science & Engineering

20CS63 – SERVER-SIDE SCRIPTING LAB

Name of the Student: _____

Registered Number: _____

Branch & Section: _____ &/Sec

Academic Year: 2022-2023

LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING (AUTONOMOUS)



CERTIFICATE

This is to certify that this is a bonafide record of the practical workdone

by Mr./Ms.,

bearing Regd. Num.: 20761A05..... of B.Tech..... Semester, Branch,

Section in the 20CS59 - OPERATING SYSTEMS LAB during the Academic

Year: 2021-2022. -

No. of Experiments/Modules held: 09

No. of Experiments Done: 09

Date: / ____ / 2023

Signature of the Faculty

INTERNAL EXAMINER

EXTERNAL EXAMINER

INDEX

S.NO	Modules	Dates
1	Modul-1	29-12-2022 & 5-1-2023
2	Module-2	12-1-2023
3	Module-3	19-1-2023 & 2-2-2023
4	Module-4	9-2-2023
5	Module-5	16-2-2023 & 2-3-2023
6	Module-6	9-3-2023
7	Module-7	16-3-2023
8	Module-8	0.-3-2023
9	Module-9	6-4-2023

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 (Max 25 Characters Allowed)

Password (Max 25 Characters Allowed)

Confirm Password (Max 25 Characters Allowed)

Email ID

Mobile Number (10 Digits Allowed)

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

StuReg.jsp

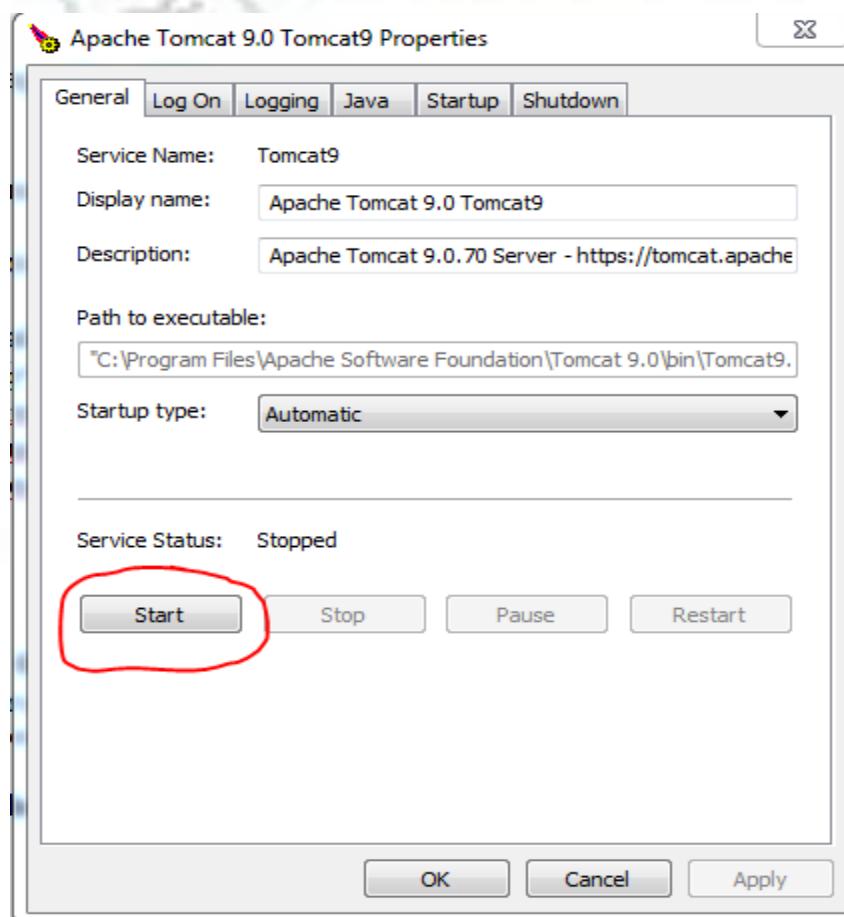
```
<%@ page language="java" %>
<html>
<head>
<title>JSP Response Page</title>
</head>
<body bgcolor="pink">
<center>
<h1>Student Details are</h1>
User Name is :<%=request.getParameter("un")%> <br>
Password is :<%=request.getParameter("pd")%><br>
Email is :<%=request.getParameter("eid")%><br>
Mobile 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>

The screenshot shows a web browser window with the URL localhost:8080 in the address bar. The page title is "Apache Tomcat/9.0.70". A green banner at the top says "If you're seeing this, you've successfully installed Tomcat. Congratulations!". Below the banner is a cartoon cat icon. To the right of the cat are links for "Recommended Reading" including "Security Considerations How-To", "Manager Application How-To", and "Clustering/Session Replication How-To". On the far right, there are three buttons: "Server Status", "Manager App", and "Host Manager", with "Manager App" circled in red. The footer contains a "Developer Quick Start" section with links to "Tomcat Setup", "First Web Application", "Realms & AAA", "JDBC DataSources", "Examples", "Servlet Specifications", and "Tomcat Versions".

Click on Manager App then Enter UserName & Password

The screenshot shows the same Apache Tomcat 9.0.70 homepage as above, but with a sign-in dialog box overlaid. The dialog box has "localhost:8080" in the title bar and asks "This site is asking you to sign in." It contains two input fields: "Username" with "admin" and "Password" with "*****". There are "Sign in" and "Cancel" buttons. The rest of the page is dimmed, indicating it is not interactive while the dialog is open.

Browse the .war file & click on Deploy Button

The screenshot shows the Tomcat Web Application Manager interface. At the top, there's a message bar stating "ok - Undeployed application at context path [/WEBAPP]". Below it is a navigation bar with tabs for "Manager", "List Applications", "HTML Manager Help", "Manager Help", and "Server Status". The main area is titled "Applications" and lists several contexts: "/", "/LABEXE", "/docs", "/examples", "/host-manager", and "/manager". Each context has columns for "Path", "Version", "Display Name", "Running", "Sessions", and "Commands". In the "Deploy" section, there are fields for "Context Path", "Version (for parallel deployment)", "XML Configuration file path", and "WAR or Directory path". A "Deploy" button is located below these fields. A red circle highlights the "Browse..." button in the "WAR or Directory path" field.

This screenshot shows the same Tomcat Web Application Manager interface after a deployment. The message bar now says "OK". The "Applications" list includes the newly deployed context "/WEBAPP". A red circle highlights the "OK" message in the message bar, and another red circle highlights the "/WEBAPP" entry in the Applications list.

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

Student Registration Form Using Table in HTML

User Name (Max 25 Characters Allowed)

Password (Max 25 Characters Allowed)

Confirm Password (Max 25 Characters Allowed)

Email ID

Mobile Number (10 Digits Allowed)

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
}

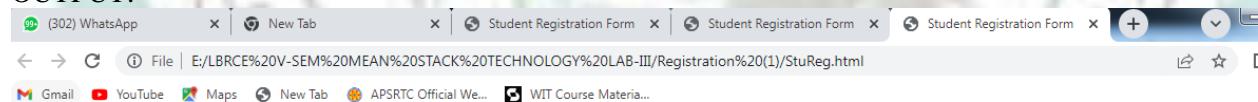
input[type=text], input[type=email], input[type=number]{
    width: 50%;
    padding: 6px 12px;
    margin: 5px 0;
    box-sizing: border-box;
}

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	<input type="text"/> User Name (Max 25 Characters Allowed)
Password	<input type="password"/> Password (Max 25 Characters Allowed)
Confirm Password	<input type="password"/> Confirm Password (Max 25 Characters Allowed)
Email ID	<input type="text"/> Sample@gmail.com
Mobile Number	<input type="text"/> 1234567890 (10 Digits Allowed)
<input type="button" value="Clear Form"/> <input type="button" value="Validate"/>	

This page says
Please enter your name

OK

User Name (Max 25 Characters Allowed)

Password (Max 25 Characters Allowed)

Confirm Password (Max 25 Characters Allowed)

Email ID Sample@gmail.com

Mobile Number 1234567890 (10 Digits Allowed)

Clear Form Validate

This page says
Please enter Password

OK

User Name (Max 25 Characters Allowed)

Password (Max 25 Characters Allowed)

Confirm Password (Max 25 Characters Allowed)

Email ID Sample@gmail.com

Mobile Number 1234567890 (10 Digits Allowed)

Clear Form Validate

[Student Registration Form Using Table in HTML](#)

Nagesh

User Name Nagesh (Max 25 Characters Allowed)

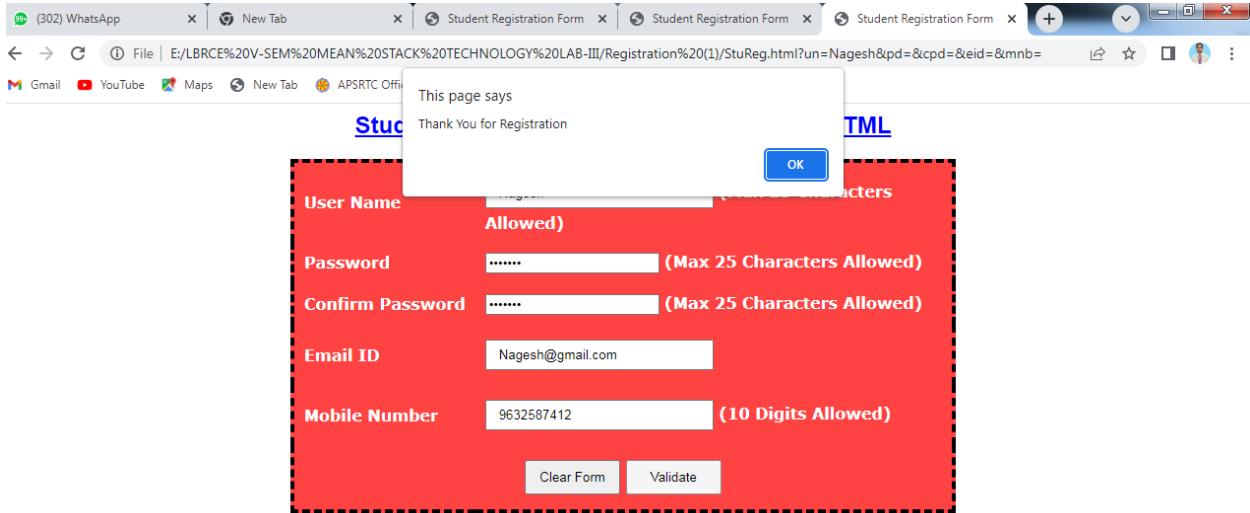
Password (Max 25 Characters Allowed)

Confirm Password (Max 25 Characters Allowed)

Email ID Nagesh@gmail.com

Mobile Number 9632587412 (10 Digits Allowed)

Clear Form Validate



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:\UI-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**](http://localhost:8080/LABPROGRAMS/s2.html)



The XMLHttpRequest Object

Click Here to Get the Data From 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)

- 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>
```

```
</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

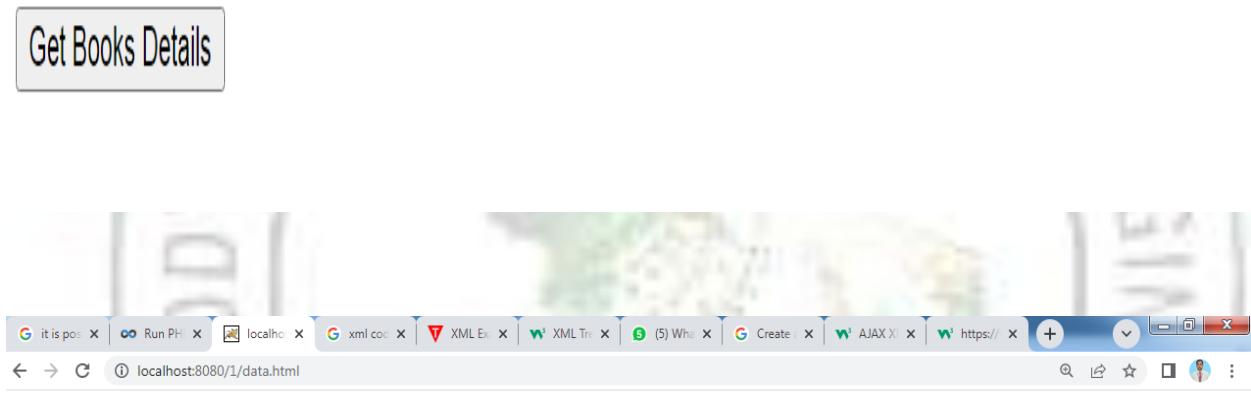
C:\2b>**jar cvf 2b.war .**

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



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>
```



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

Start typing a name in the input field below:

First name:

Suggestions:

Start typing a name in the input field below:

First name: v

Suggestions: Violet, Vicky

Start typing a name in the input field below:

First name: e

Suggestions: Eva, Eve, Evita, Elizabeth, Ellen

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
<code>\$_GET</code>	Contains a list of all the field names and values sent by a form using the get method
<code>\$_POST</code>	Contains a list of all the field names and values sent by a form using the post method
<code>\$_REQUEST</code>	Contains the values of both the <code>\$_GET</code> and <code>\$_POST</code> arrays combined, along with the values of the <code>\$_COOKIE</code> 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.

```
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">
```

registration.html

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

registration.php

OUTPUT:



Name: Email: Submit



Welcome CSEPHP
! Your email address is CSEPHP32@GMAIL.COM

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.

```
<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.html

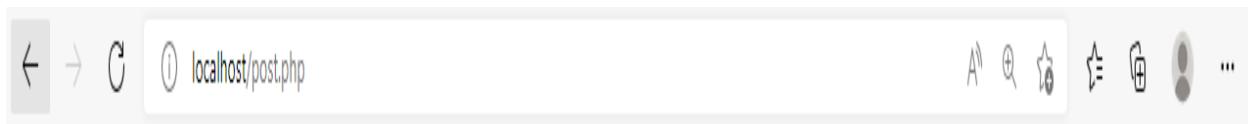
```
<html>  
  <body>  
    Welcome <?php echo $_POST["name"]; ?>  
    <br>  
    ! Your email address is <?php echo $_POST["email"]; ?>  
  </body>  
</html>
```

post.php



\$_POST Variable Example Program

Name:
Email:



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.

Advantage

- **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';
```

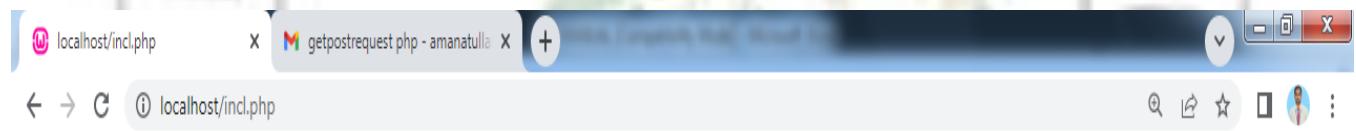
Or

```
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>
```

```
<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" ?>
```

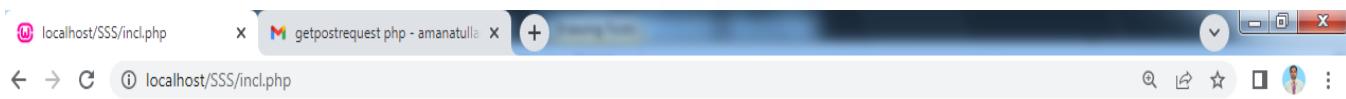
OUTPUT:

The Following URLs are Reglarling using for searching & Buying

[Google](#) | [Yahoo](#) | [FlipKart](#) | [Amazon](#) | [BigBasket](#) | Sample include
[Google](#) | [Yahoo](#) | [FlipKart](#) | [Amazon](#) | [BigBasket](#) | Sample require

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

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

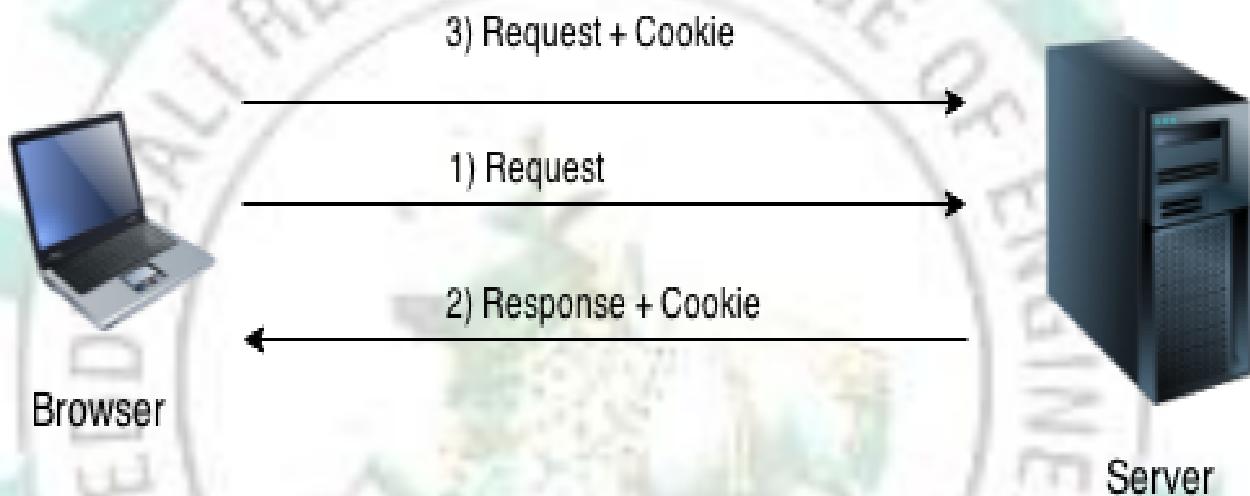


I am a student of III , CSE , LBRCE . I am a student of III , CSE , LBRCE .

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 to 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:

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

</body>
</html>
```

links.html

```
<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");
    </?php

```

```

?>

<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";

?>

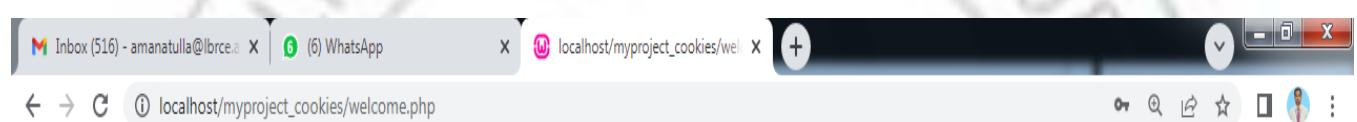
```

OUTPUT:

[Login](#) | [View Profile](#) | [Logout](#)

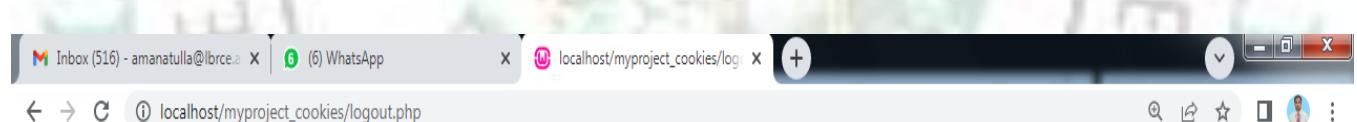
Enter User Name:

Enter Password:



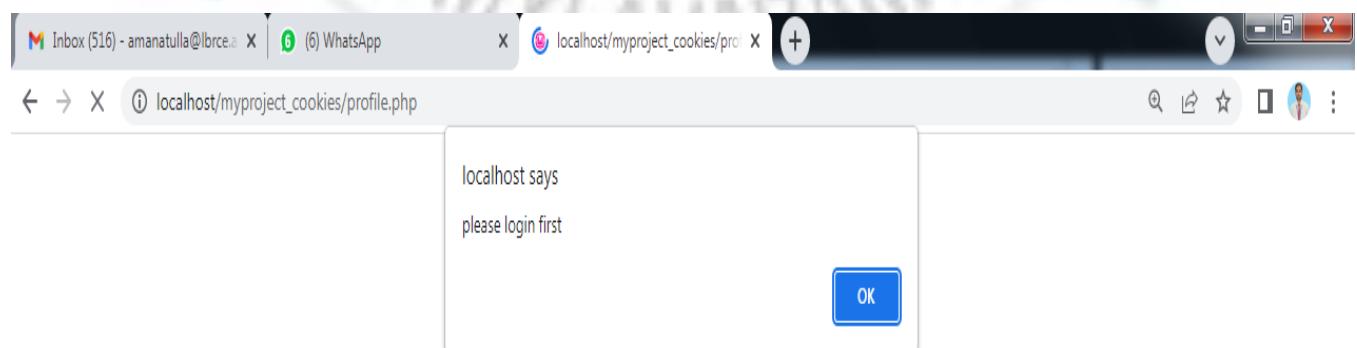
[Login](#) | [View Profile](#) | [Logout](#)

Welcome to cse



[Login](#) | [View Profile](#) | [Logout](#)

Logout successfully



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.

Realworld 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 websites, shopping cart is generally 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

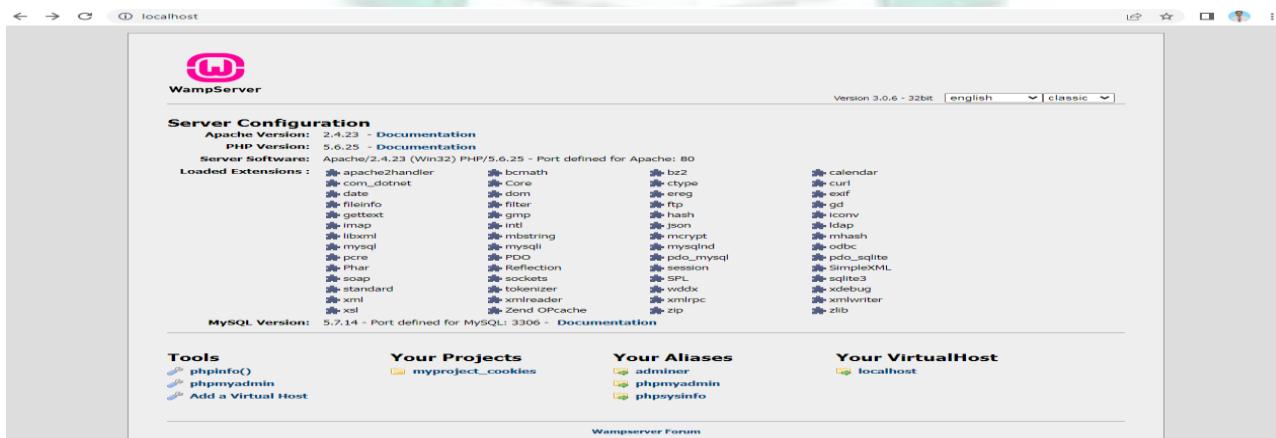
In the code above, we have started a session and set two session variables. Above webpage will also have a link to navigate to Second page **second_page.php**.

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>
```

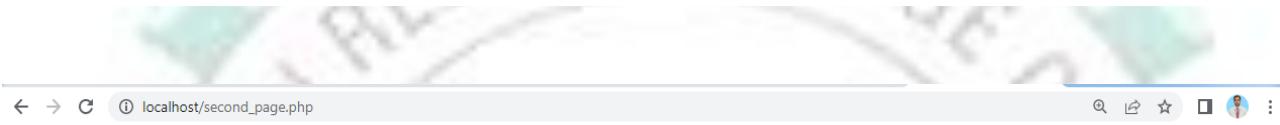
OUTPUT:



The screenshot shows the WampServer control panel interface. At the top, it displays the WampServer logo and the version information: Version 3.0.6 - 32bit | english | classic. Below this, the 'Server Configuration' section is visible, showing Apache Version: 2.4.23 - Documentation, PHP Version: 5.6.25 - Documentation, and Server Software: Apache/2.4.23 (Win32) PHP/5.6.25 - Port defined for Apache: 80. A list of loaded extensions is provided, including: apache2handler, com_dotnet, data, dom, filter, gettext, imap, libxml, mbstring, mbstring_compatibility, pcntl, Phar, soap, standard, xml, xsl, bz2, Core, ctype, ereg, ftp, hash, json, ldap, mhash, mbstring, mysqli, pdo_mysql, session, SPL, wddx, Zend_Opcache, calendar, curl, exif, gd, iconv, ldap, mbstring, pdo_sqlite, SimpleXML, sqlite3, xdebug, xmlwriter, zip. At the bottom of the configuration section, it says MySQL Version: 5.7.14 - Port defined for MySQL: 3306 - Documentation. The footer of the page includes links for Tools (phpinfo(), phpmyadmin, Add a Virtual Host), Your Projects (myproject_cookies), Your Aliases (administrator, phpmyadmin, phpsysinfo), Your VirtualHost (localhost), and Wampserver Forum.

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:

localhost/print_all_session.php



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>
```

Before Update User Name & User Id

User Name is: cse

User Id is: 501

After Update User Name & User Id

User Name is: cse

User Id is: 511

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 destroy the session, we use `session_destroy()` function.

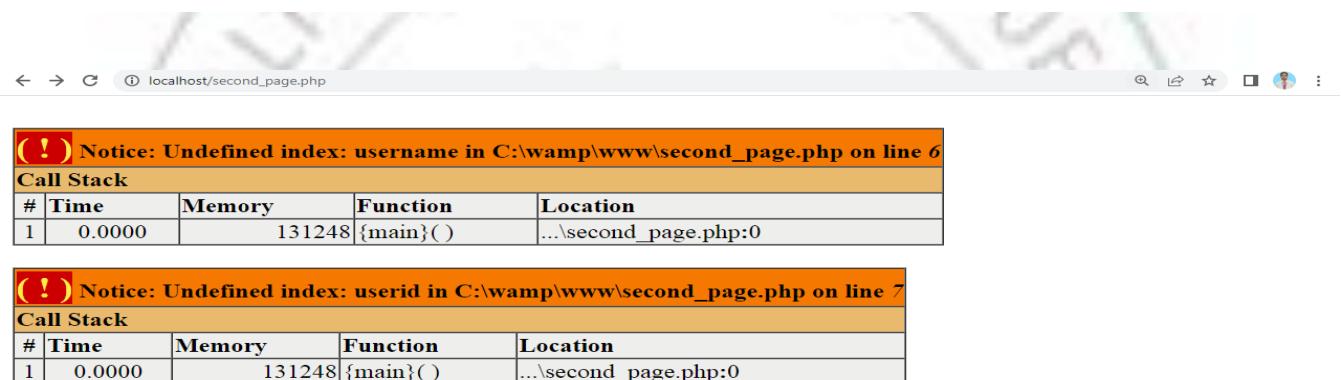
session_destroy.php

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

OUTPUT:



Session Destroy Successfully



**User Name is:
User Id is:**

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.

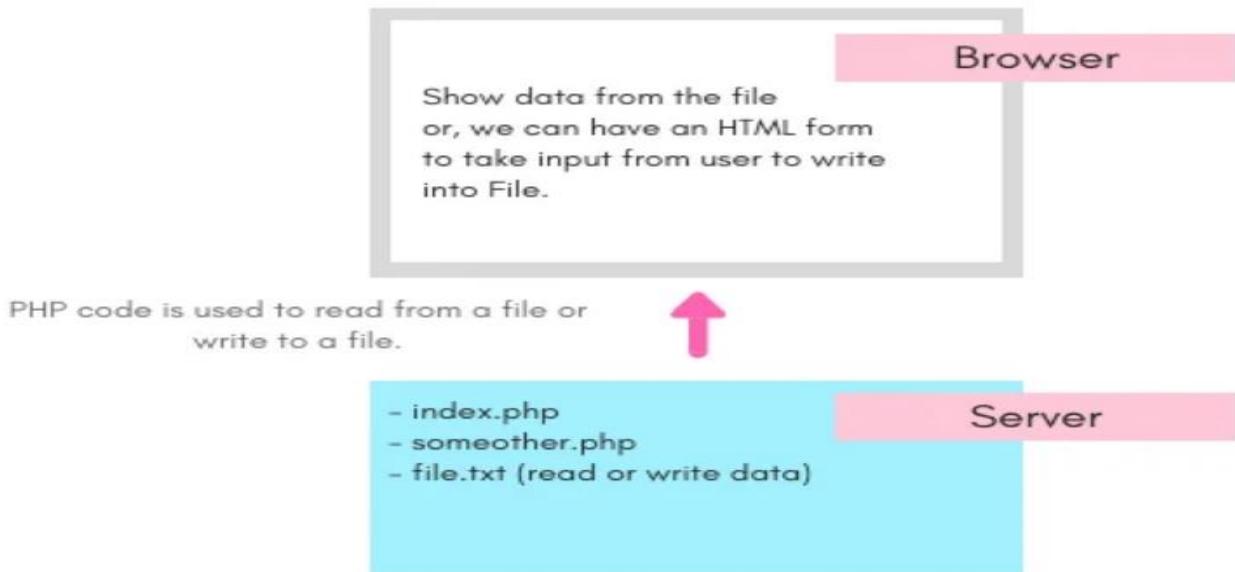
File Handling Operations

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.



File Handling in PHP

Uses of File in Applications

Following are some of the practical use cases where we need files in our web applications.

- Oftenly data is stored in for of JSON files, amd Php code has to read the file and then display the data on web page.
- In some simple applications, no database is used, rather data is stored in files.
- In some web applications, you may have to prepare a file for user to download, in that case PHP code will create a file, write data in it and then allow the user to download it.

Create, Open and Close a File in PHP

To create a new file or to open an existing file, we use the `fopen()` function provided by PHP and to close a file resource `fclose()` function is used.

Create a File with PHP

When we use the function `fopen()` to open a file which doesn't exist then that file gets created.

Let's take an example,

```
$myfile = 'file.txt';  
//opens the file.txt file or implicitly creates the file  
$handle = fopen($myfile, 'w') or die('Cannot open file: '.$myfile);
```

`fopen()` function takes the **filename** as the first argument and the second argument represents the **mode** in which the file will be opened.

Open a File with PHP

To open a file, the same `fopen()` function is used. There can be many reasons behind opening a file, like reading content of the file, writing new content to a file or appending additional content to the already existing content in the file.

Let's take a simple example, and then we will talk about the different modes in which a file can be opened.

```
$myfile = 'file.txt';  
//opens the file.txt file  
$handle = fopen($myfile, 'w') or die('Cannot open file: '.$myfile);
```

argument in the `fopen()` function.

Mode	String Literal	Description
Write mode	w	If the file exists then it is opened to allow write operation in it and if it doesn't exist, then a new file is created. In this mode all

		the existing data in file gets erased.
Read mode	r	File is opened in read mode with the file pointer starting from the beginning of the file.
Append mode	a	File is opened in write mode where existing content of the file is note erased. The new content is added after the existing content.
Create Write-only file	x	A new file is created with write-only access. If the file already exists then error is returned.

Apart from the mode specified above, we can add `+` along with the string literals to allow both read and write(default) operation for a given mode.

For example, `r+` mode opens file for read and write both. `w+, a+` allow read operation on the file along with the default write and append operations respectively.

Technically, file is not opened, `fopen()` binds the resource(file) to a stream, which can then be used to read from the file or write data to the file.

Also, the **filename** should be fully qualified name along with relative path if the file is a local file. The **filename** can also be a URL to specify a remote file's path. In that case, once PHP realise that the file path is not local, it will check for the value of `allow_url_fopen` property in the **php.ini**(PHP's configuration file). If it is **false**, PHP will print a warning and `fopen()` call will fail.

Close a File with PHP

It is good practice to close a file resource after using it. In PHP `fclose()` function is used to close a file resource. It takes the file name as argument or the variable holding the file resource pointer as argument. Let's take an example:

```
$myfile = 'file.txt';

//opens the file.txt file or implicitly creates the file

$handle = fopen($myfile, 'w') or die('Cannot open file: '.$myfile);

// closing a file

fclose($handle);
```

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.

- o 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.
- o 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.
- o 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

Start typing a name in the input field below:

First name:

Suggestions:

Start typing a name in the input field below:

First name: v

Suggestions: Violet, Vicky

Start typing a name in the input field below:

First name: e

Suggestions: Eva, Eve, Evita, Elizabeth, Ellen

Module 8:

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

REGISTRATION.HTML

```
<!DOCTYPE html>
<html>
<head>
    <title>Reg Form</title>
</head>
<body>
| <a href="Registration.html">Registration</a> | <a href="login.html"> Login </a> | <a
href="view.html"> View </a> | <a href="viewall.html"> ViewAll </a> | <a href="update.html">
Update </a> | <a href="delete.html"> Delete </a>
<center><h1>Student Registration Form</h1></center>
```

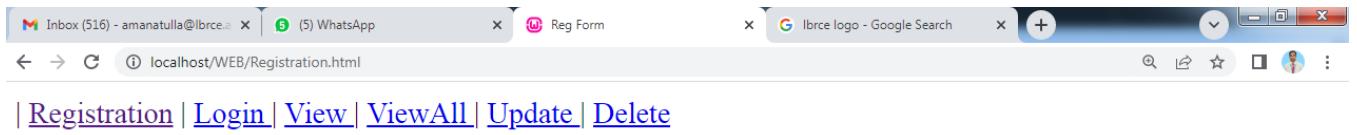
```
<hr>
<form method="post" action="reg.php" name="reg_form">
    <center>
        <h2>Registration Form</h2>
    <table>
        <tr>
            <td><label>Enter Reg.No: </label></td>
            <td>
                <input type="text" name="regno" placeholder="Enter Reg.NO">
            </td>
        </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="female">Female
            </td>
        </tr>
        <tr>
            <td><label>DOB: </label></td>
            <td>
                <input type="date" name="dt">
            </td>
        </tr>
        <tr>
            <td><label>Email Id: </label></td>
            <td>
                <input type="text" name="email" placeholder="example@gmail.com">
            </td>
        </tr>
        <tr>
            <td><label>Password: </label></td>
            <td>
                <input type="password" name="pwd" placeholder="enter password">
            </td>
        </tr>
        <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><label>Mobile: </label></td>
    <td>
        <input type="number" name="mobile">
    </td>
</tr>
<tr>
    <td><label>Address: </label></td>
    <td>
        <input type="textarea" size="50" name="address" placeholder="Address">
    </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:



Student Registration Form

Registration Form

Enter Reg.No:	<input type="text" value="Enter Reg.NO"/>
Enter Name:	<input type="text" value="Enter Name"/>
Gender:	<input type="radio"/> Male <input type="radio"/> Female
DOB:	<input type="text" value="mm / dd / yyyy"/> <input type="button" value=""/>
Email Id:	<input type="text" value="example@gmail.com"/>
Password:	<input type="text" value="enter password"/>
Course:	<input type="text" value="select course"/>
Mobile:	<input type="text"/>
Address:	<input type="text" value="Address"/>
<input type="button" value="Submit"/> <input type="button" value="Reset"/>	

Reg.php

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

```
$adr=$_POST['address'];

$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());

}

$sql = "INSERT INTO student_info (regno,sname,gender,dob,email,
password,course,mobile,address)VALUES
('$rno','$name','$gen','$dob','$eid','$pd','$crs','$mno','$adr')";

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:

Inbox (516) - amanatulla@lbrce.e... x (5) WhatsApp x Reg Form x lbrce logo - Google Search x +

localhost/WEB/Registration.html

| [Registration](#) | [Login](#) | [View](#) | [ViewAll](#) | [Update](#) | [Delete](#)

Student Registration Form

Registration Form

Enter Reg.No:

Enter Name:

Gender: Male Female

DOB:

Email Id:

Password:

Course:

Mobile:

Address:

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.html

```
<!DOCTYPE html>
<html>
<head>
<title>Viewall Form</title>
```

```
</head>

<body>

<a href="Registration.html">Registration</a> | <a href="login.html"> Login </a> | <a href="view.html"> View </a> | <a href="viewall.html"> ViewAll </a> | <a href="update.html"> Update </a> | <a href="delete.html"> Delete </a>

<center><h1>Display all the Records</h1></center>

<hr>

<form method="post" action="viewall.php" name="view_form">

    <center>

        <h2>View Form</h2>

        <table>

            <tr>

                <td>

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

                    <input type="reset" name="reset" value="Reset">

                </td>

            </tr>

        </table>

    </center>

</form>

</body>

</html>
```

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 Information</h1>  
  
<table align="center" border="5">  
  
<tr>  
  
<th>Reg.No</th>  
  
<th>Student Name</th>  
  
<th>Gender</th>  
  
<th>DOB</th>
```

```
<th>Email</th>
<th>course</th>
<th>mobile</th>
<th>Address</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 $row["emailid"]?> </td>
```

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

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

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

```
</tr>
```

```
<?php
```

```
}
```

```
?>
```

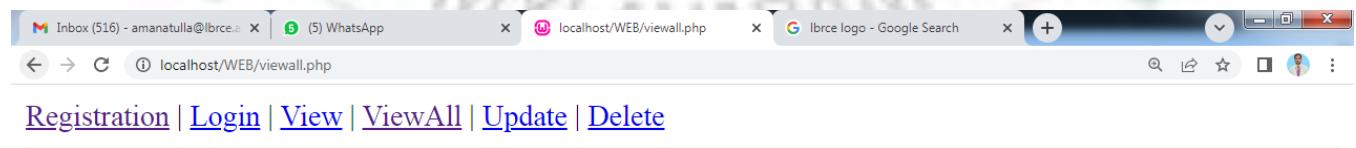
```
</table>
```

```
<?php
```

```
mysqli_close($conn);
```

```
?>
```

OUTPUT:



A screenshot of a web browser window. The address bar shows 'localhost/WEB/viewall.php'. The page content includes navigation links: 'Registration | Login | View | ViewAll | Update | Delete'. Below this, the title 'Students Information' is displayed in a large, bold font. A table with student data is shown.

Students Information

Reg.No	Student Name	Gender	DOB	Email	course	mobile	Address
ASDF123	YASEEN	male	2007-06-13		CSS	369258741	lbrce
20761A0501	AMANATULLA	male	1998-06-05		PHP	963852741	lbrce

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.



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

To store the data inside the MySQL database we need to access the phpMyAdmin by following the link references to the <http://localhost/phpmyadmin/>, By providing Username: **root** and Password: "" (Empty)

It will be looked like below

The screenshot shows the phpMyAdmin interface with the following sections:

- General settings:** Server connection collation: utf8mb4_unicode_ci
- Appearance settings:** Language: English, Theme: pmahomme
- Database server:**
 - Server: 127.0.0.1 via TCP/IP
 - Server type: MariaDB
 - Server connection: SSL is not being used
 - Server version: 10.4.27-MariaDB - mariadb.org binary distribution
 - Protocol version: 10
 - User: root@localhost
 - Server charset: UTF-8 Unicode (utf8mb4)
- Web server:**
 - Apache/2.4.54 (Win64) OpenSSL/1.1.1p PHP/8.1.12
 - Database client version: libmysql - mysqld 8.1.12
 - PHP extension: mysqli curl mbstring
 - PHP version: 8.1.12
- phpMyAdmin:**
 - Version information: 5.2.0, latest stable version: 5.2.1
 - Documentation
 - Official Homepage
 - Contribute
 - Get support
 - List of changes

To create a new database click on the “New” button as shown in below figure

The screenshot shows the phpMyAdmin interface with a red arrow pointing to the "New" button in the sidebar. The sidebar also lists existing databases: information_schema, lbrce, mysql, performance_schema, phpmyadmin, and test.

Now to create the database, give some database name as “lbrce” and click on **Create** button to create the database.

To open the created database find the database name in the left column menu and click on the database name,

The screenshot shows the phpMyAdmin interface for the 'lbrce' database. The left sidebar lists databases: information_schema, lbrce, New, mysql, performance_schema, phpmyadmin, and test. The 'student_info' table is highlighted with a red arrow. The main area displays the table structure with two rows and two columns. Below the table, there is a 'Create new table' section.

Now to create a new table in the database click on the new button available below the database name,
Now give the table details like below to create a sample table

The screenshot shows the phpMyAdmin interface for the 'student_info' table within the 'lbrce' database. The left sidebar shows the database structure. The main area displays the table structure with 9 columns: regno, sname, gender, dob, emailid, password, mobileno, course, and address. Each column has its type, collation, attributes, and other details listed.

This will create a student_info table in the lbrce database.

To store the data we need to create registration.html file as shown in the code below

```
1.      <!DOCTYPE html>
2.      <html lang="en">
3.          <head>
4.              <meta charset="UTF-8">
5.              <meta http-equiv="X-UA-Compatible" content="IE=edge">
6.              <meta name="viewport" content="width=device-width, initial-scale=1.0">
7.              <title>Reg Form</title>
8.              <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJOZ" crossorigin="anonymous">
9.              <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ENjd04Dr2bkBIFxQpeoTz1Hicje39Wm4jDKdf19U8gI4ddQ3GYNS7NTKfAdVQSze" crossorigin="anonymous"></script>
10.             <style type="text/css">
11.                 td {
12.                     padding: 5px 15px;
13.                     text-align: left;
14.                 }
15.                 input[type="radio"]{
16.                     margin: 0 5px 0 5px;
17.                 }
18.             </style> 19.
20.         </head>
21.         <body>
22.             <a href="registration.html">Registration</a>|
23.             <a href="login.html">Login</a>|
24.             <a href="view.html">View</a>|
25.             <a href="viewall.html">View All</a>|
26.             <a href="update.html">Update</a>|
27.             <a href="delete.html">Delete</a>
28.             <h1 style="text-align: center;">Student Registration Form</h1> 29.      <hr>
30.             <h3 style="text-align: center;" class="pb-3">Registration Form</h3>
31.             <form style="text-align: center;" name="regform" action="reg.php" method="post">
32.                 <table align="center">
33.                     <tr><td><label for="rollno">Enter Reg.No:</label></td><td><input type="text" name="rno" id="rollno" placeholder="Enter Reg.No"></td></tr>
34.                     <tr><td><label for="uname">Enter Name:</label></td><td><input type="text" name="uname" id="uname" placeholder="Enter Name"></td></tr>
35.                     <tr><td><label for="gender">Gender:</label></td><td><input type="radio" name="ugender" id="male" value="male"><label for="male" class="mr-3">Male</label><input type="radio" name="ugender" id="female" value="female"><label for="female">Female</label></td></tr>
36.                     <tr><td><label for="dob">DOB:</label></td><td><input type="date" name="dob" id="dob"></td></tr> 37.
37.                     <tr><td><label for="email">Email ID:</label></td><td><input type="email" name="umail" id="email" placeholder="example@domain.com"></td></tr>
38.                     <tr><td><label for="pwd">Password:</label></td><td><input type="password" name="pwd" id="pwd" placeholder="Enter password"></td></tr>
39.                     <tr><td><label for="mobile">Mobile:</label></td><td><input type="number" name="mobile" id="mobile"></td></tr>
40.                     <tr><td><label for="course">Course:</label></td><td><select id="course" name="course">
41.                         <option value="javascript">JavaScript</option>
42.                         <option value="php">PHP</option>
43.                         <option value="css">CSS</option>
44.                         <option value="html">HTML</option>
45.                     </select></td></tr>
46.                     <tr><td><label for="address">Address:</label></td><td><textarea name="address" id="address" placeholder="Address"></textarea></td></tr>
47.                 </table>
48.                 <br>
49.                 <button type="submit" class="btn btn-primary mr-3">Submit</button>
50.                 <button type="reset" class="btn btn-warning">Reset</button>
51.             </form>
52.         </body>
53.     </html> 54.
```

This will show the webpage of output below, Contains the Student Registration Form containing some fields to store the data inside the database, This page has the form with method = “POST” and action = “reg.php” to store the data inside the database.

[Registration](#) | [Login](#) | [View](#) | [View All](#) | [Update](#) | [Delete](#)

Student Registration Form

Registration Form

Enter Reg.No:	<input type="text" value="Enter Reg.No"/>
Enter Name:	<input type="text" value="Enter Name"/>
Gender:	<input type="radio"/> Male <input type="radio"/> Female
DOB:	<input type="text" value="dd/mm/yyyy"/> 
Email ID:	<input type="text" value="example@domain.com"/>
Password:	<input type="password" value="Enter password"/>
Mobile:	<input type="text"/>
Course:	<input type="button" value="JavaScript"/>
Address:	<input type="text" value="Address"/>

reg.php

```
1.      <!DOCTYPE html>
2.      <html>
3.          <head>
4.              <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJOZ" crossorigin="anonymous">
5.              <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ENjd04Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYN57NTKfAdVQSZe" crossorigin="anonymous"></script>
6.          </head>
7.      </html>  8.
8.      <?php
9.          include("links.html");
10.         $rno = $_POST['rno'];
11.         $name = $_POST['uname'];
12.         $gen = $_POST['ugender'];
13.         $dob = $_POST['dob'];
14.         $eid = $_POST['umail'];
15.         $pd = $_POST['pwd'];
16.         $mno = $_POST['mobile'];
17.         $crs = $_POST['course'];
18.         $adr = $_POST['address'];
19.         $servername = "localhost"; // local server name default
20.         $username = "root"; // mysql username default is root
21.         $password = ""; // blank if no password is set for mysql
22.         $dbname = "lbrece";
23.         //Create connection
24.         $conn = mysqli_connect($servername,$username,$password,$dbname);
25.         // check connection
26.         if(!$conn){
27.             die("Connection failed:".mysqli_connect_error());
28.         }
29.         $sql = "INSERT INTO student_info
30.             (regno,sname,gender,dob,emailid,password,mobileno,course,address)
31.             VALUES('$rno','$name','$gen','$dob','$eid','$pd','$mno','$crs','$adr')"; 33.
32.         if(mysqli_query($conn,$sql)){
33.             echo "<center><h1>New Record created successfully</h1></center>"; 36.      } else{
34.                 echo "Error: " . $sql . "<br>" . mysqli_error($conn); 38.
35.             }
36.         mysqli_close($conn);
37.     ?> 41.
```

After successfully submitted the form without errors this page will shows the output “New Record create successfully”

To check the data inside the table go to the browse section of the table, and check the inserted data

The screenshot shows the phpMyAdmin interface for a MySQL database named 'ibco'. The left sidebar lists databases like 'information_schema', 'ibco', 'mysql', 'performance_schema', 'phpmyadmin', and 'test'. The 'student_info' table is selected. The main area displays the table structure with columns: regno, sname, gender, dob, emailid, password, mobileno, course, and address. Two rows of data are present:

regno	sname	gender	dob	emailid	password	mobileno	course	address
20761A0589	Srikanth	male	2023-04-28	srikanthkandi@gmail.com	Sri123	123456789	php	Vijayawada
20761A05C2	Joshiith	male	2023-04-21	joshiith@gmail.com		12345	javascript	Vijayawada

Here we can see 2 rows are inserted into to table already without any errors.

- 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

To see the data inside the table using PHP from MySQL database we need to create some basic form with a button and onclick action using JavaScript.

```
1.      <!DOCTYPE html>
2.      <html lang="en">
3.      <head>
4.          <meta charset="UTF-8">
5.          <meta http-equiv="X-UA-Compatible" content="IE=edge">
6.          <meta name="viewport" content="width=device-width, initial-scale=1.0">
7.          <title>Login</title>
8.          <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJOZ" crossorigin="anonymous">
9.          <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ENjd04Dr2bkBIFxQpeoTz1HICje39Wm4jDKdf19U8gI4ddQ3GYN57NTKfAdVQSze" crossorigin="anonymous"></script>
10.         <style type="text/css">
11.             td {
12.                 padding: 5px 15px;
13.                 text-align: left;
14.             }
15.         </style>
16.     </head>
17.     <body>
18.         <a href="registration.html">Registration</a>|
19.         <a href="login.html">Login</a>|
20.         <a href="view.html">View</a>|
21.         <a href="viewall.html">View All</a>|
22.         <a href="update.html">Update</a>|
23.         <a href="delete.html">Delete</a>
24.         <h1 align="center">Display all the Records</h1>
25.         <hr>
26.         <h3 style="text-align: center;" class="pb-3">View Form</h3>
27.         <form style="text-align: center;" action="viewall.php" method="POST" name="viewallform">
```

```
28.      <button type="submit" class="btn btn-primary mr-3 mt-3">View All</button>
29.      </form>
30.      </body>
31.      </html>
```

This form will show the output as below,

[Registration](#) | [Login](#) | [View](#) | [View All](#) | [Update](#) | [Delete](#)

Display all the Records

[View Form](#)

[View All](#)



viewall.php

```
1.      <!DOCTYPE html>
2.      <html>
3.      <head>
4.      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYeM/SXH301p5ILy+dN9+nJOZ" crossorigin="anonymous">
5.      <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ENjd0ODr2bkBIFxQpeoTz1HICje39Wm4jDKdf19U8gI4ddQ3GYN57NTkfAdVQSZe" crossorigin="anonymous">/</script> 6. </head> 7.
8.      <?php
9.      include("links.html");
10.     $servername="localhost"; //local server name default localhost 11.      $username="root"; //mysql
11.     $username default is root.
12.     $password=""; //blank if no password is set for mysql.
13.     $dbname = "lbrece";
14.     // Create connection
15.     $conn = mysqli_connect($servername, $username, $password,$dbname); // Check connection
16.     if(!$conn)
17.     {
18.         die("Connection failed:".mysqli_connect_error());
19.     }
20.     $sql = "SELECT * FROM student_info";
21.     $result = mysqli_query($conn, $sql); 22.
22.         if (mysqli_num_rows($result) > 0)
23.         {
24.             // output data of each row
25.             ?>
26.             <body>
27.             <center><h3>Student Details</h3></center>
28.             <hr>
29.             <div class="container"><div class="row"><div class="col-2"></div><div class="col-8">
30.                 <table class="table table-striped">
31.                     <thead>
32.                         <th scope="col">Reg.No</th>
33.                         <th scope="col">Student Name</th>
34.                         <th scope="col">Gender</th>
35.                         <th scope="col">DOB</th>
36.                         <th scope="col">Email</th>
37.                         <th scope="col">Course</th>
38.                         <th scope="col">Mobile</th>
39.                         <th scope="col">Address</th>
40.                     </thead>
41.                     <tbody>
42.                         <?php
43.                         while($row = mysqli_fetch_assoc($result))
44.                         { ?>
45.                         <tr>
46.                             <td><?php echo $row["regno"]?> </td>
47.                             <td><?php echo $row["sname"]?> </td>
48.                             <td><?php echo $row["gender"]?> </td>
49.                             <td><?php echo $row["dob"]?> </td>
50.                             <td><?php echo $row["emailid"]?> </td>
51.                             <td><?php echo $row["course"]?> </td>
52.                             <td><?php echo $row["mobileno"]?> </td>
53.                             <td><?php echo $row["address"]?> </td>
54.                         </tr>
55.                         <?php 57.           }
56.                     ?></tbody></table>
57.                 </div>
58.                 <div class="col-2"></div>
59.             </div>
60.             </div>
61.             </div>
62.             </div>
63.             </body><?php
64.         }
65.     else
66.     { ?>
```

```
67.      <script>
68.      alert("Invalid Register Number");
69.      </script>
70.      <?php
71.      include("view.html"); 72.      } 73.
74.      mysqli_close($conn);
75.      ?>
76.      </html> 77.
```

By clicking on the “View All” button the action will be called to “viewall.php” with method “POST” which will show the table output as below

[Registration](#) | [Login](#) | [View](#) | [View All](#) | [Update](#) | [Delete](#)

Student Details

Reg.No	Student Name	Gender	DOB	Email	Course	Mobile	Address
20761A0589	Srikanth	male	2023-04-28	srikanthkandi@gmail.com	php	123456789	Vijayawada
20761A05C2	Joshith	male	2023-04-21	joshith@mail.com	javascript	12345	Vijayawada

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

Here for the demonstration of the application, we are deleting the record based on the roll number because it acts as a primary key.

delete.html

```

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.   <meta charset="UTF-8">
5.   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6.   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7.   <title>Delete</title>
8.   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KK94CHFLLe+nY2dmCWGm91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJOZ" crossorigin="anonymous">
9.   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ENjd04Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNs7NTkFAdVQSze" crossorigin="anonymous"></script>
10.  <style type="text/css">
11.    td {
12.      padding: 5px 15px;
13.      text-align: left;
14.    }
15.  </style>
16. </head>
17. <body>
18.   <a href="registration.html">Registration</a>|
19.   <a href="login.html">Login</a>|
20.   <a href="view.html">View</a>|
21.   <a href="viewall.html">View All</a>|
22.   <a href="update.html">Update</a>|
23.   <a href="delete.html">Delete</a>
24.   <h1 align="center">Student Delete Form</h1>
25.   <hr>
26.   <h3 style="text-align: center;" class="pb-3">Delete Form</h3>
27.   <form style="text-align: center;" action="delete.php" method="POST" name="deleteform">
28.     <table align="center">
29.       <tr>
30.         <td>label for="rno">Enter Reg. No:</label></td>
31.         <td><input type="text" name="rno" placeholder="20761A05XX" id="rno"/></td>
32.       </tr>
33.     </table>
34.     <button type="submit" class="btn btn-primary mr-3 mt-3">Delete</button>
35.     <button type="reset" class="btn btn-warning mt-3">Reset</button>
36.   </form>
37. </body>
38. </html>
39.

```

delete.php

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KK94CHFLLe+nY2dmCWGm91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJOZ" crossorigin="anonymous">
5.   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ENjd04Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNs7NTkFAdVQSze" crossorigin="anonymous"></script>
6. </head>
7. </html>
8.
9. <?php
10.  include("links.html");
11.  $rno=$_POST['rno'];
12.  $servername="localhost"; //local server name default localhost
13.  $username="root"; //mysql username default is root.
14.  $password=""; //blank if no password is set for mysql.
15.  $dbname = "lbrce";

```

```

16. // Create connection
17. $conn = mysqli_connect($servername, $username, $password,$dbname);// Check connection
18. if (!$conn)
19. {
20.     die("Connection failed:".mysqli_connect_error());
21. }
22. $sql = "DELETE FROM student_info WHERE regno='$rno'";
23. if (mysqli_query($conn,$sql)){
24.     echo "<center><h1>Record Deleted Successfully</h1></center>";
25. }
26. else
27. { ?>
28.     <script>
29.         alert("Invalid Register Number");
30.     </script>
31.     <?php
32.         include("delete.html");
33.     }
34. mysqli_close($conn);
35. ?>
36.

```

The form will shows the output like this,

Registration| Login| View| View All| Update| Delete

Student Delete Form

Delete Form

Enter Reg. No:

Delete Reset

When a record is successfully deleted then the page displays the message, “Record Deleted Successfully”

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

update.html

```

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.     <meta charset="UTF-8">
5.     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6.     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7.     <title>Reg Form</title>
8.     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KK94CHFLLe+nY2dmCWGm91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJOZ" crossorigin="anonymous">
9.     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ENjd04Dr2bkBIFxQpeoTz1HICje39Wm4jDKdf19U8gI4ddQ3GYNS7NTkfAdVQSze" crossorigin="anonymous"></script>
10.    <style type="text/css">
11.        td {
12.            padding: 5px 15px;
13.            text-align: left;
14.        }
15.        input[type="radio"]{
16.            margin: 0 5px 0 5px;
17.        }
18.    </style>
19.
20. </head>
21. <body>

```

```

22. <a href="registration.html">Registration</a>
23. <a href="login.html">Login</a>
24. <a href="view.html">View</a>
25. <a href="viewall.html">View All</a>
26. <a href="update.html">Update</a>
27. <a href="delete.html">Delete</a>
28. <h1 style="text-align: center;">Student Updation Form</h1>
29. <hr>
30. <h3 style="text-align: center;" class="pb-3">Updation Form</h3>
31. <form style="text-align: center;" name="updateform" action="update.php" method="post">
32.   <table align="center">
33.     <tr><td><label for="rno">Enter Reg.No:</label></td><td><input type="text" name="rno" id="rno" placeholder="Enter Reg.No"></td></tr>
34.     <tr><td><label for="email">Email ID:</label></td><td><input type="email" name="umail" id="email" placeholder="example@domain.com"></td></tr>
35.     <tr><td><label for="mobile">Mobile:</label></td><td><input type="number" name="mobile" id="mobile"></td></tr>
36.   </table>
37.   <br>
38.   <button type="submit" class="btn btn-primary mr-3">Update</button>
39.   <button type="reset" class="btn btn-warning">Reset</button>
40. </form>
41. </body>
42. </html>
43.

```

The above code will show the output like below,

[Registration](#) | [Login](#) | [View](#) | [View All](#) | [Update](#) | [Delete](#)

Student Updation Form

Updation Form

Enter Reg.No:	<input type="text" value="Enter Reg.No"/>
Email ID:	<input type="text" value="example@domain.com"/>
Mobile:	<input type="text"/>
<input style="background-color: #007bff; color: white; border-radius: 5px; padding: 5px 10px; margin-right: 10px;" type="button" value="Update"/> <input style="background-color: #ffc107; color: white; border-radius: 5px; padding: 5px 10px;" type="button" value="Reset"/>	

update.php

```

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KK94CHFLLe+nY2dmCWGQ91rCGa5gtU4mk92HdvYeM/SXH301p5ILy+dN9+nJOZ" crossorigin="anonymous">
5. <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ENjd04Dr2bkBTxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NTkfAdVQSze" crossorigin="anonymous"></script>
6. </head>
7. </html>
8.
9. <?php
10. include("links.html");
11. $rno = $_POST['rno'];
12. $eid = $_POST['umail'];
13. $mno = $_POST['mobile'];
14. $servername = "localhost"; //local server name default localhost
15. $username = "root"; //mysql username default is root.
16. $password = ""; //blank if no password is set for mysql.
17. $dbname = "lrcce";
18. // Create connection
19. $conn = mysqli_connect($servername, $username, $password, $dbname); // Check connection
20. if (!$conn)
21. {
22.   die("Connection failed:".mysqli_connect_error());
23. }
24. $sql = "UPDATE student_info SET emailid='$eid', mobileno=$mno where regno='$rno'";
25. if ($conn->query($sql) === TRUE){
26. ?>
27.   <script>
28.     alert("Record Updated Successfully");
29.   </script>

```

```
30.      <center><h1>Record Updated Successfully</h1></center>
31.      <?php
32.      } else{
33.          echo "Error updating record: " . $conn->error;
34.      }
35.      mysqli_close($conn);
36. ?>
37.
```

After successfully updated the form, the page will show the message “Record Updated Successfully”