

**NAME : DATE VAISHNAVI BHALCHANDRA**

**DIV : A ROLL NO: 30**

**SUB: WT LAB**

Name: Vaishnavi Date  
Div: A Roll no: 30  
WT Lab

### Experiment No: 1

Aim: Tomcat Server installation

Objectives: Understand how to install tomcat  
Server & setup environment variables

software & Hardware: Java 7  
Apache tomcat server

Theory: web Application -

- 1] HTTP Server - eg. Google webserver, Apache http server, Tomcat server
- 2] HTTP Client - eg. Internet explorer, Firefox, Google Chrome.
- 3] Database - eg. MySQL, MS SQL Server, DB, etc.
- 4] Client Side Program - It can be written HTML for JavaScript etc.
- 5] Serverside Program - could be written in java serv JSP, ASP, PHP, Perl.

A web appl<sup>n</sup> is 3tier client-server database appl<sup>n</sup>

### Apache Tomcat:

- Is an open source project under Apache SLW found
- The other side for Tomcat is <http://tomcat.apache.org>

### Execution steps:

- 1] Go to <http://tomcat.apache.org>
- 2] Select version & from left side list under download
- 3] Download & run '.exe' file
- 4] Use default setting & provide password that you will remember
- 5] Then install tomcat

### How to run Tomcat:

- 1] Find start program in programs menu look under Apache Tomcat & select 'start Tomcat' or 'startup bat'.
- 2] Open any web browser & type given URL  
<http://localhost:8080/>

### How to set a path for windows:

Through Command prompt:

Set path: `C:/Program Files (x86)/Java/jdk/7.0/bin`

### Conclusion:

We have learned how to install & configure tomcat server.

Name: Vaishnavi Date

Div: A Roll no: 30

WT Lab

Experiment No: 2Aim: HTML, CSS, XMLObjectives: Understand Basic concept of HTML, CSS & XMLProblem statement: WAP to design registration form for students by using HTML & CSSSW & HW: Notepad & BrowserTheory: HTML:

- HTML stands for Hyper Text Markup language
- HTML describes structure of web pages
- HTML elements are building block of HTML pages & represented by tags.

<u>HTML Versions:</u>	HTML	1991
	HTML 2.0	1995
	HTML 3.2	1997
	HTML 4.01	1999
	XHTML	2000
	HTML 5	2014

CSS:

- CSS stands for Cascading style sheets
- CSS handles text & look part of web page
- By using CSS one can control color of text, style



## CSS Modules

- BOX Model - Background
- selectors - Text effect, Animations
- Border - 2D/3D transformation

## User Interface

## Technology Tool:

- <!DOCTYPE html> define HTML5
- <html> Root element of html page
- <head> element contain meta information
- <title> Specifies a title for document
- <body> element contain visible page content
- <h1> large heading
- <p> defines paragraph

CSS can be added to HTML in 3 ways -

- ① Inline - By using style attribute in html element
- ② Internal - Using <style> element in <head> section
- ③ External - By using external CSS file

Steps: 1] Write HTML code in notepad & save with .html extension & CSS code & save with .css extension

- 2] Import CSS file in HTML Page
- 3] Open HTML page in Browser

## Conclusion:

Hence, we have designed static web pages using HTML & CSS.

## **Program:**

```
<html>
<head>
<title>Registration Form </title>
<style>
.con
{
width:400px;
height:400px;
background-color:pink;
color:blue;
padding:0px 10px 0px 10px;
margin:auto;
}
.main
{
width:400px;
height:80px;
float:left;
padding:60px 10px 10px 10px;
margin: auto;
}
.con1
{
width:200px;
height:30px;
float:left;
margin:auto;
```

padding:0px 0px 0px0px;

}

</style>

<body>

<div class="con">

<form>

<div class="main">

<table align="center">

<tr>

<td><b><br>Welcome to Registration

Page<br><br></td></tr>

</table>

<div class="con1">

<table align="center">

<tr>

<td>Full Name:</td>

<td><input type="text" value=""></input></td>

</tr>

<tr>

<td>Class</td>

<td><input type="text" value=""></input></td>

</tr>

<tr>

<td>Department</td>

<td><input type="text" value=""></input></td>

</tr>

<tr>

<td>address:</td>

<td><textarea row="3" column="2"></textarea></td>

</tr>

<tr>

<td>email:</td>

<td><input type="text" value=""></input></td>

</tr>

<tr>

<td>phone:</td>

<td><input type="text" value=""></input></td>

</tr>

<tr></tr><tr></tr>

</table>

<table align="center">

<tr>

<td><input type="submit" value="SUBMIT"></td>

<td><input type="reset" value="CANCEL"></td>

</tr>

</table>

</div>

</div>

</form>

</div>

</body>

</html>

Name: Vaishnavi Date  
Div: A Roll no: 30  
WT Lab

### Experiment No: 3

Aim: XML & CSS

Problem Statement: WAP to design book catalog by using XML & CSS to display title, author, price & year of the book

SW & HW: Notepad, Any Browser

#### Theory:

- XML stands for extensible Markup language. It is nothing but text-based marked up language
- XML tags identify data & used to store & organize data rather than specify how to display it like HTML tags which are used to display data.
- characteristics - (1) extensible  
(2) carries data, does not present it  
(3) Public standard.

Technology: The XML document have XML declaration but it is optional. `<?xml version='1.0' encoding='UTF-8' ?>` each XML-element needs to be closed either with start or end elements as shown  
`<element> - - - </element>`

XML document can have only one root element  
`<root> <x> - - - </x> <y> - - - </y> </root>`



XML Attributes: Using a name/value pair, attribute specifies a single property for an element. An XML element can have one or more attribute.

```
<a href = 'http://www.google.com'> XML Tutorial </a>
```

### Design Steps:

- 1] write XML code in notepad & save with .xml extension.
- 2] write CSS code in notepad & save with .css extension
- 3] import CSS file in XML page
- 4] open XML page in browser

### Conclusion:

Hence, we have design static web pages using XML & CSS.

**Program:**

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="ass3.css"?>
<CATALOG>
<BOOK>
<TITLE>Database Management System</TITLE>
<AUTHOR>Korth</AUTHOR>
<PRICE>500</PRICE>
<YEAR>1985</YEAR>
</BOOK>
<BOOK>
<TITLE>Computer Network</TITLE>
<AUTHOR>Tenenbaum</AUTHOR>
<PRICE>600</PRICE>
<YEAR>1985</YEAR>
</BOOK>
<BOOK>
<TITLE>Software Engineering and project Management</TITLE>
<AUTHOR>Roger Pressman</AUTHOR>
<PRICE>600</PRICE>
<YEAR>1985</YEAR>
</BOOK>
</CATALOG>
```

.css file:

BOOK {

Display: block;

margin-left:0;

margin-bottom: 30pt;

}

CATALOG {

Width:100%;

background-color: #ffffff;

}

TITLE {

Color: #ff0000;

display: block;

font-size: 20pt;

}

AUTHOR {

display: block;

color: #0000ff;

font-size: 20pt;

}

YEAR, PRICE {

Color:#000000;

Display: block;

Margin-left: 20pt;

}

Name: Vaishnavi, Date

Div: A Roll no: 20

WT Lab

Experiment No: 4Aim: HTML, JavaScriptObjective:

1. Understand Basic concept of JS.
2. Use Javascript for validation of data

Problem Statement: WAP to design registration form for students by using HTML, CSS & Javascript & perform validation all fields are mandatory. Phone no & email address.

Software: Notepad, Any browser

Theory: JavaScript is a programming language of HTML as well as web. It is preferred for creating n/w centric application.

Advantages:

- It requires less server interaction
- Immediate feedback to visitors
- Increased interactivity
- Richer interfaces

Validation: When client enters all necessary data & press submit button from validation is done at server side if data entered by client is



incorrect or missing, server needs to send all data back to client & request for resubmission of form must be checked to make sure all mandatory fields are filled in & check its value.

Technology: It can be implemented using JS  
Statement that are place within `<script>`  
language - This attribute specifies with Scripting  
language you are using.

Type - This attribute is what is now recommended  
to indicate Scripting language in use.

Test cases: All fields like full name, class,  
dept, address, phone no, email id  
are mandatory fields.

- Phone no should be in numbers only
- Email id should be in proper format  
like abc@gmail.com

Conclusion:

Hence, we applied validate data  
using Java Script.

## **Program:**

```
<html>
<head>
<title>Registration Form Validation</title>
<style>
.con
{
width:400px;
height:400px;
background-color:pink;
color:blue;
padding:0px 10px 0px 10px;
margin:auto;
}
.main
{
width:400px;
height:80px;
float:left;
padding:60px 10px 10px 10px;
margin: auto;
}
.con1
{
width:200px;
height:30px;
float:left;
```

```
margin:auto;
padding:0px 0px 0px0px;
}
```

```
</style>
```

```
<script type="text/javascript">
```

```
function Validation()
```

```
{
```

```
    varemailID = document.myForm.EMail.value;
```

```
    varnum=document.myForm.ph.value;
```

```
    atpos = emailID.indexOf("@");
```

```
    dotpos = emailID.lastIndexOf(".");
```

```
    if (atpos< 1 || ( dotpos - atpos< 2 ))
```

```
    {
```

```
        alert("Please enter correct email ID")
```

```
        document.myForm.EMail.focus() ;
```

```
        return false;
```

```
    }
```

```
    else if (isNaN(num)){
```

```
        document.getElementById("ph");
```

```
        alert("Please enter numeric only.")
```

```
        return false;
```

```
    }else{
```

```
        return true;
```

```
    }
```

```
}
```

```
</script>
</head>
<body>
<div class="con">
<form>
<div class="main">
<table align="center">
<tr><td><b><br>Welcome to Registration
Page<br><br></td></tr> </table>
<div class="con1">
<table align="center">
<tr>
<td>Full Name:</td>
<td><input type="text" id="FirstName" value="" required></input></td> </tr>
<tr>
<td>Class:</td>
<td><input type="text" id="Class" value="" required></input></td>

</tr>
<tr>
<td>Department:</td>
<td><input type="text" id="Department" value="" required></input></td>

</tr>
<tr>
<td>address:</td>
<td><textarea row="3" column="2" value="" required></textarea></td>
</tr>
```



```
<tr>
<td>email Id:</td>
<td><input type="text" value="" id="EMail"required></input></td>
```

```
</tr>
```

```
<tr>
```

```
<td>phone no:</td>
```

```
<td><input type="text" id="ph" value="" required></input></td>
```

```
</tr>
```

```
<tr>
```

```
<tr></tr><tr></tr></table>
```

```
<table align="center">
```

```
<td><input type="submit" value="SUBMIT" onclick="return
```

```
(Validation());"></td>
```

```
<td><input type="reset" value="CANCEL"></td>
```

```
</tr>
```

```
</table>
```

```
</div>
```

```
</div>
```

```
</form>
```

```
</div>
```

```
</body>
```

```
</html>
```

Name: Vaishnavi Date

Div: A Roll no: 30

WT Lab.

Experiment No: 5Aim: JSP, Servlet & MySQL (Backend)Objective: understand Basic Concept of html, css  
understand basic functionalities of JSP  
Having knowledge of SQL query of Create DB.Problem Statement:

- 1] Design & build student login Page using JSP, Servlet, MySQL.
- 2] Design & Build employee login page using JSP, Servlet & MySQL.

Software: Any Operating System, JDK, Editors -  
NetBeans/Eclipse, Web browser, Tomcat 7Theory: Java Server pages -

- It is a Server side programming technology i.e. used to create dynamic web-based application
- It is a technology helps slw developers to create dynamic web pages based on HTML, XML & other document types.
- A JSP element is type of java servlet i.e. designed accomplished role of user interface for a java web appl<sup>n</sup>
- Using JSP, you can collect i/p from users through webpage forms current records from database.

Why we need JSP:- JSP is used for design dynamic web page & servlet is used to code the logic i.e. present in MVC architecture. Servlet is controller & JSP preview.

Syntax of JSP:

```
declare variables <% text %>
// Hello.jsp
<html> <head>
<title> Jsp title </title> </head>
<body> <%
out.println("welcome to JSP class");
%>
</body>
</html>
```

O/P → welcome to JSP class

Technology:

1. JSP & Servlet
2. IDE - Net Beans 7.0
3. Database - MySQL

Test Cases: Manual testing is used to validate fields like username, password, Mob no & email ids of user entered by user with database.

Conclusion: Hence, we have performed dynamic web application using JSP, Servlet & MySQL.

```
// index.jsp
```

```
<%--
```

Document : Login

Created on : Dec 31, 2017, 2:07:34 PM

Author : Admin --

```
%>
```

```
<% @page contentType="text/html" pageEncoding="UTF-8"%>
```

```
<!DOCTYPE html>
```

Page 29

35

Department of Computer Engineering2019-20

Web Technology Lab Manual

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"> <title> Login </title>
```

```
</head>
```

```
<body>
```

```
<table>
```

```
<tr>
```

```
<td width="300"></td>
```

```
<td> <font color="#339900" size="3.5">User Login</font>
```

```
<table>
```

```
<tr>
```

```
<td>
```

```
<form id="user1" method="post" action="User.jsp" name="s" onSubmit="return valid()">
```

```
<table border="2.0">
```



```
<tr>
<td>
User Id</td>
<td><input type="text" name="user1"></td>
</tr>
<tr>
<td>
Password</td>
<td><input type="password" name="pass"></td>
</tr>
<tr>
<td>
<center><input type="submit" name="user" value="Login"></center>
</td>
</tr></table>
<center><table>
<tr>
<td>
<a href="SignUp.jsp"><font color="#000000" size="2"><b>New
User?</b></font></a></td></tr></table>
</td>
</tr></table></center>
</td>
</tr>
</table>
</form>
</body>
</html>
```

```

//SignUp.jsp
<% @ page import="java.sql.*"%>
<% @ page import =
"java.util.Date,java.text.SimpleDateFormat,java.text.ParseException"%> <% --
Document : SignUp
Created on : Dec 30, 2017, 11:52:03 PM
Author : Admin
--%>
<% @page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-
8"> <title>JSP Page</title>
</head>
<body bgcolor="Skyblue">
<%
java.util.Date now = new java.util.Date();
String DATE_FORMAT1 = "dd/MM/yyyy";
SimpleDateFormat sdf1 = new SimpleDateFormat(DATE_FORMAT1);
String strDateNew1 = sdf1.format(now);
%>
<table border="black" bgcolor=" black" align ="center" >
<tr>
<td> <strong><em><font color="#990000" size="+1" face="Times New Roman"
style="text-decoration:underline">User Registration</font></em></strong>
</td>
</tr>

```

```

</table>

<table border="black" align="center">

<tr>

<td><form name="s" action="LoginServlet.jsp" method="get" onSubmit="return
valid()">

<table align="center" cellpadding="6" cellspacing="6" width="400" height="300">

<td>

<tr>

<td><font face="Times New Roman" size="+1"><strong>Name</strong></font></td>

<td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="text" name="unn" class="b"></td>

</tr>

<tr>

<td><font face="Times New Roman" size="+1"><strong>User ID</strong></font></td>

<td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="text" name="uidd" id="name"></td>

</tr>

<tr>

<td><font face="Times New Roman"
size="+1"><strong>Password</strong></font></td>

<td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="password" name="pass" class="b"></td>

</tr>

<tr>

<td><font face="Times New Roman" size="+1"><strong>Mobile</strong></font></td>

<td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="text" name="mobb" class="b"></td>

</tr>

<tr>

<td><font face="Times New Roman" size="+1"><strong>Email
ID</strong></font></td>

<td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="text" name="eidd" class="b"></td>

```

[illegible]



```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-
8"> <title>JSP Page</title>
</head>
<body>
<h1> User Logged In Successfully!...</h1>
</body>
</html>
```

//LoginFailure.jsp

<%--

Document : LoginFailure

Created on : Dec 31, 2017, 2:42:21 PM

Author : Admin

--%>

<% @page contentType="text/html" pageEncoding="UTF-8"%>

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-
8"> <title>JSP Page</title>
```

```
</head>
```

```
<body>
```

```
<h1> <font color="Red"> Please Provide Correct Username or Password!..Retry Again!..
```

```
</font> </h1>
```

```
<table>
```

```
<tr>
```

```

<td>

<form id="user1" method="post" action="User.jsp" name="s" onSubmit="return
valid()">

<table>

<tr>

<td>

User Id</td>

<td><input type="text" name="user1"></td>

</tr>

<tr>

<td>

Password</td>

<td><input type="password" name="pass"></td>

</tr>

<tr>

<td>

</td>

<td>

<input type="submit" name="user"
value="Login"></td> </tr></table>

</form>

</td>

</tr></table>

</body>

</html>

//LoginServlet.jsp

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

<% @ page import="java.io.*"%>

```

```
<% @ page import = "java.util.Date,java.text.SimpleDateFormat,java.text.ParseException"%>
<%
String a = request.getParameter("uid");
String x = request.getParameter("unn");
String b = request.getParameter("pass");
String c = request.getParameter("mob");
String d = request.getParameter("eid");
session.setAttribute("d",d);
//String f=request.getParameter("date");
//out.print(strDateNew1);
//out.print(x);
//out.print(b);
//out.print(c);
//out.print(d);
//out.print(f);
java.util.Date now = new java.util.Date();
String DATE_FORMAT = "yyyy-MM-dd";
SimpleDateFormat sdf = new SimpleDateFormat(DATE_FORMAT);
String strDateNew = sdf.format(now) ;
//response.sendRedirect("signup.jsp?message=success"); //String
userid=null;
//String m="avl";
try
{
Class.forName("com.mysql.jdbc.Driver");
Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/loginpage","root","root");
PreparedStatement ps=con.prepareStatement("insert into
```

```

student(sname,suserid,spass,mobile,email,date) values(?,?,?,?,?,"+strDateNew+"");
//ResultSet rs=ps.executeQuery();
ps.setString(1,x);
ps.setString(2,a);
ps.setString(3,b);
ps.setString(4,c);
ps.setString(5,d);
ps.executeUpdate();
response.sendRedirect("SignUp.jsp?success");
} catch(Exception e1)
{
out.println(e1.getMessage());
response.sendRedirect("SignUp.jsp?Failure");%>
}
%>
//User.jsp
<% @ page import="java.sql.*;"%>
<%
String a=request.getParameter("user1");
String b=request.getParameter("pass");
String id=null,name=null,userId=null,email=null;
try{
Class.forName("com.mysql.jdbc.Driver");
Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/loginpage","root","root");
//Connection con = databasecon.getConnection();
PreparedStatement ps=con.prepareStatement("select sid,sname,suserid,email from student where
suserid='"+a+"' && spass='"+b+"'");

```

```
ResultSet rs=ps.executeQuery();
if(rs.next())
{
    d=rs.getString("sid");
    name=rs.getString("sname");
    userid=rs.getString("suserid");
    email=rs.getString("email");
    session.setAttribute("sid",id);
    session.setAttribute("sname",name);
    session.setAttribute("suserid",userid);
    session.setAttribute("email",email);
    //response.sendRedirect("user5.jsp");
    response.sendRedirect("LoginSuccess.jsp?Success");
    //out.print(name2);
} else
{
    response.sendRedirect("LoginFailure.jsp?Failure");
}
}
catch(Exception e2){
    out.println(e2.getMessage());
}
%>
```

Name: Vaishnavi Dab  
Div: A Roll no: 30  
WT lab

### Experiment No: 6

Aim: Add dynamic web application, essence using PHP, HTML & MySQL

Objectives: To understand principles & methodologies of PHP web based applications development process.

Problem Statement: Design & develop dynamic web application using PHP & MySQL as a back-end for employee data with insert, delete, view & update options.

S/W & H/W req: Ubuntu 64 bit / windows XP.  
XAMPP Server, Intel P4 m/c + RAM & 32GB

Theory: PHP:

- PHP Hypertext Preprocessor (PHP) began as little open source venture that advanced as an ever increasing no. of individuals discovered how valuable it was.
- PHP is a Server side programming is installed in HTML. It is incorporated with various prevalent databases, including MySQL, Oracle, Sybase, Informix.
- PHP Performs framework Capabilities i.e. from document on a framework it can open, read, compose & close them.

PAGE: / /  
DATE: / /

```
<html> <head>
<title> Hello world </title> </head>
<body>
<? php
echo ('Hello php');
?> </body> </html>
```

O/P  $\Rightarrow$  Hello world.

Technology: Used to PHP & tool XAMPP server is used to execute PHP web application

- XAMPP server embeds the PHP, MySQL & phpmyadmin these 3 tools must be required to run php web appl.

Design | execution steps:

- 1] Download XAMPP & configure PHP, MySQL server & install.
- 2] After ab
- 3] over command following installation window appear
- 4] open browser & type 'localhost' in URL
- 5] Started XAMPP server
- 6] To navigate to root folder .(cd)
- 7] navigate htdocs cd opt 1/ xampp/htdocs
- 8] Create file
- 9] Go to browser type localhost/ filename.php
- 10] Create MySQL database.

Conclusion: we have studied how to design & develop small web appl<sup>n</sup> using PHP Script, xampp with Apache server & MySQL as backend.



## **Program:**

### **1. index.php**

```
<?php
//including the database connection file
include_once("config.php");
//fetching data in descending order (lastest entry first)
//$result = mysql_query("SELECT * FROM users ORDER BY id DESC"); // mysql_query is
deprecated
$result = mysqli_query($mysqli, "SELECT * FROM users ORDER BY id DESC"); // using
mysqli_query instead
```

Page 49

55

Department of Computer Engineering2019-20

Web Technology Lab Manual

```
?>
<html>
<head>
<title>Homepage</title>
</head>
<body>
<a href="add.html">Add New Data</a><br/><br/>
<table width='80%' border=0>
<tr bgcolor='#CCCCCC'>
<td>Name</td>
<td>Age</td>
<td>Email</td>
<td>Update</td>
</tr>
```

```

<?php
//while($res = mysql_fetch_array($result)) { // mysql_fetch_array is deprecated, we need to
use mysqli_fetch_array
while($res = mysqli_fetch_array($result)) {
echo "<tr>";
echo "<td>".$res['name']. "</td>";
echo "<td>".$res['age']. "</td>";
echo "<td>".$res['email']. "</td>";
echo "<td><a href=\"edit.php?id=$res[id]\">Edit</a> | <a
href=\"delete.php?id=$res[id]\" onClick=\"return confirm('Are you sure you want to
delete?')\">Delete</a></td>";
}
?>
</table>
</body>
</html>

```

## 2. add.html

```

<html>
<head>
<title>Employee Data</title>
</head>
<body>
<a href="index.php">Home</a>
<br/><br/>
<form action="add.php" method="post" name="form1">
<table width="25%" border="0">
<tr>

```

```
<td>Name</td>
<td><input type="text" name="name"></td>
</tr>
<tr>
<td>Age</td>
<td><input type="text" name="age"></td>
</tr>
<tr>
<td>Email</td>
<td><input type="text" name="email"></td>
</tr>
<tr>
<td></td>
<td><input type="submit" name="Submit" value="Add"></td>
</tr>
</table>
</form>
</body>
</html>
```

### 3. add.php

```
<html>
<head>
<title>Add Data</title>
</head>
<body>
<?php
//including the database connection file
```

```
include_once("config.php");

if(isset($_POST['Submit'])) {

$name = mysqli_real_escape_string($mysqli, $_POST['name']);
$age = mysqli_real_escape_string($mysqli, $_POST['age']);
$email = mysqli_real_escape_string($mysqli, $_POST['email']);

// checking empty fields

if(empty($name) || empty($age) || empty($email)) {

if(empty($name)) {

echo "<font color='red'>Name field is empty.</font><br/>";

}

if(empty($age)) {

echo "<font color='red'>Age field is empty.</font><br/>";

}

if(empty($email)) {

echo "<font color='red'>Email field is empty.</font><br/>";

}

//link to the previous page

echo "<br/><a href='javascript:self.history.back();'>Go Back</a>";

} else {

// if all the fields are filled (not empty)

//insert data to database

$result = mysqli_query($mysqli, "INSERT INTO users(name,age,email)
VALUES('$name','$age','$email')");

//display success message

echo "<font color='green'>Data added successfully.";

echo "<br/><a href='index.php'>View Result</a>";

}

}
```

```
?>
</body>
</html>
```

#### 4. edit.php

```
<?php
// including the database connection file
include_once("config.php");
if(isset($_POST['update']))
{
    $id = mysqli_real_escape_string($mysqli, $_POST['id']);
    $name = mysqli_real_escape_string($mysqli, $_POST['name']);
    $age = mysqli_real_escape_string($mysqli, $_POST['age']);
    $email = mysqli_real_escape_string($mysqli, $_POST['email']);
    // checking empty fields
    if(empty($name) || empty($age) || empty($email)) {
        if(empty($name)) {
            echo "<font color='red'>Name field is empty.</font><br/>";
        }
        if(empty($age)) {
            echo "<font color='red'>Age field is empty.</font><br/>";
        }
        if(empty($email)) {
            echo "<font color='red'>Email field is empty.</font><br/>";
        }
    } else {
        //updating the table
        $result = mysqli_query($mysqli, "UPDATE users SET
```

```
name='$name',age='$age',email='$email' WHERE id=$id");
//redirectig to the display page. In our case, it is index.php
header("Location: index.php");
}
}
?>

<?php
//getting id from url
$id = $_GET['id'];
//selecting data associated with this particular id
$result = mysqli_query($mysqli, "SELECT * FROM users WHERE id=$id");
while($res = mysqli_fetch_array($result)) {
    $name = $res['name'];
    $age = $res['age'];
    $email = $res['email'];
}
?>

<html>
<head>
<title>Edit Data</title>
</head>
<body>
<a href="index.php">Home</a>
<br/><br/>
<form name="form1" method="post" action="edit.php">
<table border="0">
<tr>
<td>Name</td>
```

```

<td><input type="text" name="name" value="<?php echo
$name;?>"></td>

</tr>

<tr>

<td>Age</td>

<td><input type="text" name="age" value="<?php echo
$age;?>"></td>

</tr>

<tr>

<td>Email</td>

<td><input type="text" name="email" value="<?php echo
$email;?>"></td>

</tr>

<tr>

<td><input type="hidden" name="id" value="<?php echo
$_GET['id'];?>"></td>

<td><input type="submit" name="update" value="Update"></td>

</tr>

</table>

</form>

</body>

</html>

```

## 5. delete.php

```

<?php
include("config.php");

$id = $_GET['id'];

$result = mysqli_query($mysqli, "DELETE FROM users WHERE id=$id");

```



```
header("Location:index.php");
```

```
?>
```

## **6. config.php**

```
<?php
```

```
$database = 'localhost';
```

```
$dbName = 'test';
```

```
$dbUser = 'root';
```

```
$dbPass = '';
```

```
$mysqli = mysqli_connect($database, $dbUser, $dbPass, $dbName);
```

```
?>
```

## **7. database.sql**

a. create database emp;

b. use emp;

c. CREATE TABLE `users` (

`id` int(11) NOT NULL auto\_increment,

`name` varchar(100) NOT NULL,

`age` int(3) NOT NULL,

`email` varchar(100) NOT NULL,

PRIMARY KEY (`id`)

);

Name: Vaishnavi Date  
Div. A Roll no: 30  
WT Lab.

### Experiment No: 7

Aim: Add dynamic web application essence with PHP, AJAX & MySQL.

Objective: to understand principles & methodologies of web browser appl<sup>n</sup> development process

Problem statement: Design & develop dynamic web appl<sup>n</sup> using PHP, AJAX & MySQL as back-end for employee data with insert & view operation.

S/W & H/W: Ubuntu 64 / windows XP, XAMPP server, Intel P4 m/c with 4GB RAM

#### Theory:

- AJAX remains asynchronous Javascript & XML. AJAX is another procedure for making better speedier & more intelligent dynamic web appl<sup>n</sup> with assistance XML, HTML, CSS & JS.
- AJAX enables you to send & get information non-currently with reloading page.
- AJAX speaks with server utilizing XMLHttpRequest.
- AJAX communicates with server using XMLHttpRequest object.

Technology: AJAX, PHP & MySQL

Design: For design Purpose HTML & CSS to be used. For this design part contains GUI of web applications

Conclusion: We have studied how to design & develop small web application using PHP, javascript, ajax, Xampp server with apache server & mysql as backend.

## **Program:**

### **1.index.html**

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>PHP Database Example with Ajax </title>
```

```
<script
```

```
src="https://code.jquery.com/jquery-3.2.1.min.js"
```

```
integrity="sha256-hwg4gsxgFZhOsEEamdOYGBf13FyQuiTwlAQgxVSNgt4="
```

```
crossorigin="anonymous">
```

```
</script>
</head>
<body>
<h1> Enter Employee Details </h1>
<form method="post" action="insert.php">
<input type="text" id="name" name="name" placeholder="Enter Name" />
<input type="text" id="age" name="age" placeholder="Enter Age" />
<input type="text" id="city" name="city" placeholder="Enter City" />
<button> Save Data </button>
</form>
<p id="result">
</p>
<p id="result"></p>
<a href="display.php">Display</a>
<!--jquery and ajax code-->
<script>
$("form").submit(function(e){
e.preventDefault();
$.post(
"insert.php",
{
name: $("#name").val(),
age: $("#age").val(),
city: $("#city").val()
},
function(result)
{
if(result == "success")
```

```

{
$("#result").html("Data Inserted Successfully..!");
}
else
{
$("#result").html("Error Occured!");
}
}
);
});
</script>
</script>
</body>
<html>

```

## 2. insert.php

```

<?php
$name = $_POST['name'];
$age = $_POST['age'];
$city = $_POST['city'];
$con = new mysqli('localhost', 'root', "", 'emp');
if($con->connect_error)
{
echo("Error");
}
$stmt = $con->prepare("insert into users(name,age,city) values (?,?,?)");
$stmt->bind_param("sis",$name,$age,$city);
if($stmt->execute())

```

```
{
echo("success");
}
else
{
echo("fail");
}
?>
```

### 3. display.php

```
<?php
include_once("config.php");

// $result = mysql_query("SELECT * FROM users ORDER BY id DESC"); // mysql_query is
// deprecated
$result = mysqli_query($mysqli, "SELECT * FROM users ORDER BY id DESC"); // using
// mysqli_query instead

?>

<html>
<body>
<a href="index.html">Add New Data</a><br/><br/>
<table width='80%' border=0>
<tr bgcolor='#CCCCCC'>
<td>Name</td>
<td>Age</td>
<td>CIty</td>
</tr>

<?php
// while($res = mysql_fetch_array($result)) { // mysql_fetch_array is deprecated, we need
```

```
to use mysqli_fetch_array

while($res = mysqli_fetch_array($result)) {
    echo "<tr>";
    echo "<td>".$res['name']. "</td>";
    echo "<td>".$res['age']. "</td>";
    echo "<td>".$res['city']. "</td>";
    echo "</tr>";
}

?>

</table>

</body>

</html>
```

#### **4. config.php**

```
<?php

$host = 'localhost';

$dbname = 'emp';

$dabUser = 'root';

$dbPass = '';

$mysqli = mysqli_connect($host, $dbUser, $dbPass,
$dbname);

?>
```



Name: Vaishnavi Date

Div: A

Roll no: 30

WT Lab.

Experiment No: 8

Aim: Design & develop any web application using struts framework.

Problem statement: create login page for the web appl<sup>n</sup> using struts framework.

SW & HW: Java 1.7 or higher, Apache tomcat 7 or Higher, struts API's, Eclipse, IDE.

Theory:

- The framework plays a vital role in industries for managable & well designed appl<sup>n</sup> development as well as enterprise appl<sup>n</sup> development.
- The core of Struts is a flexible control layer based on standard technologies like Java Servlets, JavaBeans, Resource bundles & XML as well as previous Jakarta commons package.
- The model view controller architecture.
- It is a way to build appl<sup>n</sup> that promotes complete separation bet<sup>n</sup> business logic & presentation.
- It is not specific to web appl<sup>n</sup> or java or J2EE, but it can be applied to building J2EE web applications.

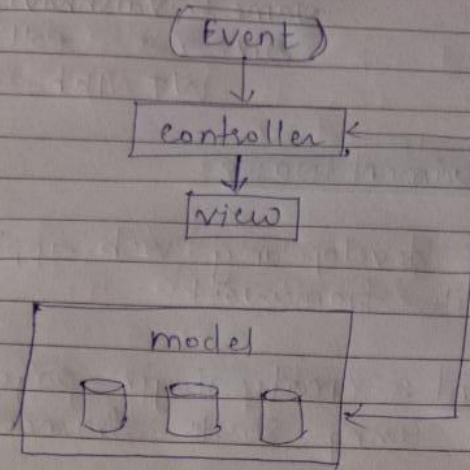


Fig. Basic MVC architecture

What is struts?

struts is a framework that advances the utilization of the model view controller engineering for planning substantial scale applications.

### Steps / Design :

- 1] create directory structure
- 2] create jsp page
- 3] Provide entry of controller in file
- 4] Load Jar files
- 5] Start server & deploy project or create .war file paste it in webpages folder & run.

Conclusion: Hence, we have successfully tested struts framework & tested the results.



Name: Vaishnavi Date  
Div-A Roll No-30.  
WT Lab.

### Experiment - No- 9

Aim :- Design & develop any web appl<sup>n</sup> Using Js.

Problem statement :-

Create an application for bill payment record using Angular Js.

S/W Req :-

Eclipse IDE/ Notepad, modern web browser.

Theory :-

- Angular Js is open source web appl<sup>n</sup> framework. It was initially created in 2009 by miskothevaty and adam abron.
- It presently kept by google. Its most recent adoption is 1.2.21. Angular Js is quillary system for dynamic web appl<sup>s</sup>.
- It gives you a chance to utilize HTML as layout dialect & gives you a chance to stretch out HTML linguistic structure to express your appl<sup>n</sup> parts plainly & compactly.
- Its information official & reliance infusion take out a significant part of code. you as of now need to compose.

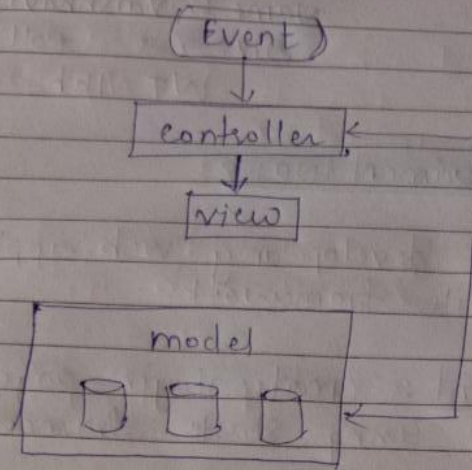


Fig. Basic MVC architecture

What is struts?

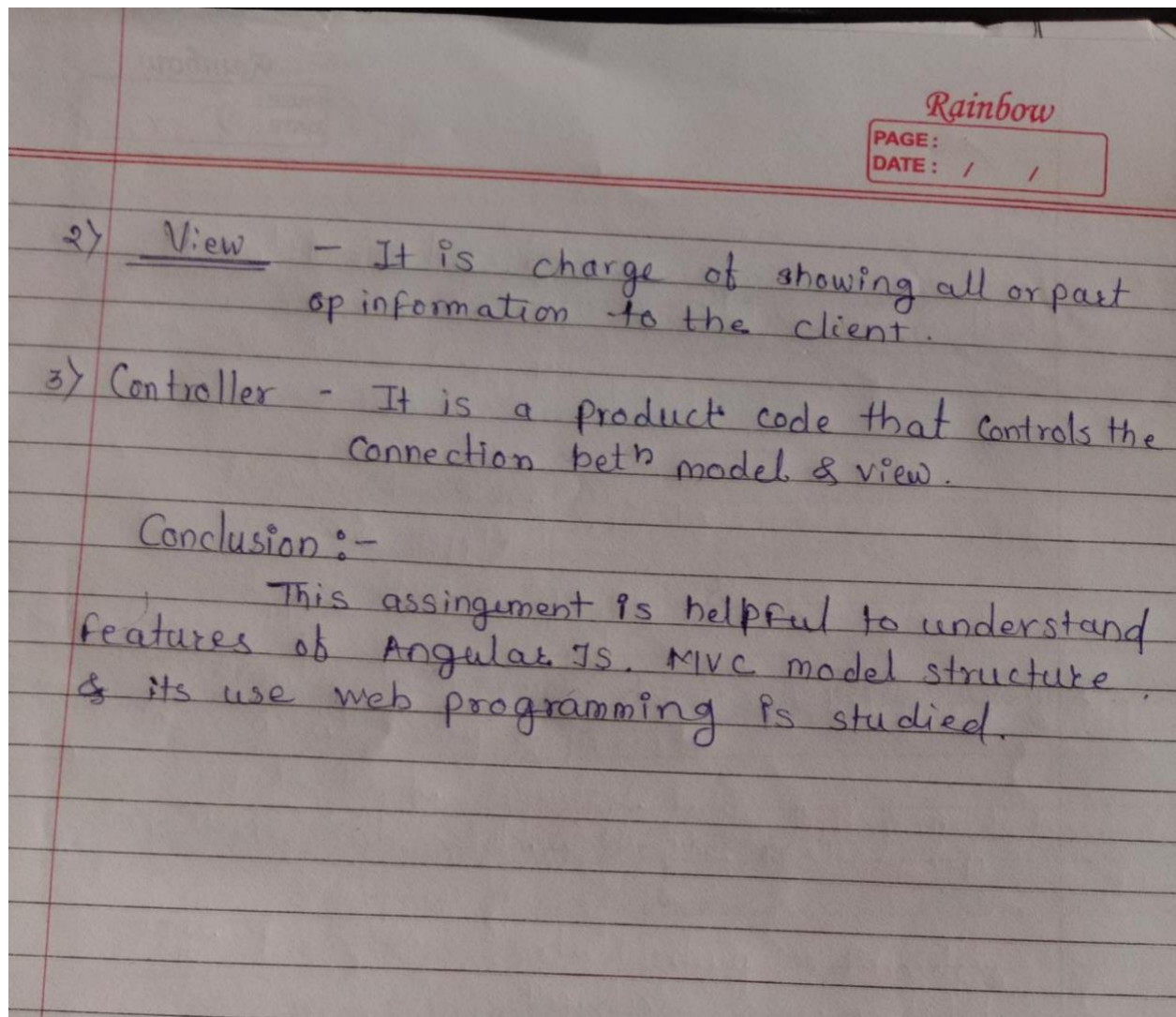
struts is a framework that advances the utilization of the model view controller engineering for planning substantial scale applications.

Steps / Design:

- 1] create directory structure
- 2] create jsp page
- 3] Provide entry of controller in file
- 4] Load Jar files
- 5] Start server & deploy project or create .war file paste it in webpages folder & run.

Conclusion: Hence, we have successfully tested struts framework & tested the results.





### **Program:**

```
<html ng-app="billpayApp">  
<!-- SCRIPTS TO BE ADDED IN HEAD TAG -->  
> <head>  
<title>Bill Payment Record using angular and bootstram  
framework</title>  
<meta http-equiv="content-type" content="text/html; charset=utf-8" />  
<!-- ACCESSING ANGULARJS BY CDN METHOD-->
```

```
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.4/angular.min
.js"></script>

<!-- ACCESSING STYLESHEET FOR DESIGN [OPTIONAL PART CAN BE SKIP]-->

<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.mi
n.css">

<!-- MODEL PART-->

<script>
var model = {
customer: "Student",
items: [{
bill: "Electricity",
status: false
},
{
bill: "Internet(Wi/fi)",
status: false
},
{
bill: "Parking Charges",
status: false
},
{
bill: "Phone",
status: true
},
{
```

```

    bill: "House Tax",
    status: true
  }
]
}

var billpayApp = angular.module("billpayApp", []);
billpayApp.controller("billpayctrl", function($scope)
{
  $scope.billpay = model;
  $scope.dueBills = function() {
    var items = $scope.billpay.items;
    var counter = 0;
    items.forEach((item) => {
      if (!item.status) {
        counter++;
      }
    })
    return counter;
  }
  $scope.redFlag =
  function() {
    return $scope.dueBills() <= 2 ? "label-success" : "label-danger";
  }
  $scope.addBills = function(billName)
  {
    obj = {
      bill: billName,
      status: false
    }
    $scope.billpay.items.push(obj);
  }
}

```

```

} $
scope.removeBills =
function(rmvBills) {
$scope.billpay.items.splice($scope.billpay.items.indexOf(rmvBills), 1);
}
});
</script>
</head>
<!-- HTML BODY PART-->
<body ng-controller="billpayctrl">
<div class="container">
<div class="page-header">
<h1>{{ billpay.customer }}'s Bill's remained to Be Paid -
<span class="lable" ng-class="redFlag()" ng-hide="dueBills()==0">
{{ dueBills() }}
</span>
</h1>
</div>
<h3><center><b>Add extra biller fields if any</center></b></h3>
<div class="panel">
<div class="input-group">
<input class="form-control" ng-model="billName" />
<span class="input-group-btn">
<button class="btn btn-danger" ngclick="
addBills(billName)">+ADD+</button>
</span>
</div>
<table class="table table-striped">

```



```
<thead>

<tr>

<th>Bill Name</th>

<th>Status</th>

<th>Status</th>

<th>Close</th>

</tr>

</thead>

<tbodyng-model="rmvBills">

<trng-repeat="item in billpay.items" ng-model="item">

<td>{{ item.bill }}</td>

<td><input type="checkbox" ng-model="item.status" /></td>

<td>{{ item.status }}</td>

<td>

<button type="button" ng-click="removeBills(item)">&times;</button>

</td>

</tr>

</tbody>

</table>

</div>

</div>

</div>

</body>

</html>
```

Page \_\_\_\_\_  
Name :- Vaishnavi Date \_\_\_\_\_

Div - A Roll No - 30

WT Lab

## Experiment No - 10

Aim :- Web application using EJB.

Problem statement :- Design develop & deploy web application using EJB.

- S/W Req -
1. Ubuntu 64 bit/windows.
  2. JDK 7 or Higher.
  3. EJB 3.0
  4. Eclipse/una.
  5. JBoss appl<sup>n</sup> server.

## Theory.

Java Beans :-

- J2EE appl<sup>n</sup> Container contains the components that can be used by clients for executing the business logic.
- These components are known as enterprise Java Bean (EJB). J2EE platform has components based architecture to provide multitiered, distributed & highly transactional features to enterprise level appl<sup>n</sup>.

Date \_\_\_\_\_  
Page \_\_\_\_\_

It is used for developing very much scalable, re-business enterprise level appl's. to be deployed. application server such as JBoss, WebLogic etc.

EJB 3.0 is being a large shift from EJB & make development of EJB based applications.

Features of EJBs.

client Comm :-

- i) state management.
- ii) Transaction mangement.
- iii) Database connection management.
- iv) User Authentication & Role-based Authoraization
- v) Asynchronous messaging.
- vi) Appl'n Server administration.

Types of Enterprise Java Beans. (EJB).

1. Session Beans.
2. Entity Beans.
3. Message Driven Beans.

Enterprise Java Beans Architecture :-

The EJB architecture is an extension of web architecture. It has an add'l tier.

The clients of an enterprise bean can be a tradition Java appl'n, applet, JSP or Servlet.

CLASSMATE  
Date \_\_\_\_\_  
Page \_\_\_\_\_

### Design / Execution steps :-

1. Design EJB project
2. Start JBoss & deploy it on JBoss Server.
3. Design html & JSP Files with an extension of Html & JSP.
4. Run the appl<sup>n</sup> in browser & get the result

### Conclusion :-

Hence we have created a simple EJB Stateless session bean & Local Java appl<sup>n</sup> client. which will call / invoke the bean to develop for performing addition of two number.