LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING (AUTONOMOUS)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

20CS63 - SERVER - SIDE SCRIPTING LAB

Name of the student : ....

Registered Number : ....

Branch and Section : ....

Academic Year : ....

LAKIREDDY BALI REDDY COLLEGE OF ENGINEERING (AUTONOMOUS)

This is to certify that this is a bonafide record of the practical work done by Mr./Ms.……………………………………………….. bearing Regd.num: …………………….. Of **B.Tech** .... Semester …….. Branch …….. Section in the **20CS63 - SERVER - SIDE SCRIPTING LAB** during the Academic Year: 2022-2023.

No. of Experiments/Modules held : 10

No. of Experiments Done : 10

Date: …../…../23

INTERNAL EXAMINER **EXTERNAL EXAMINER**

INDEX

S.no	Experiments	Page no.	Date	sign
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				

Module 1

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

```
<!DOCTYPE html>
<html lang="en">
 <head>

  <!-- Declared Vars To Go Here -->

  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,

initial-scale=1">

  <!-- Metadata -->
  <meta name="description" content="">
  <meta name="author" content="">

  <link rel="icon" href="mysource_files/favicon.ico">

  <!-- Page Name and Site Name -->
  <title>CSE</title>

  <!-- CSS -->
  <link

href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/boo

tstrap.min.css" rel="stylesheet">
  <link href="mysource_files/style.css" rel="stylesheet">

 </head>

 <body>

  <div class="container">

   <header class="header clearfix" style="background-

color: #ffffff">

    <!-- Main Menu -->
    <nav>
     <ul class="nav nav-pills pull-right">
```

```
          <li class="active"><a

href="lbrce.ac.in">Home</a></li>
          <li><a href="#">About</a></li>
          <li><a href="#">Contact</a></li>
        </ul>
      </nav>

      <!-- Site Name -->
      <h1 class="h3 text-muted">COMPUTER SCIENCE AND

ENGINEERING</h1>

      <!-- Breadcrumbs -->
      <ol class="breadcrumb">
        <li><a href="#">Home</a></li>
        <li><a href="#">CSE</a></li>
        <li class="active">AIML</li>
      </ol>

    </header>

    <div class="page-heading">

      <!-- Page Heading -->
      <h1>DEPARTMENTS</h1>

    </div>

    <div class="row">

      <div class="col-sm-3">

        <!-- Sub Navigation -->
        <ul class="nav nav-pills nav-stacked">
          <li><a href="#">I CSE</a></li>
          <li class="active"><a href="#">II CSE</a>
            <ul>
              <li><a href="#">III CSE</a></li>
              <li><a href="#">IV CSE</a></li>

            </ul>
          </li>
          <li><a href="#">AIML</a></li>
        </ul>

      </div>
```

```html
    <div class="col-sm-6">

     <div class="page-contents">

      <!-- Design Body -->
      <h2> Heading</h2>
      <p>Donec id elit non mi porta gravida at eget

metus. Maecenas faucibus mollis interdum.</p>
      <h4>Sub Heading</h4>
      <p>Morbi leo risus, porta ac consectetur ac,

vestibulum at eros. Cras mattis consectetur purus sit amet

fermentum.</p>
      <h4>mini Heading</h4>
      <p>Maecenas sed diam eget risus varius blandit

sit amet non magna.</p>

     </div>

    </div>

    <div class="col-sm-3">

     <!-- Student Login Section -->
     <h2>Student Login</h2>

     <!-- Student Registration Section -->
     <h2>Student Registration</h2>

     <!-- Nested Right Column Content -->

    </div>

   </div>

   <footer class="footer">
    <p class="pull-right">
     <!-- Last Updated Design Area-->
     Last Updated: TUESDAY, MAY 2, 2023
    </p>
    <p>&copy; 2019 Company, Inc.</p>
   </footer>
```

```
  </div> <!-- /container -->

 </body>
</html>
```

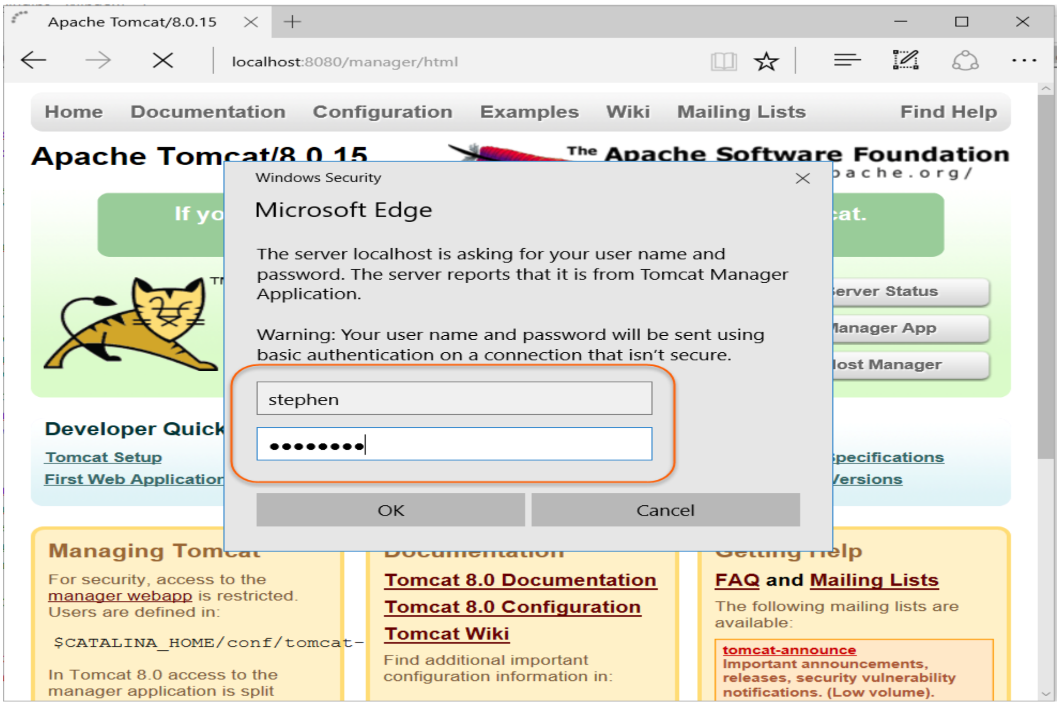
PROCEDURE

To deploy a web application to Apache Tomcat, you can copy a WAR file to the application base directory, e.g., c:/Tomcat8/webapps. This operation of course presupposes we know the application base directory. We could consult server.xml and look up the Host element to determine the directory name. A more straightforward approach, though, is to use the Tomcat manager web application.

To learn how to deploy a web application using the Apache Tomcat manager web application, follow these 12 steps:

1. Stop the Tomcat server.
2. You will need to edit CATALINA_BASE/conf/tomcat-users.xml, e.g., c:/Tomcat8/conf/tomcat-users.xml to add credentials to access the Tomcat manager web application.
3. Add the following XML statements immediately before the root end tag, i.e., </tomcat-users>:
4. Replace "your name" and "your password" with a user name and password of your choosing (no embedded spaces).
5. Save your changes.
6. You'll need to create a WAR file so that you will have web application to deploy. The instructions for creating a WAR file were contained in steps 1 and 2 in [How to deploy a WAR file in Apache Tomcat (Windows)](). Complete only steps 1 and 2. If you have already completed that project, delete the WAR file (DemoWebsite.war) and the unpacked folder (DemoWebsite) from CATALINA_HOME/webapps.
7. Start the Tomcat server.
8. In the address area of your browser, type http://localhost:8080/manager/html and submit that address to the browser.
9. You will be challenged for credentials. Provide the user name and password you

   specified in tomcat-users.xml (make sure to provide the correct spelling and case):
   Click **OK**



Windows Security

Microsoft Edge

The server localhost is asking for your user name and password. The server reports that it is from Tomcat Manager Application.

Warning: Your user name and password will be sent using basic authentication on a connection that isn't secure.

stephen

••••••••

OK Cancel

For security, access to the manager webapp is restricted. Users are defined in:

$CATALINA_HOME/conf/tomcat-

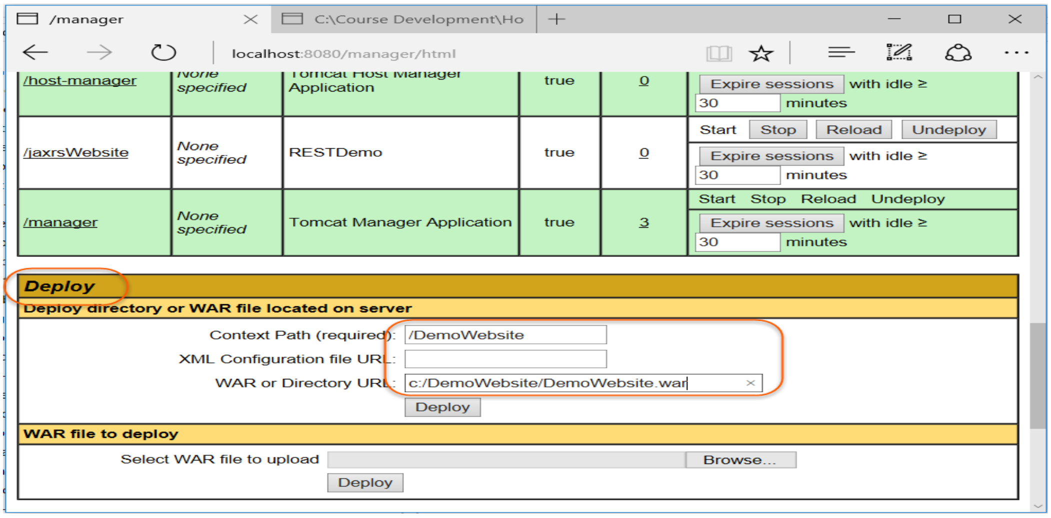
In Tomcat 8.0 access to the manager application is split

Find additional important configuration information in:

The following mailing lists are available:

tomcat-announce
Important announcements, releases, security vulnerability notifications. (Low volume).

9.The manager web page is now displayed. Scroll down until you see the Deploy section:



/host-manager	None specified	Tomcat Host Manager Application	true	0	Expire sessions with idle ≥ 30 minutes
/jaxrsWebsite	None specified	RESTDemo	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/manager	None specified	Tomcat Manager Application	true	3	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes

Deploy

Deploy directory or WAR file located on server

Context Path (required): /DemoWebsite

XML Configuration file URL:

WAR or Directory URL: c:/DemoWebsite/DemoWebsite.war

Deploy

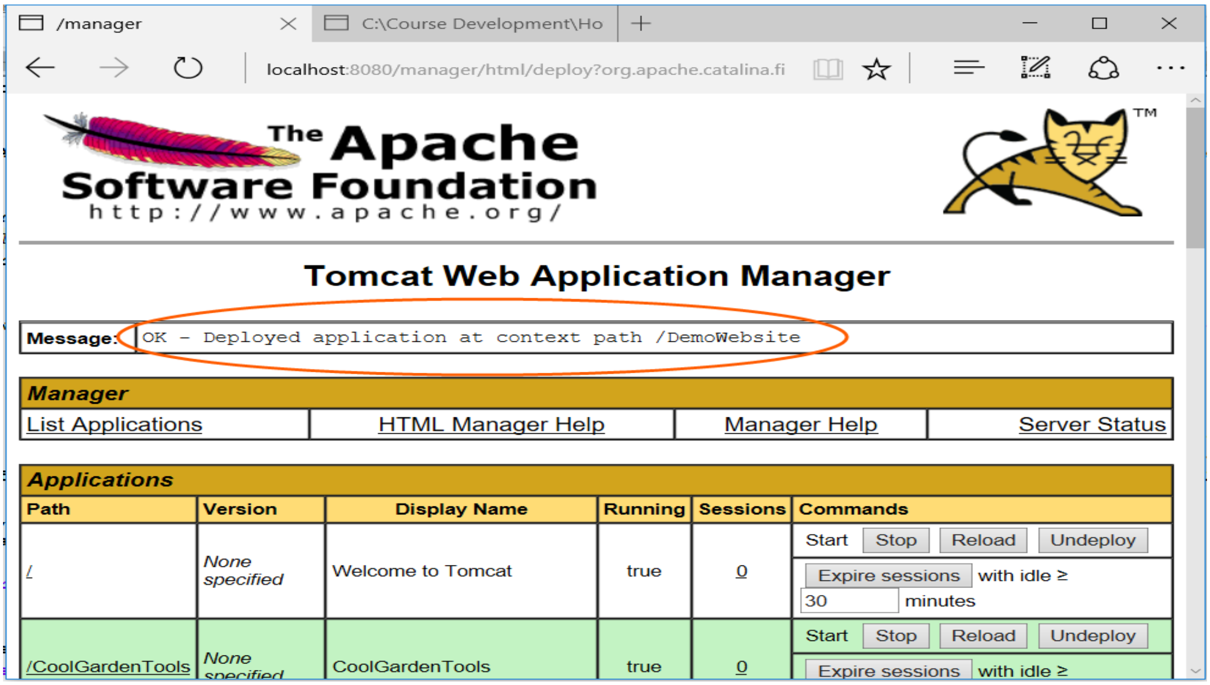
WAR file to deploy

Select WAR file to upload Browse...

Deploy

Note that I have already provided the Context Path (/DemoWebsite) and the WAR URL ((c:/DemoWebsite/DemoWebsite.war) in the text boxes. Fill in the text boxes as I've done. Then click Deploy.

10.The manager will indicate that the web site has been deployed.



/manager
C:\Course Development\Ho
localhost:8080/manager/html/deploy?org.apache.catalina.fi
The Apache Software Foundation
http://www.apache.org/
TM

Tomcat Web Application Manager

Message: OK - Deployed application at context path /DemoWebsite

Manager			
List Applications	HTML Manager Help	Manager Help	Server Status

Applications					
Path	Version	Display Name	Running	Sessions	Commands
/	*None specified*	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/CoolGardenTools	*None specified*	CoolGardenTools	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥

11.In a separate tab in your browser, type http://localhost:8080/DemoWebsite/DateJSP.jsp and submit that address to the browser.

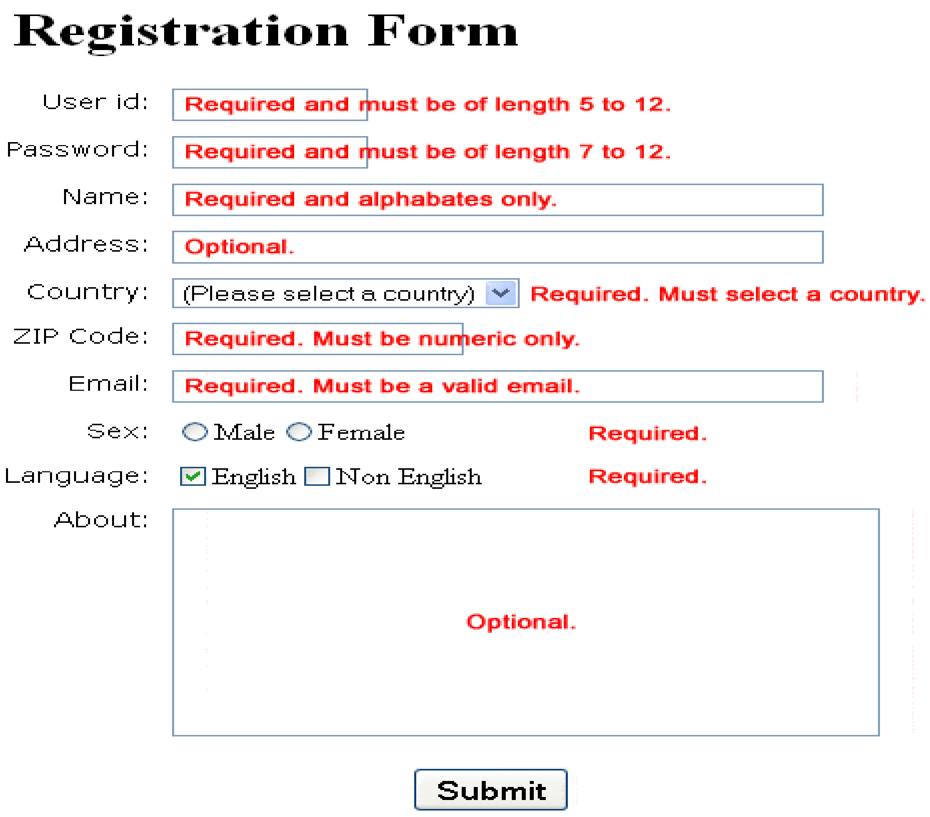
12.The web page will be displayed verifying successful deployment:

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

Form Validation

In this document, we have discussed JavaScript Form Validation using a sample registration form. The tutorial explores JavaScript validation on submit with detail explanation.

Following pictorial shows in which field, what validation we want to impose.



Registration Form

User id:	Required and must be of length 5 to 12.	
Password:	Required and must be of length 7 to 12.	
Name:	Required and alphabates only.	
Address:	Optional.	
Country:	(Please select a country)	Required. Must select a country.
ZIP Code:	Required. Must be numeric only.	
Email:	Required. Must be a valid email.	
Sex:	○ Male ○ Female	Required.
Language:	☑ English □ Non English	Required.
About:	Optional.	

How would we set those validations

We will create JavaScript functions (one for each input field whose value is to validate) which check whether a value submitted by user passes the validation.

All those functions are called from another function.

It sets the focus to the input field until the user supplies a valid value.

When the user does so, they may proceed and can supply value to the next available field.

The later JavaScript function created is called on the onsubmit event of the form.

HTML Code of the Sample Registration Form

```
<!DOCTYPE html>

<html lang="en"><head>

<meta charset="utf-8">

<title>JavaScript Form Validation using a sample registration form</title>

<meta name="keywords" content="example, JavaScript Form Validation, Sample
registration form" />

<meta name="description" content="This document is an example of
JavaScript Form Validation using a sample registration form. " />

<link rel='stylesheet' href='js-form-validation.css' type='text/css' />

<script src="sample-registration-form-validation.js"></script>

</head>

<body onload="document.registration.userid.focus();">

<h1>Registration Form</h1>

Use tab keys to move from one input field to the next.

<form name='registration' onSubmit="return formValidation();">

<ul>

<li><label for="userid">User id:</label></li>

<li><input type="text" name="userid" size="12" /></li>

<li><label for="passid">Password:</label></li>

<li><input type="password" name="passid" size="12" /></li>

<li><label for="username">Name:</label></li>

<li><input type="text" name="username" size="50" /></li>

<li><label for="address">Address:</label></li>

<li><input type="text" name="address" size="50" /></li>

<li><label for="country">Country:</label></li>

<li><select name="country">

<option selected="" value="Default">(Please select a country)</option>
```

```
<option value="AF">Australia</option>

<option value="AL">Canada</option>

<option value="DZ">India</option>

<option value="AS">Russia</option>

<option value="AD">USA</option>

</select></li>

<li><label for="zip">ZIP Code:</label></li>

<li><input type="text" name="zip" /></li>

<li><label for="email">Email:</label></li>

<li><input type="text" name="email" size="50" /></li>

<li><label id="gender">Sex:</label></li>

<li><input type="radio" name="msex" value="Male" /><span>Male</span></li>

<li><input type="radio" name="fsex" value="Female"
/><span>Female</span></li>

<li><label>Language:</label></li>

<li><input type="checkbox" name="en" value="en" checked
/><span>English</span></li>

<li><input type="checkbox" name="nonen" value="noen" /><span>Non
English</span></li>

<li><label for="desc">About:</label></li>

<li><textarea name="desc" id="desc"></textarea></li>

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

</ul>

</form>

</body>

</html>
```

CSS Code of the Sample Registration Form

```
h1 {

margin-left: 70px;
```

```css
}
form li {
list-style: none;
margin-bottom: 5px;
}

form ul li label{
float: left;
clear: left;
width: 100px;
text-align: right;
margin-right: 10px;
font-family:Verdana, Arial, Helvetica, sans-serif;
font-size:14px;
}

form ul li input, select, span {
float: left;
margin-bottom: 10px;
}

form textarea {
float: left;
width: 350px;
height: 150px;
}

[type="submit"] {
```

```
clear: left;

margin: 20px 0 0 230px;

font-size:18px

}


p {

margin-left: 70px;

font-weight: bold;

}
```

Copy

JavaScript code for validation

JavaScript function which is called on onSubmit

This function calls all other functions used for validation.

```
function formValidation()

{

var uid = document.registration.userid;

var passid = document.registration.passid;

var uname = document.registration.username;

var uadd = document.registration.address;

var ucountry = document.registration.country;

var uzip = document.registration.zip;

var uemail = document.registration.email;

var umsex = document.registration.msex;

var ufsex = document.registration.fsex; if(userid_validation(uid,5,12))

{

if(passid_validation(passid,7,12))

{
```

```
if(allLetter(uname))

{

if(alphanumeric(uadd))

{

if(countryselect(ucountry))

{

if(allnumeric(uzip))

{

if(ValidateEmail(uemail))

{

if(validsex(umsex,ufsex))

{

}

}

}

}

}

}

}

}

return false;

}
```

Copy

JavaScript function for validating userid

```
function userid_validation(uid,mx,my)

{

var uid_len = uid.value.length;
if (uid_len == 0 || uid_len >= my || uid_len < mx)

{

alert("User Id should not be empty / length be between "+mx+" to "+my);

uid.focus();

return false;

}

return true;

}
```

Module 2

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

XMLHttpRequest is an object that is used to send a request to the webserver for exchanging data or transferring and manipulating to it and from the server behind the scenes. You can use the received data to update the data present on the web page without even reloading the page.
Below is the complete syntax to use XMLHttpRequest object.

Syntax:
At first, you have to invoke the **XMLHttpRequest()** method as shown below.

```
var variable_name = new XMLHttpRequest();
```

After calling the **XMLHttpRequest()** method you have to define a callback function that will trigger after getting the response.

```
variable_name.onload = function () {

    // Content of callback function

    // after getting the response

}
```

Sending a request using the **open()** and **send()** methods as shown below.

```
variable_name.open("GET", "textFile.txt");

variable_name.send();
```

The XMLHttpRequest object has different functions and properties that are listed below.

XMLHttpRequest object methods:

- **new XMLHttpRequest():** It creates a new XMLHttpRequest object.
- **abort():** This method will cancel the current request to exchange data.
- **getAllResponseHeaders():** It will return a set of HTTP headers in form of a string.
- **getResponseHeader():** It will return the specified HTTP header information.
- **open(method, URL, async, userName, password):** This method specifies the method, URL, and other parameters of a request. In this function call, the method parameter defines the operation that has to be performed on the data like **GET, POST, HEAD,** and some other HTTP methods such as **PUT, DELETE.** The **async** parameter specifies the asynchronous behavior of the request. It holds two values *true* and *false***.** If the value is *true* then the script processing will continue after **send()** method without waiting for the response while *false* value means that the script will wait for a response before processing it.
- **send():** It will send the request to the server for data exchange. To GET the requests, **send()** is used.
- **send(string):** It also sends the request to the server for data exchange, but It is used to POST the requests.
- **setRequestHeader():** It will add the label and value pair to the header that has to be sent.

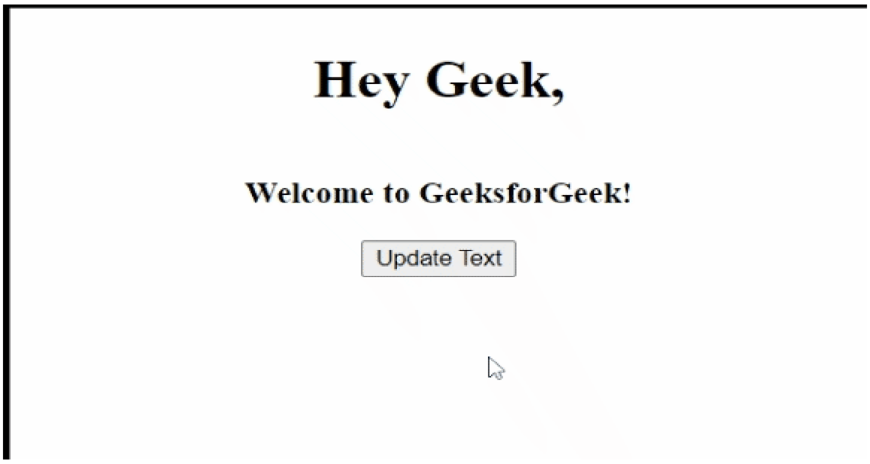
Example: Below example illustrates the use of **XMLHttpRequest()** object methods.

```
<!DOCTYPE html>
<html>
```

```
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible"
content="IE=edge">
    <meta name="viewport" content=
        "width=device-width, initial-scale=1.0">
    <title>Ready states of a request in ajax</title>

    <style>
        #container {
            display: flex;
            flex-direction: column;
            justify-content: center;
            align-items: center;
        }
    </style>
</head>

<body>
    <div id="container">
        <h1>Hey Geek,</h1>
        <h3>Welcome to GeeksforGeek!</h3>
        <button type="button" onclick="stateChange()">
            Update Text
        </button>
    </div>

    <script>
        function stateChange() {
            var state = new XMLHttpRequest();
            state.onload = function () {
                document.getElementById("container")
                    .innerHTML =
state.getAllResponseHeaders();
            }
            state.open("GET", "gfgInfo.txt", true);
```

```
                        state.send();
                    }
                </script>
            </body>

            </html>
```

Output:



Hey Geek,

Welcome to GeeksforGeek!

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

To read the XML file and print the details of an XML file in a Tabular form by using javaScript. We need to create an XML file which data we want to print. XML stands for Extensible Markup Language · It is a markup language much similar to HTML.The main purpose of the XML file for designed to store and transport the data. Creating an XML file is very simple because it uses custom tags.

Approach: After creating the XML file, we will write JavaScript to read and extract data from the file in tabular form. So, we will send the XMLHttpRequest to the server and fetch the details from the XML file by using JavaScript. If the request is finished then the response is ready and Status is “OK” so, we get the XML data by the use of Tag Name.

Now we will create two files:

1.employee.xml: A xml file that stores the employee’s details.To create an xml file we use custom tags, here we use different custom tags like the first name, last name, title, division, etc. which store the details of every employee according to the tag name.

```
<?xml version="1.0" encoding="utf-8"?>
<employees>
    <employee id="be129">
        <firstname>Jitendra</firstname>
        <lastname>Kumar</lastname>
        <title>Engineer</title>
        <division>Materials</division>
        <building>327</building>
        <room>19</room>
    </employee>
    <employee id="be130">
        <firstname>Amit</firstname>
        <lastname>Kumar</lastname>
        <title>Accountant</title>
        <division>Accts Payable</division>
        <building>326</building>
        <room>14</room>
    </employee>
    <employee id="be131">
        <firstname>Akash</firstname>
        <lastname>Kumar</lastname>
        <title>Engineering Manager</title>
        <division>Materials</division>
        <building>327</building>
        <room>21</room>
    </employee>
    <employee id="be132">
        <firstname>Aishwarya</firstname>
        <lastname>Kulshrestha</lastname>
        <title>Engineer</title>
        <division>Materials</division>
        <building>327</building>
        <room>22</room>
    </employee>
    <employee id="be133">
        <firstname>Sachin</firstname>
        <lastname>Kumar</lastname>
        <title>Engineer</title>
        <division>Materials</division>
        <building>327</building>
        <room>24</room>
    </employee>
    <employee id="be135">
        <firstname>Vikash</firstname>
        <lastname>kumar</lastname>
        <title>COO</title>
        <division>Management</division>
        <building>216</building>
        <room>26</room>
    </employee>
    <employee id="be136">
        <firstname>Suvam</firstname>
        <lastname>Basak</lastname>
        <title>Accountant</title>
        <division>Accts Payable</division>
        <building>326</building>
        <room>30</room>
    </employee>
    <employee id="be135">
        <firstname>Abhinav</firstname>
        <lastname>kumar</lastname>
        <title>COO</title>
        <division>Management</division>
        <building>216</building>
        <room>32</room>
    </employee>
    <employee id="be131">
        <firstname>DhanPal</firstname>
        <lastname>Singh</lastname>
        <title>Engineering Manager</title>
        <division>Materials</division>
        <building>327</building>
        <room>21</room>
    </employee>

</employees>
```

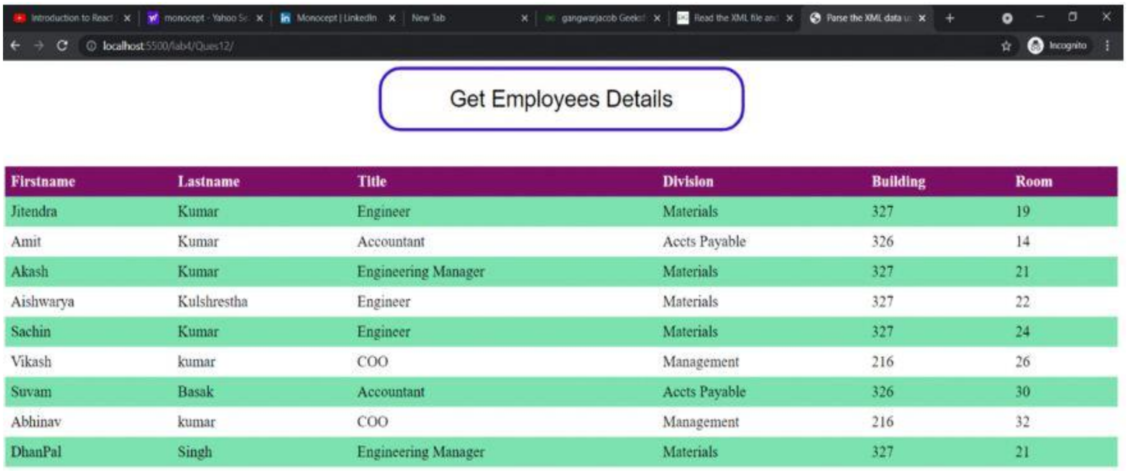
2.index.html: This file contains the HTML, CSS and JavaScript code. We use the style tag for the CSS part in which we are styling the table attribute and button after that we use the script tag in which we write the JavaScript code and insert the employee.xml file. In the loadXMLDoc( ) function we send the HTTP request to the server when the request is finished then we get the response from the server and access the data from an XML file. In empDetails( ) function if we get the response from the server then we fetch the XML file data one by one by using the custom tag name. To display the data of this xml file we simply click on the Get Employees details button and the xml file data will be displayed on your screen in tabular form.

```
<!DOCTYPE html>

<head>
	<title>Reads the XML data using JavaScript</title>

	<!-- CSS -->
	<style>
		table {
			border-collapse: collapse;
			width: 100%;
		}

		th,
		td {
			text-align: left;
			padding: 8px;
		}

		tr:nth-child(even) {
			background-color: #7ce2af
		}

		th {
			background-color: #7c0f65;
			color: white;
		}

		.button {
			position: relative;
			text-align: center;
			padding: 20px;
```

```
            border: 4px solid rgb(55, 12, 211);
            background: rgba(20, 192, 4, 0.5);
            color: rgb(230, 36, 78);
            outline: none;
            border-radius: 30px;
            font-size: 30px;
            width: 500px;

        }

        .button:hover {
            color: black;
            background: white;
        }
    </style>

    <!--JavaScript-->
    <script>
        function loadXMLDoc() {
            var xmlhttp = new XMLHttpRequest();
            xmlhttp.onreadystatechange = function () {

                // Request finished and response
                // is ready and Status is "OK"
                if (this.readyState == 4 && this.status ==
200) {
                    empDetails(this);
                }
            };

            // employee.xml is the external xml file
            xmlhttp.open("GET", "employee.xml", true);
            xmlhttp.send();
        }

        function empDetails(xml) {
            var i;
            var xmlDoc = xml.responseXML;
            var table =

    `<tr><th>Firstname</th><th>Lastname</th>
                    <th>Title</th><th>Division</th>
                    <th>Building</th><th>Room</th>
                </tr>`;
            var x =
xmlDoc.getElementsByTagName("employee");

            // Start to fetch the data by using TagName
            for (i = 0; i < x.length; i++) {
                table += "<tr><td>" +

    x[i].getElementsByTagName("firstname")[0]
                    .childNodes[0].nodeValue +
"</td><td>" +

    x[i].getElementsByTagName("lastname")[0]
                    .childNodes[0].nodeValue +
"</td><td>" +

    x[i].getElementsByTagName("title")[0]
                    .childNodes[0].nodeValue +
"</td><td>" +

    x[i].getElementsByTagName("division")[0]
                    .childNodes[0].nodeValue +
"</td><td>" +

    x[i].getElementsByTagName("building")[0]
                    .childNodes[0].nodeValue +
"</td><td>" +

    x[i].getElementsByTagName("room")[0]
                    .childNodes[0].nodeValue +
"</td></tr>";
```

```
            }

            // Print the xml data in table form
            document.getElementById("id").innerHTML = table;
        }
    </script>
</head>

<body>
    <center>
        <button type="button" class="button"
            onclick="loadXMLDoc()">
            Get Employees Details
        </button>
    </center>

    <br><br>
    <table id="id"></table>
</body>

</html>
```

Steps to run the application: To read the XML data we need to run this code on the local server So, first we start the local server and after it opens the chrome browser, start the local host and see the results. After clicking on the getting Employees Details button, we will get the Employees Details in a tabular form.



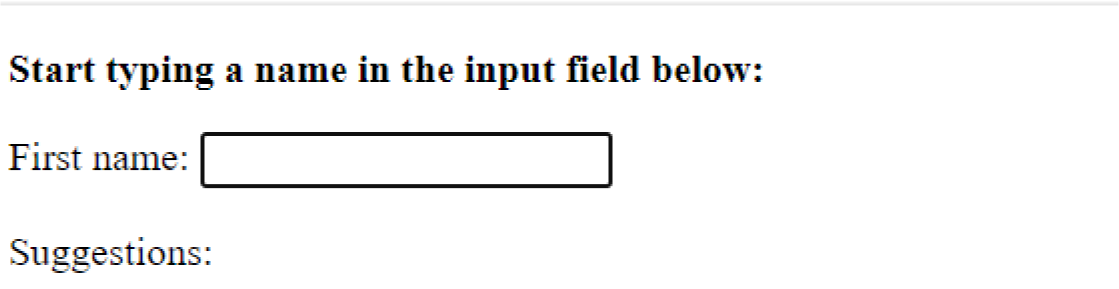
Firstname	Lastname	Title	Division	Building	Room
Jitendra	Kumar	Engineer	Materials	327	19
Amit	Kumar	Accountant	Accts Payable	326	14
Akash	Kumar	Engineering Manager	Materials	327	21
Aishwarya	Kulshrestha	Engineer	Materials	327	22
Sachin	Kumar	Engineer	Materials	327	24
Vikash	kumar	COO	Management	216	26
Suvam	Basak	Accountant	Accts Payable	326	30
Abhinav	kumar	COO	Management	216	32
DhanPal	Singh	Engineering Manager	Materials	327	21

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:

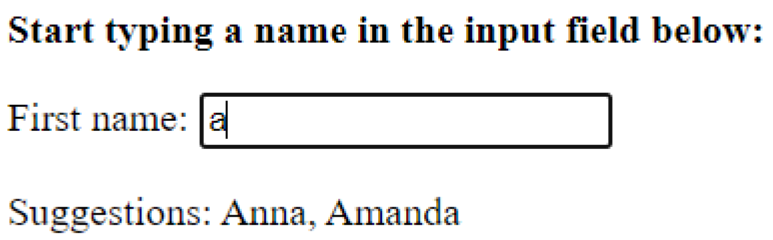
- Create an XMLHttpRequest object
- Create the function to be executed when the server response is ready
- Send the request off to a PHP file (gethint.php) on the server
- Notice that the q parameter is added to the url (gethint.php?q="+str)
- And the str variable holds the content of the input field

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:



Start typing a name in the input field below:

First name: a

Suggestions: Anna, Amanda

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

we are interacting with the database. You don't have to make any extra effort. Only write the database logic in you server side page.

In this example, we have written the server side code inside the index.jsp file.

Steps to create ajax example with database through jsp

You need to follow following steps:

1. load the org.json.jar file
2. create input page to receive any text or number
3. create server side page to process the request

Load the org.json.jar file

download this example, we have included the org.json.jar file inside the WEB-INF/lib directory.

create input page to receive any text or number

In this page, we have created a form that gets input from the user. When user press any key **sendInfo()** function is called. We have written all the ajax code inside this function.

We have called the **getInfo()** function whenever ready state changes. It writes the returned data in the web page dynamically by the help of **innerHTML** property.

table1.html

```
<html>
<head>
<script>
var request;
function sendInfo()
{
var v=document.vinform.t1.value;
var url="index.jsp?val="+v;

if(window.XMLHttpRequest){
request=new XMLHttpRequest();
}
else if(window.ActiveXObject){
request=new ActiveXObject("Microsoft.XMLHTTP");
}

try{
request.onreadystatechange=getInfo;
request.open("GET",url,true);
request.send();
}catch(e){alert("Unable to connect to server");}
}

function getInfo(){
if(request.readyState==4){
var val=request.responseText;
document.getElementById('amit').innerHTML=val;
}
}

</script>
</head>
<body>
   <marquee><h1>This is an example of ajax</h1></marquee>
<form name="vinform">
Enter id:<input type="text" name="t1" onkeyup="sendInfo()">
</form>

<span id="amit"> </span>

</body>
</html>
```

---

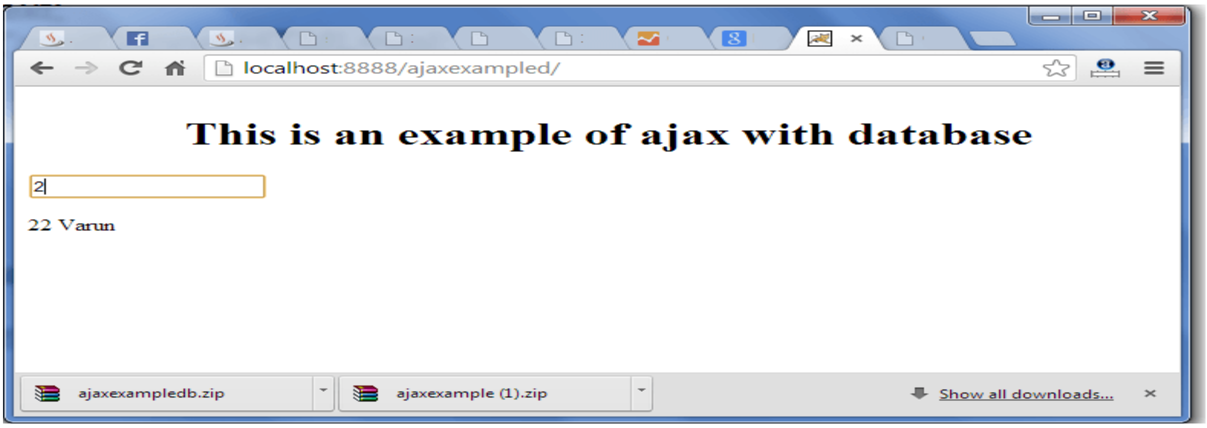
create server side page to process the request

In this jsp page, we printing the id and name of the employee for the given id.

index.jsp

```
<%@ page import="java.sql.*"%>

<%
String s=request.getParameter("val");
if(s==null || s.trim().equals("")){
out.print("Please enter id");
}else{
int id=Integer.parseInt(s);
out.print(id);
try{
Class.forName("com.mysql.jdbc.Driver");
Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/mdb","root","root");
PreparedStatement ps=con.prepareStatement("select * from emp where id=?");
ps.setInt(1,id);
ResultSet rs=ps.executeQuery();
while(rs.next()){
out.print(rs.getInt(1)+" "+rs.getString(2));
}
con.close();
}catch(Exception e){e.printStackTrace();}
}
%>
```

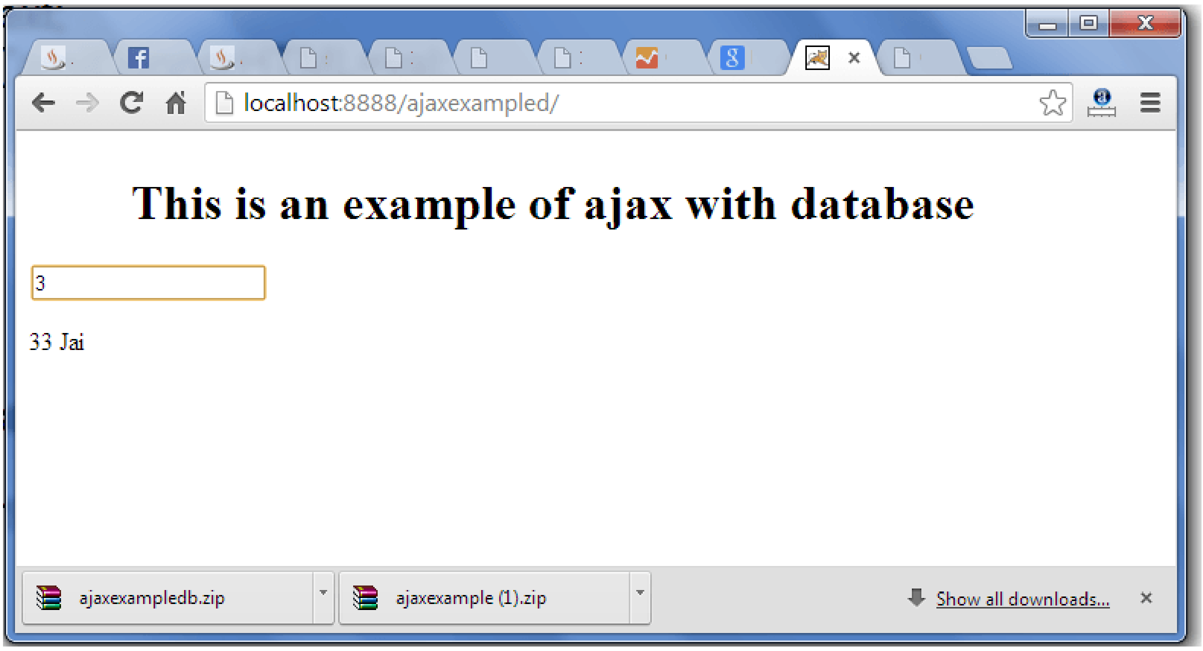
Output



This is an example of ajax with database

2

22 Varun



This is an example of ajax with database

3

33 Jai

Module 4

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

PHP provides two methods through which a client (browser) can send information to the server. These methods are given below, and discussed in detail:

1. GET method
2. POST method

Get and Post methods are the HTTP request methods used inside the *<form>* tag to send form data to the server.

HTTP protocol enables the communication between the client and the server where a browser can be the client, and an application running on a computer system that hosts your website can be the server.

GET method

The **GET** method is used to submit the HTML form data. This data is collected by the predefined **$_GET variable** for processing.

The information sent from an HTML form using the GET method is visible to everyone in the browser's address bar, which means that all the variable names and their values will be displayed in the URL. Therefore, the get method is not secured to send sensitive information.

For Example

```
localhost/gettest.php?username=Harry&bloodgroup=AB+
```

The **bold** part in the above URL is the variables name and *italic* part contains the values for their corresponding variable.

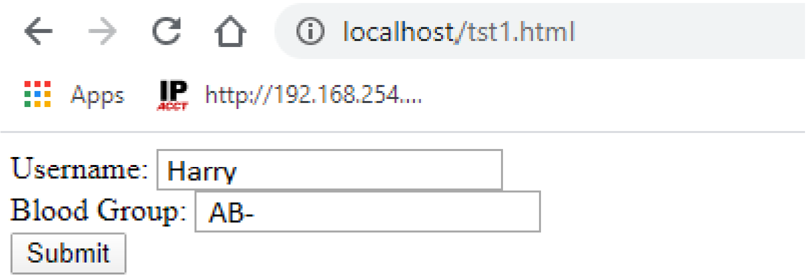
Example

The below code will display an HTML form containing two input fields and a submit button. In this HTML form, we used the method = "get" to submit the form data.

file: test1.html

```
<html>
  <body>

    <form action = "gettest.php" method = "GET">
      Username: <input type = "text" name = "username" /> <br>
```

```
        Blood Group: <input type = "text" name = "bloodgroup" /> <br>
        <input type = "submit" />
      </form>

   </body>
</html>
```



Username: Harry

Blood Group: AB-

Submit

Create gettest.php file, which will accept the data sent by HTML form.

file: gettest.php

```
<html>
   <body>

     Welcome <?php echo $_GET["username"]; ?> </br>
     Your blood group is: <?php echo $_GET["bloodgroup"]; ?>

   </body>
</html>
```

When the user will click on **Submit** button after filling the form, the URL sent to the server could look something like this:

localhost/gettest.php?username=Harry&bloodgroup=AB-

The output will look like the below output:

```
Welcome Harry
Your blood group is: AB-
```

Advantages of GET method (method = "get")

- You can bookmark the page with the specific query string because the data sent by the GET method is displayed in URL.
- GET requests can be cached.

- GET requests are always remained in the browser history.

Disadvantages of GET Method

- The GET method should not be used while sending any sensitive information.
- A limited amount of data can be sent using method = "get". This limit should not exceed 2048 characters.
- For security reasons, never use the GET method to send highly sensitive information like username and password, because it shows them in the URL.
- The GET method cannot be used to send binary data (such as images or word documents) to the server.

POST method

Similar to the GET method, the **POST** method is also used to submit the HTML form data. But the data submitted by this method is collected by the predefined superglobal variable **$_POST** instead of $_GET.

Unlike the GET method, it does not have a limit on the amount of information to be sent. The information sent from an HTML form using the POST method is not visible to anyone.

For Example

```
localhost/posttest.php
```

With the help of an example, let's understand how the POST method works-

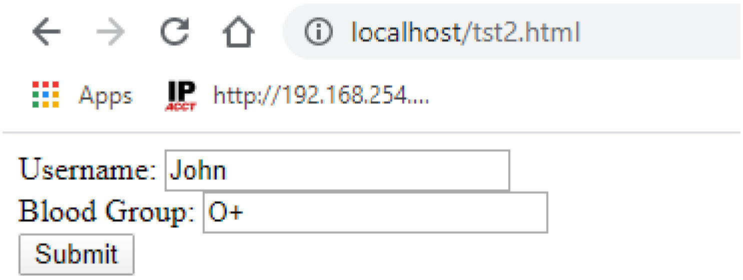
Example

The below code will display an HTML form containing two input fields and a submit button. In this HTML form, we used the method = "post" to submit the form data.

file: test2.html

```
<html>
  <body>

    <form action = "posttest.php" method = "post">
      Username: <input type = "text" name = "username" /> <br>
      Blood Group: <input type = "text" name = "bloodgroup" /> <br>
      <input type = "submit" />
    </form>

  </body>
```

```
</html>
```



Username: John
Blood Group: O+

Now create **posttest.php** file to accept the data sent by HTML form.

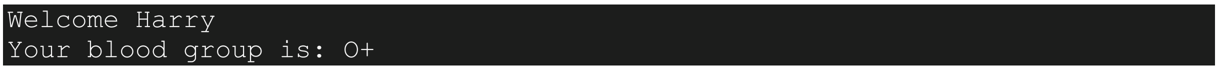
file: posttest.php

```
<html>
  <body>

    Welcome <?php echo $_POST["username"]; ?> </br>
    Your blood group is: <?php echo $_POST["bloodgroup"]; ?>

  </body>
</html>
```

When the user will click on **Submit** button after filling the form, the URL sent to the server could look something like this:

localhost/posttest.php

The output will look like the below output:



```
Welcome Harry
Your blood group is: O+
```

Advantages of POST method (method = "post")

- The POST method is useful for sending any sensitive information because the information sent using the POST method is not visible to anyone.
- There is no limitation on size of data to be sent using the POST Method. You can send a large amount of information using this method.
- Binary and ASCII data can also be sent using the POST method.
- Data security depends on the HTTP protocol because the information sent using the POST method goes through the HTTP header. By using secure HTTP, you can ensure that your data is safe.

Disadvantages of POST Method

- POST requests do not cache.
- POST requests never remain in the browser history.
- It is not possible to bookmark the page because the variables are not displayed in URL.

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. You may use a relative or absolute path of the file.

Syntax

There are two syntaxes available for include:

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

Examples

Let's see a simple PHP include example.

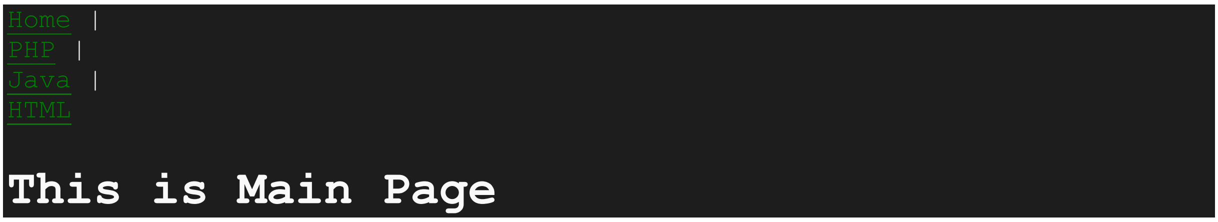
File: menu.html

```
<a href="http://www.javatpoint.com">Home</a> |
<a href="http://www.javatpoint.com/php-tutorial">PHP</a> |
<a href="http://www.javatpoint.com/java-tutorial">Java</a> |
<a href="http://www.javatpoint.com/html-tutorial">HTML</a>
```

File: include1.php

```
<?php include("menu.html"); ?>
<h1>This is Main Page</h1>
```

Output:



```
Home |
PHP |
Java |
HTML

This is Main Page
```

PHP require

PHP require is similar to include, which is also used to include files. The only difference is that it stops the execution of script if the file is not found whereas include doesn't.

Syntax

There are two syntaxes available for require:

```
require 'filename';
Or
require ('filename');
```

Examples

Let's see a simple PHP require example.

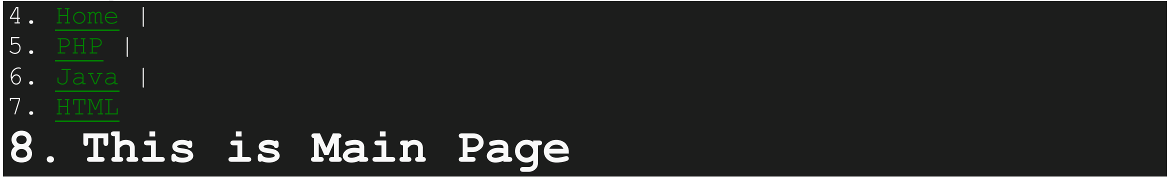
File: menu.html

```
<a href="http://www.javatpoint.com">Home</a> |
<a href="http://www.javatpoint.com/php-tutorial">PHP</a> |
<a href="http://www.javatpoint.com/java-tutorial">Java</a> |
<a href="http://www.javatpoint.com/html-tutorial">HTML</a>
```

File: require1.php

```
<?php require("menu.html"); ?>
<h1>This is Main Page</h1>
```

Output:



```
Home |
PHP |
Java |
HTML
```

This is Main Page

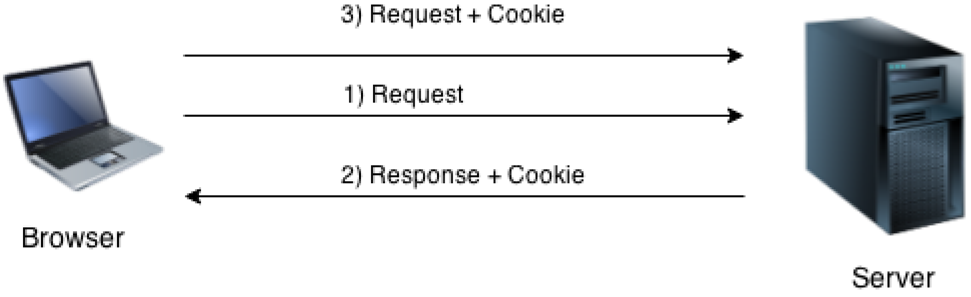
MODULE 5

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

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.



PHP setcookie() function

PHP setcookie() function is used to set cookie with HTTP response. Once cookie is set, you can access it by $_COOKIE superglobal variable.

Syntax

bool setcookie (string $name [, string $value [, int $expire = 0 [, string $path [, string $domain [, bool $secure = false [, bool $httponly = false]]]]]])

OR

Syntax:

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

Parameters: The setcookie() function requires six arguments in general which are:

- **Name:** It is used to set the name of the cookie.
- **Value:** It is used to set the value of the cookie.

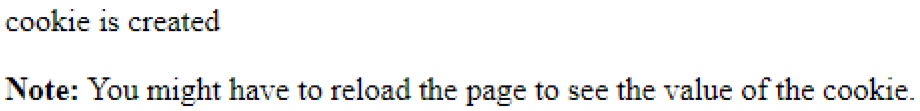
- **Expire:** It is used to set the expiry timestamp of the cookie after which the cookie can’t be accessed.
- **Path:** It is used to specify the path on the server for which the cookie will be available.
- **Domain:** It is used to specify the domain for which the cookie is available.
- **Security:** It is used to indicate that the cookie should be sent only if a secure HTTPS connection exists.

EXAMPLE PROGRAM

Cookie.php

```
<!DOCTYPE html>
<?php
    setcookie("Auction_Item", "Luxury Car", time() + 2 * 24 * 60 *
60);
?>
<html>
<body>
    <?php
        echo "cookie is created."
    ?>
    <p>
        <strong>Note:</strong>
        You might have to reload the
        page to see the value of the cookie.
    </p>

</body>
</html>
```

Output:



```
cookie is created

Note: You might have to reload the page to see the value of the cookie.
```

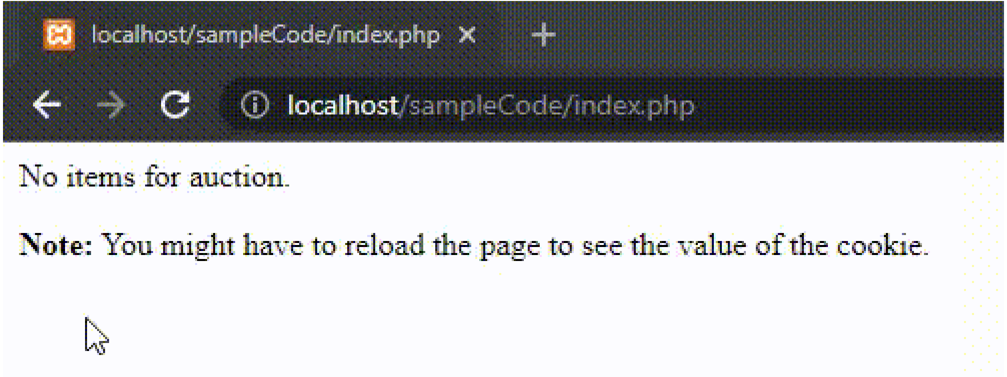
Checking Whether a Cookie Is Set Or Not: It is always advisable to check whether a cookie is set or not before accessing its value. Therefore to check whether a cookie is set or not, the PHP isset() function is used. To check whether a cookie “Auction_Item” is set or not, the isset() function is executed as follows:

Example: This example describes checking whether the cookie is set or not.

EXAMPLE PROGRAM 2

```
<!DOCTYPE html>
<?php
    setcookie("Auction_Item", "Luxury Car", time() + 2 * 24 *
60 * 60);
?>
<html>
<body>
    <?php
    if (isset($_COOKIE["Auction_Item"]))
    {
        echo "Auction Item is a  " .
$_COOKIE["Auction_Item"];
    }
    else
    {
        echo "No items for auction.";
    }
    ?>
    <p>
        <strong>Note:</strong>
        You might have to reload the page
        to see the value of the cookie.
    </p>

</body>
</html>
```

Output:


localhost/sampleCode/index.php
localhost/sampleCode/index.php

No items for auction.

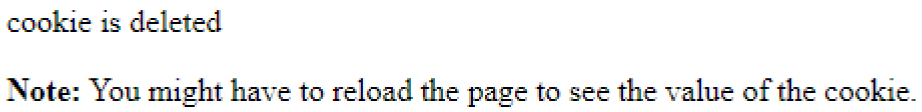
Note: You might have to reload the page to see the value of the cookie.

Deleting Cookies: The setcookie() function can be used to delete a cookie. For deleting a cookie, the setcookie() function is called by passing the cookie name and other arguments or empty strings but however this time, the expiration date is required to be set in the past. To delete a cookie named “Auction_Item”, the following code can be executed.

Example: This example describes the deletion of the cookie value.

```
<!DOCTYPE html>
<?php
    setcookie("Auction_Item", "Luxury Car", time() + 2 * 24 *
60 * 60);
?>
<html>
<body>
    <?php
       setcookie("Auction_Item", "", time() - 60);
    ?>
    <?php
        echo "cookie is deleted"
    ?>
    <p>
        <strong>Note:</strong>
        You might have to reload the page
        to see the value of the cookie.
    </p>

</body>
</html>
```

Output:



```
cookie is deleted

Note: You might have to reload the page to see the value of the cookie.
```

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

PHP Session

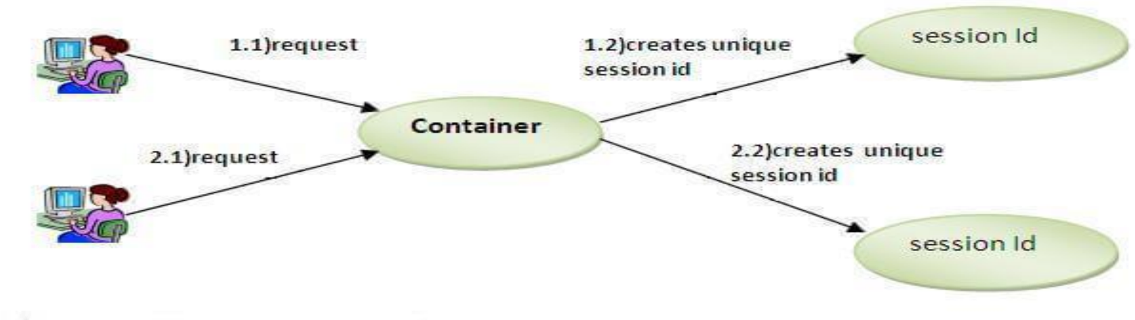
PHP session is used to store and pass information from one page to another temporarily (until user close the website).

PHP session technique is widely used in shopping websites where we need to store and pass cart information e.g. username, product code, product name, product price etc from one page to another.

PHP session creates unique user id for each browser to recognize the user and avoid conflict between multiple browsers.

or

A session means the duration spent by a Web user from the time logged in to the time logged out—during this time the user can view protected content. Protected content means the information that is not open to everyone (like your e-mail inbox). The beauty of a session is that it keeps the login credentials of users until they log out, even if they move from one Web page to another, in the same Web service, of course.



PHP session_start() function

PHP session_start() function is used to start the session. It starts a new or resumes existing session. It returns existing session if session is created already. If session is not available, it creates and returns new session.

Syntax

bool session_start (void)

Example

session_start();

PHP $_SESSION

PHP $_SESSION is an associative array that contains all session variables. It is used to set and get session variable values.

Example: Store information

```
$_SESSION["user"] = "Sachin";
```

Example: Get information

```
echo $_SESSION["user"];
```

PHP Session Example

session1.php

```
<?php
session_start();
?>
<html>
<body>
<?php
$_SESSION["user"] = "Sachin";
echo "Session information are set successfully.<br/>";
?>
<a href="session2.php">Visit next page</a>
</body>
</html>
```

session2.php

```
<?php
session_start();
?>
<html>
<body>
<?php
echo "User is: ".$_SESSION["user"];
?>
</body>
</html>
```

PHP Session Counter Example

sessioncounter.php

```php
<?php
  session_start();

  if (!isset($_SESSION['counter'])) {
    $_SESSION['counter'] = 1;
  } else {
    $_SESSION['counter']++;
  }
  echo ("Page Views: ".$_SESSION['counter']);
?>
```

PHP Destroying Session

PHP session_destroy() function is used to destroy all session variables completely.

session3.php

```php
<?php
session_start();
session_destroy();
?>
```

Module 6

a) Develop a PHP program to perform various file handling operations.

File handling is needed for any application. For some tasks to be done file needs to be processed. File handling in PHP is similar as file handling is done by using any programming language like C. PHP has many functions to work with normal files. Those functions are:

fopen() – PHP fopen() function is used to open a file. First parameter of fopen() contains name of the file which is to be opened and second parameter tells about mode in which file needs to be opened, e.g.,

Syntax

```
resource fopen ( string $filename , string $mode [, bool $use_include_path = false [, resource $context
```

Example

```php
<?php
$handle = fopen("c:\\folder\\file.txt", "r");
?>
```

Files can be opened in any of the following modes :

- **"w" –** Opens a file for write only. If file not exist then new file is created and if file already exists then contents of file is erased.
- **"r" –** File is opened for read only.
- **"a" –** File is opened for write only. File pointer points to end of file. Existing data in file is preserved.
- **"w+" –** Opens file for read and write. If file not exist then new file is created and if file already exists then contents of file is erased.
- **"r+" –** File is opened for read/write.
- **"a+" –** File is opened for write/read. File pointer points to end of file. Existing data in file is preserved. If file is not there then new file is created.
- **"x" –** New file is created for write only.

PHP Read File

PHP provides various functions to read data from file. There are different functions that allow you to read all file data, read data line by line and read data character by character.

The available PHP file read functions are given below.

- fread()
- fgets()
- fgetc()

2) **fread() —** After file is opened using fopen() the contents of data are read using fread(). It takes two arguments. One is file pointer and another is file size in bytes, e.g.,

Syntax

string fread (resource $handle , int $length)

Example

```php
<?php
$filename = "c:\\myfile.txt";
$handle = fopen($filename, "r");//open file in read mode

$contents = fread($handle, filesize($filename));//read file

echo $contents;//printing data of file
fclose($handle);//close file
?>
```

Output

```
this is first line
this is another line
this is third line
```

PHP Read File - fgets()

The PHP fgets() function is used to read single line from the file.

Syntax

string fgets (resource $handle [, int $length])

Example

```
<?php
$fp = fopen("c:\\file1.txt", "r");//open file in read mode
echo fgets($fp);
fclose($fp);
?>
```

Output

```
this is first line
```

PHP Read File - fgetc()

The PHP fgetc() function is used to read single character from the file. To get all data using fgetc() function, use !feof() function inside the while loop.

Syntax

string fgetc (resource $handle)

Example

```
<?php
$fp = fopen("c:\\file1.txt", "r");//open file in read mode
while(!feof($fp)) {
  echo fgetc($fp);
}
fclose($fp);
?>
```

Output

```
this is first line this is another line this is third line
```

PHP Write File

PHP fwrite() and fputs() functions are used to write data into file. To write data into 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

int fwrite (resource $handle , string $string [, int $length])

Example

```
<?php
$fp = fopen('data.txt', 'w');//opens file in write-only mode
fwrite($fp, 'welcome ');
fwrite($fp, 'to php file write');
fclose($fp);

echo "File written successfully";
?>
```

Output: data.txt

```
welcome to php file write
```

PHP Overwriting File

If you run the above code again, it will erase the previous data of the file and writes the new data. Let's see the code that writes only new data into data.txt file.

```
<?php
$fp = fopen('data.txt', 'w');//opens file in write-only mode
fwrite($fp, 'hello');
fclose($fp);
```

```

echo "File written successfully";
?>
```

Output: data.txt

```
hello
```

PHP Append to File

You can append data into file by using a or a+ mode in fopen() function. Let's see a simple example that appends data into data.txt file.

Let's see the data of file first.

data.txt

```
welcome to php file write
```

PHP Append to File - fwrite()

The PHP fwrite() function is used to write and append data into file.

Example

```
<?php
$fp = fopen('data.txt', 'a');//opens file in append mode
fwrite($fp, ' this is additional text ');
fwrite($fp, 'appending data');
fclose($fp);

echo "File appended successfully";
?>
```

Output: data.txt

```
welcome to php file write this is additional text appending data
```

PHP Delete File

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 ] )
```

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

PHP Delete File Example

```php
<?php
$status=unlink('data.txt');
if($status){
echo "File deleted successfully";
}else{
echo "Sorry!";
}
?>
```

Output

```
File deleted successfully
```

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

PHP File Upload

PHP allows you to upload single and multiple files through few lines of code only.

PHP file upload features allows you to upload binary and text files both. Moreover, you can have the full control over the file to be uploaded through PHP authentication and file operation functions.

PHP $_FILES

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

Here, we are assuming that file name is *filename*.

$_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

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 )
```

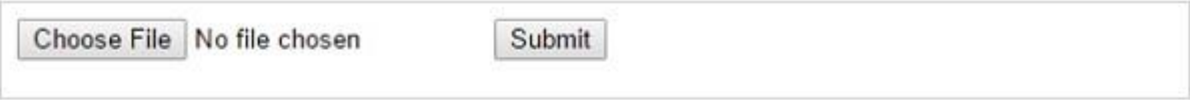
PHP File Upload Example

uploadform.html

```
 <html>
<title> FILE UPLOAD</title>
<body>
<form action="upload.php" method="post" enctype="multipart/form-data">
  Select File:
 <input type="file" name="fileToUpload"/>
 <input type="submit" value="Upload Image" name="submit"/>
</form>
</body>
</html>
```

Upload.php

```
<?php
$target_path = "e:/";
$target_path = $target_path.basename( $_FILES['fileToUpload']['name']);

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



PHP Download File

PHP enables you to download file easily using 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.

PHP Download File Example: Text File

download1.php

```
<?php
$file_url = 'http://www.javatpoint.com/f.txt';
header('Content-Type: application/octet-stream');
header("Content-Transfer-Encoding: utf-8");
header("Content-disposition: attachment; filename=\"" . basename($file_url) . "\"");
readfile($file_url);
?>
```

PHP Download File Example: Binary File

download2.php

```
<?php
$file_url = 'http://www.myremoteserver.com/file.exe';
header('Content-Type: application/octet-stream');
header("Content-Transfer-Encoding: Binary");
header("Content-disposition: attachment; filename=\"" . basename($file_url) . "\"");
readfile($file_url);  ?>
```

Module 7

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

PHP Mail

PHP mail() function is used to send email in PHP. You can send text message, html message and attachment with message using PHP mail() function.

PHP mail() function

Syntax

```
bool mail ( string $to , string $subject , string $message [, string $additional_headers [, string $additional_parameters ]] )
```

$to: specifies receiver or receivers of the mail. The receiver must be specified one of the following forms.

- user@example.com
- user@example.com, anotheruser@example.com
- User <user@example.com>
- User <user@example.com>, Another User <anotheruser@example.com>

$subject: represents subject of the mail.

$message: represents message of the mail to be sent.

$additional_headers (optional): specifies the additional headers such as From, CC, BCC etc. Extra additional headers should also be separated with CRLF (\r\n).

PHP Mail Example

File: mailer.php

```
<?php
   ini_set("sendmail_from", "sonoojaiswal@javatpoint.com");
   $to = "sonoojaiswal1987@gmail.com";//change receiver address
  $subject = "This is subject";
   $message = "This is simple text message.";
   $header = "From:sonoojaiswal@javatpoint.com \r\n";

   $result = mail ($to,$subject,$message,$header);

   if( $result == true ){
      echo "Message sent successfully...";
   }else{
     echo "Sorry, unable to send mail...";
   }
?>
```

If you run this code on the live server, it will send an email to the specified receiver.

PHP Mail: Send HTML Message

To send HTML message, you need to mention Content-type **text/html** in the message header.

```
<?php
  $to = "abc@example.com";//change receiver address
  $subject = "This is subject";
  $message = "<h1>This is HTML heading</h1>";
    $header = "From:xyz@example.com \r\n";
  $header .= "MIME-Version: 1.0 \r\n";
  $header .= "Content-type: text/html;charset=UTF-8 \r\n";
   $result = mail ($to,$subject,$message,$header);

 if( $result == true ){
     echo "Message sent successfully...";
   }else{
      echo "Sorry, unable to send mail...";
   }
?>
```

PHP Mail: Send Mail with Attachment

To send message with attachment, you need to mention many header information which is used in the example given below.

```
<?php
  $to = "abc@example.com";
  $subject = "This is subject";
  $message = "This is a text message.";
  # Open a file
  $file = fopen("/tmp/test.txt", "r" );//change your file location
  if( $file == false )
  {
     echo "Error in opening file";
     exit();
  }
  # Read the file into a variable
  $size = filesize("/tmp/test.txt");
  $content = fread( $file, $size);

  # encode the data for safe transit
  # and insert \r\n after every 76 chars.
  $encoded_content = chunk_split( base64_encode($content));

  # Get a random 32 bit number using time() as seed.
  $num = md5( time() );

  # Define the main headers.
  $header = "From:xyz@example.com\r\n";
  $header .= "MIME-Version: 1.0\r\n";
  $header .= "Content-Type: multipart/mixed; ";
  $header .= "boundary=$num\r\n";
  $header .= "--$num\r\n";

  # Define the message section
  $header .= "Content-Type: text/plain\r\n";
  $header .= "Content-Transfer-Encoding:8bit\r\n\n";
```

```
 $header .= "$message\r\n";
  $header .= "--$num\r\n";

  # Define the attachment section
  $header .= "Content-Type:  multipart/mixed; ";
  $header .= "name=\"test.txt\"\r\n";
  $header .= "Content-Transfer-Encoding:base64\r\n";
  $header .= "Content-Disposition:attachment; ";
  $header .= "filename=\"test.txt\"\r\n\n";
  $header .= "$encoded_content\r\n";
  $header .= "--$num--";

  # Send email now
  $result = mail ( $to, $subject, "", $header );
  if( $result == true ){
     echo "Message sent successfully...";
   }else{
      echo "Sorry, unable to send mail...";
   }
?>
```

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.

ANSWER:MODULE 3(A)

Module 8

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

PHP to connect an HTML form to a MySQL database. We'll use XAMPP as the server software to create a database and run PHP.

We'll use the below steps to create a connection:

1. Set up XAMPP and configure a PHP development environment
2. Create an HTML form
3. Create a MySQL database
4. Create a PHP file
5. Create a connection

Step 1: Set up XAMPP

The method to configure a PHP development environment with XAMPP

Step 2: Create an HTML form and PHP

This Answer explains what an HTML form is and how to create it.

```
<!DOCTYPE
html>
<html>
  <head>
    <title>Registration Page</title>
    <link rel="stylesheet" type="text/css" href="css/bootstrap.css" />
  </head>
  <body>
    <div class="container">
      <div class="row col-md-6 col-md-offset-3">
        <div class="panel panel-primary">
          <div class="panel-heading text-center">
            <h1>Registration Form</h1>
          </div>
          <div class="panel-body">
            <form action="connect.php" method="post">
              <div class="form-group">
                <label for="firstName">First Name</label>
                <input
                  type="text"
                  class="form-control"
                  id="firstName"
                  name="firstName"
                />
```

```
</div>
<div class="form-group">
  <label for="lastName">Last Name</label>
  <input
    type="text"
    class="form-control"
    id="lastName"
    name="lastName"
  />
</div>
<div class="form-group">
  <label for="gender">Gender</label>
  <div>
    <label for="male" class="radio-inline"
      ><input
        type="radio"
        name="gender"
        value="m"
        id="male"
      />Male</label
    >
    <label for="female" class="radio-inline"
      ><input
        type="radio"
        name="gender"
        value="f"
        id="female"
      />Female</label
    >
    <label for="others" class="radio-inline"
      ><input
        type="radio"
        name="gender"
        value="o"
        id="others"
      />Others</label
    >
  </div>
</div>
<div class="form-group">
  <label for="email">Email</label>
  <input
    type="text"
    class="form-control"
    id="email"
    name="email"
  />
</div>
<div class="form-group">
  <label for="password">Password</label>
```

```
                <input
                  type="password"
                  class="form-control"
                  id="password"
                  name="password"
                />
              </div>
              <div class="form-group">
                <label for="number">Phone Number</label>
                <input
                  type="number"
                  class="form-control"
                  id="number"
                  name="number"
                />
              </div>
              <input type="submit" class="btn btn-primary" />
            </form>
          </div>
          <div class="panel-footer text-right">
            <small>&copy; Technical Babaji</small>
          </div>
        </div>
      </div>
    </div>
    <div class="toast" role="alert" aria-live="assertive" aria-atomic="true">
  <div class="toast-header">
    <img src="..." class="rounded mr-2" alt="...">
    <strong class="mr-auto">Bootstrap</strong>
    <small>11 mins ago</small>
    <button type="button" class="ml-2 mb-1 close" data-dismiss="toast" aria-
label="Close">
      <span aria-hidden="true">&times;</span>
    </button>
  </div>
  <div class="toast-body">
    Hello, world! This is a toast message.
  </div>
</div>
  </body>
</html>
```

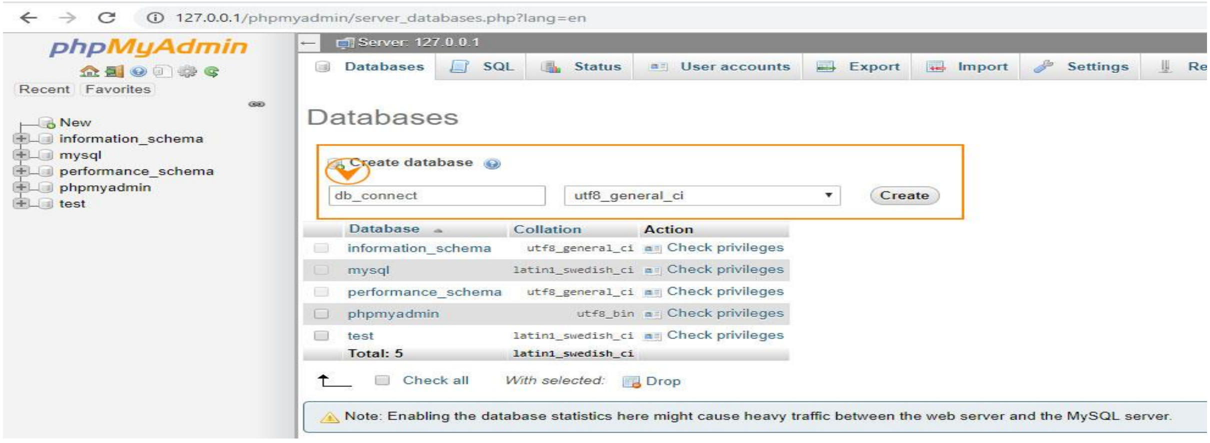
CONNECT.PHP

```
<?php
		$firstName = $_POST['firstName'];
		$lastName = $_POST['lastName'];
		$gender = $_POST['gender'];
		$email = $_POST['email'];
		$password = $_POST['password'];
		$number = $_POST['number'];

		// Database connection
		$conn = new mysqli('localhost','root','','test');
		if($conn->connect_error){
			echo "$conn->connect_error";
			die("Connection Failed : ". $conn->connect_error);
		}
		 else
		   {
			$stmt = $conn->prepare("insert into
	registration(firstName, lastName, gender, email, password, number)
	values(?, ?, ?, ?, ?, ?)");
			$stmt->bind_param("sssssi", $firstName, $lastName,
	$gender, $email, $password, $number);
			$execval = $stmt->execute();
			echo $execval;
			echo "Registration successfully...";
			$stmt->close();
			$conn->close();
		}
	?>
```

Step 3: Create a MySQL database

In this step, we'll create a simple MySQL database since our server is already running.

We'll open the browser and type http://localhost/phpmyadmin/.



127.0.0.1/phpmyadmin/server_databases.php?lang=en

Server: 127.0.0.1

Databases

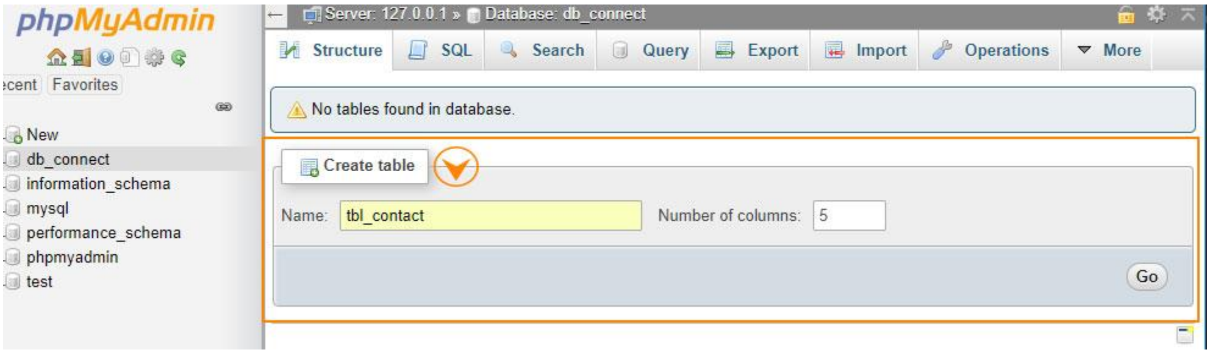
Create database

db_connect | utf8_general_ci | Create

Database	Collation	Action
information_schema	utf8_general_ci	Check privileges
mysql	latin1_swedish_ci	Check privileges
performance_schema	utf8_general_ci	Check privileges
phpmyadmin	utf8_bin	Check privileges
test	latin1_swedish_ci	Check privileges
Total: 5	latin1_swedish_ci	

Check all With selected: Drop

Note: Enabling the database statistics here might cause heavy traffic between the web server and the MySQL server.



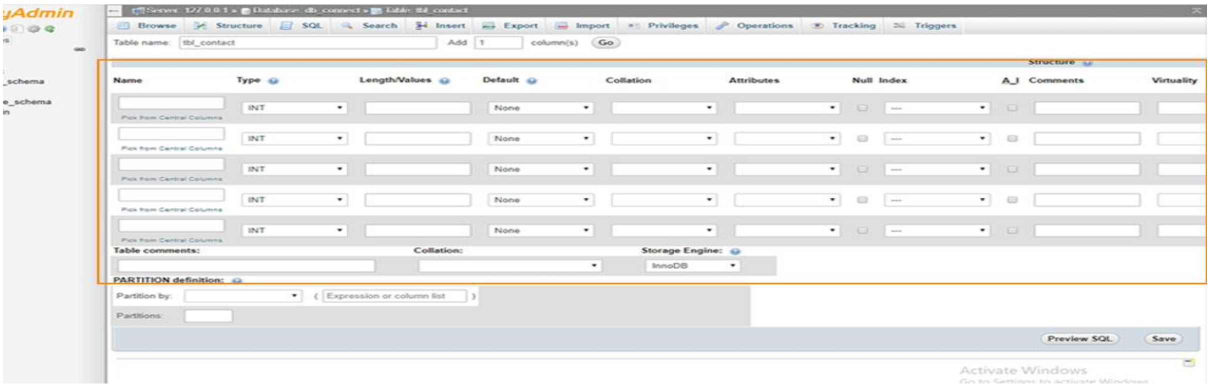
Server: 127.0.0.1 » Database: db_connect

No tables found in database.

Create table

Name: tbl_contact Number of columns: 5

Go



Server: 127.0.0.1 » Database: db_connect » Table: tbl_contact

Table name: tbl_contact Add 1 column(s) Go

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A_I	Comments	Virtuality
	INT		None			□	---	□		
	INT		None			□	---	□		
	INT		None			□	---	□		
	INT		None			□	---	□		
	INT		None			□	---	□		

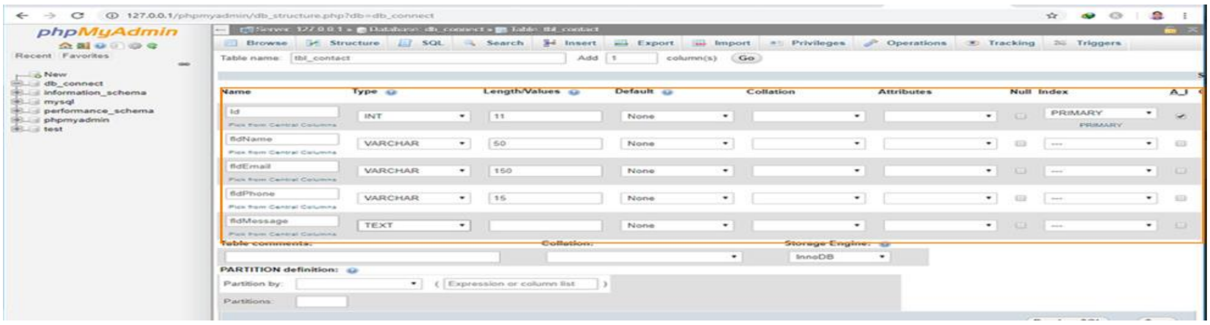
Table comments: Collation: Storage Engine: InnoDB

PARTITION definition:

Partition by: (Expression or column list)

Partitions:

Preview SQL Save



127.0.0.1/phpmyadmin/db_structure.php?db=db_connect

Server: 127.0.0.1 » Database: db_connect » Table: tbl_contact

Table name: tbl_contact Add 1 column(s) Go

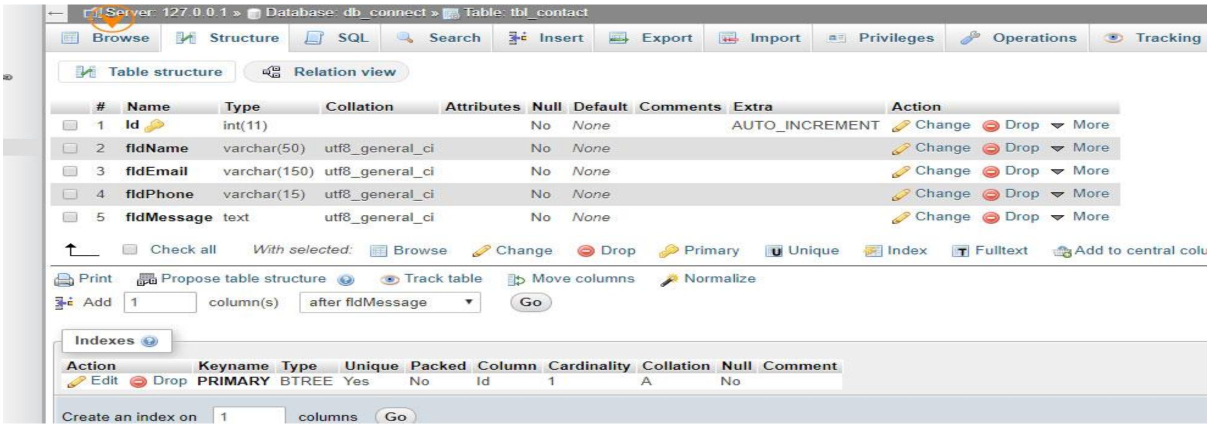
Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A_I
id	INT	11	None			□	PRIMARY	☑
fldName	VARCHAR	50	None			□	---	□
fldEmail	VARCHAR	150	None			□	---	□
fldPhone	VARCHAR	15	None			□	---	□
fldMessage	TEXT		None			□	---	□

Table comments: Collation: Storage Engine: InnoDB

PARTITION definition:

Partition by: (Expression or column list)

Partitions:


Server: 127.0.0.1 » Database: db_connect » Table: tbl_contact
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Table structure
Relation view

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**Id**	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
□	2	**fldName**	varchar(50)	utf8_general_ci		No	*None*			Change Drop More
□	3	**fldEmail**	varchar(150)	utf8_general_ci		No	*None*			Change Drop More
□	4	**fldPhone**	varchar(15)	utf8_general_ci		No	*None*			Change Drop More
□	5	**fldMessage**	text	utf8_general_ci		No	*None*			Change Drop More

□ Check all *With selected:* Browse Change Drop Primary Unique Index Fulltext Add to central colu
Print
Propose table structure
Track table
Move columns
Normalize
Add 1 column(s) after fldMessage Go
Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	**PRIMARY**	BTREE	Yes	No	Id	1	A	No	

(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.

Steps to Display Data From MySQL Database with PHP

You can test yourself to display data with the following folder structure –

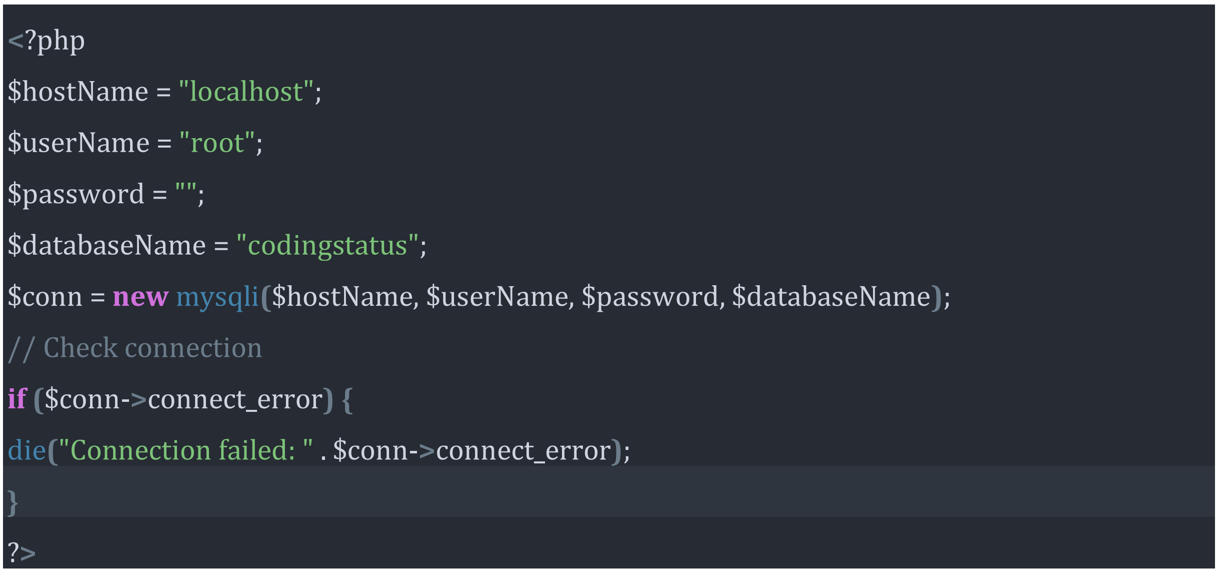
```
codingstatus/
      |__database.php
      |__table.php
      |__developers.php
      |
```

1. Connect PHP to MySQL Database

You can use the following database connection query to connect PHP to the MySQL database

- $hostName – It contains host name
- $userName – It contains database username
- $password – It contains database password
- $databaseName – It contains database name.

File Name – database.php

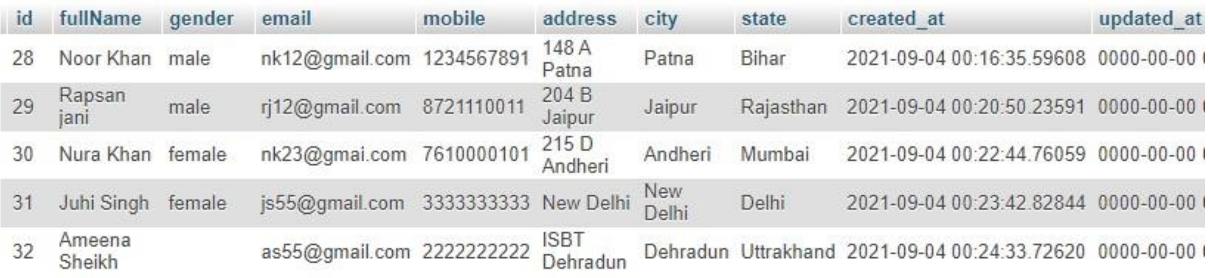


```
<?php
$hostName = "localhost";
$userName = "root";
$password = "";
$databaseName = "codingstatus";
$conn = new mysqli($hostName, $userName, $password, $databaseName);
// Check connection
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}
?>
```

2. Insert Data Into PHPMyAdmin Table

Before displaying data, you have to insert data into the Database. If you have not already done it and don't know about it, then you can learn it through the following URL –

[Insert Data into Database using PHP & MySQL]



id	fullName	gender	email	mobile	address	city	state	created_at	updated_at
28	Noor Khan	male	nk12@gmail.com	1234567891	148 A Patna	Patna	Bihar	2021-09-04 00:16:35.59608	0000-00-00
29	Rapsan jani	male	rj12@gmail.com	8721110011	204 B Jaipur	Jaipur	Rajasthan	2021-09-04 00:20:50.23591	0000-00-00
30	Nura Khan	female	nk23@gmai.com	7610000101	215 D Andheri	Andheri	Mumbai	2021-09-04 00:22:44.76059	0000-00-00
31	Juhi Singh	female	js55@gmail.com	3333333333	New Delhi	New Delhi	Delhi	2021-09-04 00:23:42.82844	0000-00-00
32	Ameena Sheikh		as55@gmail.com	2222222222	ISBT Dehradun	Dehradun	Uttrakhand	2021-09-04 00:24:33.72620	0000-00-00

3. Fetch Data From MySQL Table

Now, You have to fetch data from the MySQL table. So, just follow these points –

- First of all, include a database connection file database.php
- assign $conn to a new variable $db and table name to another variable $table

- Define columns name in an indexed array and assign them to the $columns
- Also, assign fetch_data() function to the $fetchData

fetch_data() – This function accepts three parameters like $db, $table & $column and It contains MySQLi SELECT query that will return records in an array format by fetching from the database

File name – developers.php

```
<?php
include("database.php");
$db= $conn;
$tableName="developers";
$columns= ['id', 'fullName','gender','email','mobile', 'address','city','state'];
$fetchData = fetch_data($db, $tableName, $columns);
function fetch_data($db, $tableName, $columns){
if(empty($db)){
$msg= "Database connection error";
}elseif (empty($columns) || !is_array($columns)) {
$msg="columns Name must be defined in an indexed array";
}elseif(empty($tableName)){
$msg= "Table Name is empty";
}else{
$columnName = implode(", ", $columns);
$query = "SELECT ".$columnName." FROM $tableName"." ORDER BY id DESC";
$result = $db->query($query);
if($result== true){
if ($result->num_rows > 0) {
$row= mysqli_fetch_all($result, MYSQLI_ASSOC);
$msg= $row;
} else {
$msg= "No Data Found";
}
}else{
$msg= mysqli_error($db);
}
}
return $msg;
}
?>
```

4. Display Data in HTML Table

Now, You have to display data in the HTML table. So, you have to follow these points –

- First of all, Include PHP script file developers.php
- Create an HTML table using Bootsrap 4
- Check $fetchData is an array or not with if & else condition
- Then apply foreach loop to the $fetchData
- After that print the required data in the table

File Name – table.php

```
<?php

include("developers.php");

?>

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

</head>

<body>

<div class="container">

<div class="row">

<div class="col-sm-8">

<?php echo $deleteMsg??''; ?>

<div class="table-responsive">

<table class="table table-bordered">

<thead><tr><th>S.N</th>

<th>Full Name</th>

<th>Gender</th>

<th>Email</th>

<th>Mobile Number</th>

<th>Address</th>

<th>City</th>

<th>State</th>
```

```
</thead>
<tbody>
<?php
if(is_array($fetchData)){
$sn=1;
foreach($fetchData as $data){
?>
<tr>
<td><?php echo $sn; ?></td>
<td><?php echo $data['fullName']??''; ?></td>
<td><?php echo $data['gender']??''; ?></td>
<td><?php echo $data['email']??''; ?></td>
<td><?php echo $data['mobile']??''; ?></td>
<td><?php echo $data['address']??''; ?></td>
<td><?php echo $data['city']??''; ?></td>
<td><?php echo $data['state']??''; ?></td>
</tr>
<?php
$sn++;}}else{ ?>
<tr>
<td colspan="8">
<?php echo $fetchData; ?>
</td>
<tr>
<?php
}?>
</tbody>
</table>
</div>
</div>
</div>
</div>
</body>
</html>
```

Module 9

*a)*Develop a PHP web Application which Demonstrates the process of deleting the particular record in mysql database table

PHP MySQL Delete Record

PHP mysql_query() function is used to delete record in a table. Since PHP 5.5, **mysql_query()** function is *deprecated*. Now it is recommended to use one of the 2 alternatives.

- **mysqli_query()**
- **PDO::__query()**

PHP MySQLi Delete Record Example

```
<?php
$server="localhost";
$uname="root";
$pwd="";
$db="csec";
$con=mysqli_connect($server,$uname,$pwd,$db);
if(!$con)
{
die("connection failed");
}
$sql="delete from emp where id=2";
if(mysqli_query($con,$sql))
{
```

```
echo "record deleted successfully";
}
else
{
echo "unable to delete record";
}
mysqli_close($con);
?>
```

b)Develop a php web application to illustrate the process of updating my sql database table data.

PHP MySQL Update Record

PHP mysql_query() function is used to update record in a table. Since PHP 5.5, **mysql_query()** function is *deprecated*. Now it is recommended to use one of the 2 alternatives.

- **mysqli_query()**
- **PDO::_query()**

PHP MySQLi Update Record Example

```
<?php
$server="localhost";
$uname="root";
$pwd="";
$db="csec";
$con=mysqli_connect($server,$uname,$pwd,$db);
if(!$con)
```

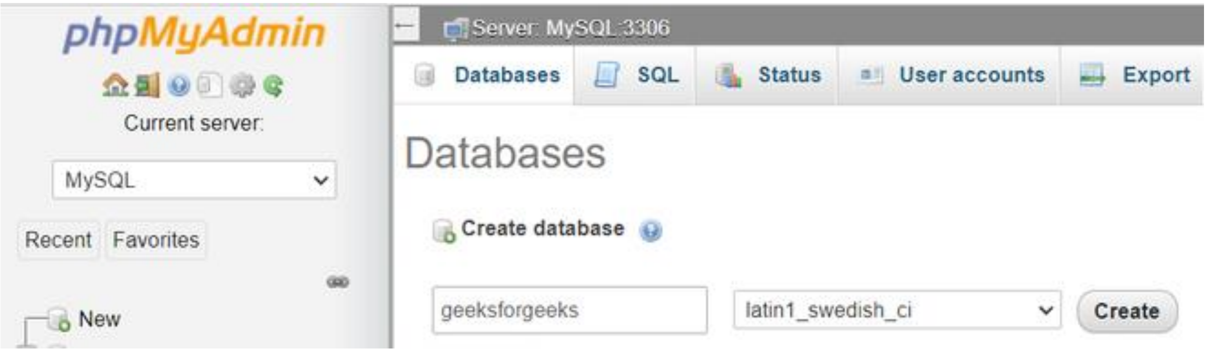
```
{
die("connection failed");
}
$sql="update emp set adress='vza' where id=1";
if(mysqli_query($con,$sql))
{
echo "record updates successfully";
}
else
{
echo "unable to update record";
}
mysqli_close($con);
?>
```

MODULE 10

A)Develop a PHP web Application which demonstrates how to upload image into database and display it.

Uploading the image/videos into the database and displaying it using PHP is the way of uploading the image into the database and fetching it from the database. Using the PHP code, the user uploads the image or videos they are safely getting entry into the database and the images should be saved into a particular location by fetching these images from the database.
If any of the websites contain the functionality to upload images/videos with some detail, then by using this code we will upload the image into your database and whether you would like to ascertain what the person has got to be uploaded. And by this code the image which is uploaded that where save in your system where you are given the location.

Approach: Make sure you have XAMPP or WAMP server installed on your machine. In this tutorial, we will be using the WAMP server.
1. Create Database: First, we will create a database named '**geeksforgeeks**'. You can use your existing database or create a new one.



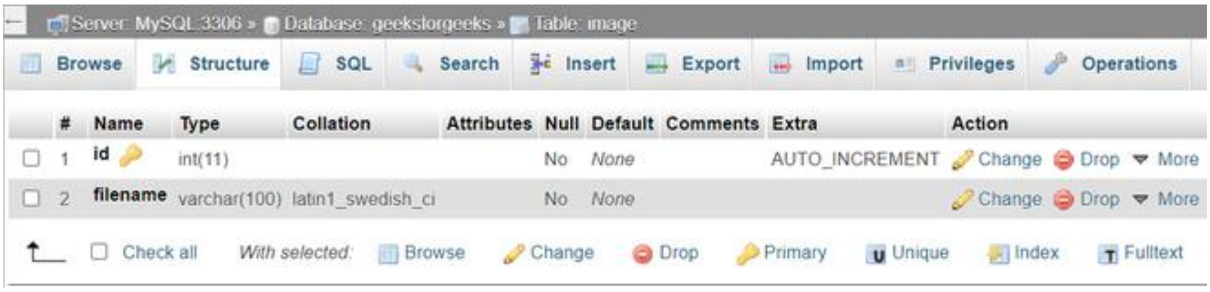
geeksforgeeks

latin1_swedish_ci

2. Create Table: Create a table named '**image**'. The table contains two fields:

- id – int(11)
- filename – varchar(100)

The id should be in **Auto incremented(AI)**. Your table structure should look like this:



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	id	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
□	2	filename	varchar(100)	latin1_swedish_ci		No	*None*			Change Drop More

Program: Now, we will create an HTML form for uploading image files (you can upload any type of file like .pdf or .mp4) and will display the uploaded image.

- **HTML code:**

```
<!DOCTYPE html>

<html>



<head>

        <title>Image Upload</title>

        <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css">

        <link rel="stylesheet" type="text/css" href="style.css" />

</head>



<body>

        <div id="content">

                <form method="POST" action="" enctype="multipart/form-data">

                        <div class="form-group">

                                <input class="form-control" type="file" name="uploadfile" value="" />

                        </div>

                        <div class="form-group">

                                <button class="btnbtn-primary" type="submit" name="upload">UPLOAD</button>

                        </div>
        </form>
    </div>
    <div id="display-image">
    <?php
        $query = " select * from image ";
        $result = mysqli_query($db, $query);

        while ($data = mysqli_fetch_assoc($result)) {
    ?>
        <imgsrc="./image/<?php echo $data['filename']; ?>">

    <?php
        }
    ?>
    </div>
</body>

</html>
```

- **CSS code:** The **style.css** is the file that styles the form into a new design and the code is given below.

```
*{
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

#content{
    width: 50%;
    justify-content: center;
    align-items: center;
    margin: 20px auto;
    border: 1px solid #cbcbcb;
}
form{
    width: 50%;
    margin: 20px auto;
}

#display-image{
    width: 100%;
    justify-content: center;
    padding: 5px;
    margin: 15px;
}
img{
    margin: 5px;
    width: 350px;
    height: 250px;
}
```

PHP code: The PHP code is for the uploading images, the file name is saved with the **index.php**, you can also save it with another name as you prefer.

```
<?php
error_reporting(0);
```

```
$msg = "";


// If upload button is clicked ...
if (isset($_POST['upload'])) {


    $filename = $_FILES["uploadfile"]["name"];
    $tempname = $_FILES["uploadfile"]["tmp_name"];
    $folder = "./image/" . $filename;


    $db = mysqli_connect("localhost", "root", "", "geeksforgeeks");


    // Get all the submitted data from the form
    $sql = "INSERT INTO image (filename) VALUES ('$filename')";


    // Execute query
    mysqli_query($db, $sql);


    // Now let's move the uploaded image into the folder: image
    if (move_uploaded_file($tempname, $folder)) {
        echo "<h3> Image uploaded successfully!</h3>";
    } else {
        echo "<h3> Failed to upload image!</h3>";
    }
}
?>
```

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.

JSON stands for JavaScript Object Notation. Like XML, it is a text-based format for the exchange of data which is easier to read and write and it is lighter than other formats. JSON is based on two basic structures namely Objects and Arrays.

Parsing JSON data in PHP: There are built-in functions in PHP for both encoding and decoding JSON data. These functions are **json_encode()** and **json_decode()**. These functions works only with UTF-8 encoded string.

Decoding JSON data in PHP: It is very easy to decode JSON data in PHP. You just have to use **json_decode()** function to convert JSON objects to the appropriate PHP data type.

Example: By default the **json_decode()** function returns an object. You can optionally specify a second parameter that accepts a boolean value. When it is set as “true”, JSON objects are decoded into associative arrays.

PHP

```
<?php

        $student_data = '{"Ram":96,"Prashant":76,"Varun":65,"Mark":34}';



        // Decoding above JSON String into JSON object

        $decoded = json_decode($student_data);



        // Dump the $decoded variable

        var_dump($decoded);

?>
```

Encoding JSON data in PHP: Encoding JSON data is as easy as encoding JSON data in PHP. We use the **json_encode()** function, the data being encoded can be any PHP data type except a resource like a filehandle.

Example 1: The following code demonstrates encoding PHP associative array.

PHP

```
<?php



        // PHP associative array
        $student_data = array(
                "Ram"=>96,
                "Prashant"=>76,
                "Varun"=>65,
                "Mark"=>34
        );

        // Encoding PHP Associative array using json_encode()
        $encoded = json_encode($student_data);

// Echo the data
        echo $encoded;
?>
```

Output:

```
{"Ram":96,"Prashant":76,"Varun":65,"Mark":34}
```