
Web Technology

Lab Manual

ACADEMIC YEAR 2017-18

SEMESTER- IV



1,Kennedy Road, Pune-01

Table of Content

Sr. No	Content	Page No.
1.	About Savitribai Phule Pune University	i
2.	About Program	i
3.	About Institute	ii
4.	About College	ii
5.	Assignment No.1	1-5
6.	Assignment No.2	6-13
7.	Assignment No.3	14-18
8.	Assignment No.4	19-24
9.	Assignment No.5	25-39
10.	Assignment No.6	40-60
11.	Assignment No.7	61-70
12.	Assignment No.8	71-81
13.	Assignment No.9	82-96
14.	Assignment No.10	97-119

About Savitribai Phule Pune University

Savitribai Phule Pune University, one of the premier universities in India, is positioned in the North-western part of Pune city. It occupies an area of about 411 acres. It was established on 10th February, 1949 under the Poona University Act. The university houses 46 academic departments. It is popularly known as the 'Oxford of the East'. It has about 307 recognized research institutes and 612 affiliated colleges offering graduate and under-graduate courses.

The university attracts many foreign students due to its excellent facilities. It offers good accommodation facility. There is a provision of hostel for the students. There is a well-stocked library containing plenty of books regarding various subjects. The university offers different scholarships to the students. The university conducts seminars and conferences for the students.

About Program

(Web Technology Lab - Third Year Engineering)

Nowadays, there are a few companies that, if having a traditional business, do not have a web presence. The Internet environment is not only a simple tool to promote a business, but it also offers opportunities to supply information; it is an efficient platform to communicate with the clients. The integration of Web technologies has an important place into the process of accomplishing companies' objectives to increase the competitiveness degree on the market by building customer loyalty.

The Future of Web Development: Javascript has been ubiquitous on the front-end for a long time and Node has grown into a well developed, enterprise ready technology with a massive amount of support via the npm. New tools such as Ionic(mobile) and Electron(desktop) have made it possible to build applications across all platforms using HTML, CSS and Javascript. The server side applications are PHP, AJAX, Servlet etc.

About Institute

The AISSMS COE, Pune is a co-education Institute established in 1992. The College of Engineering is affiliated to the University of Pune. It conducts AICTE approved courses leading to the degree of Bachelor of Engineering (BE) in eight engineering streams and Master of Engineering (ME) in Seven engineering streams. The AISSMS COE, is located in heart of Pune city. It is in the vicinity of Pune Railway station & Shivaji Nagar..

About Department

The Computer Engineering Department of AISSMS College of Engineering was started in the year 1998. The goal of the Department is to prepare students for successful careers in software and hardware industry that meets the needs of Indian and multinational companies or to excel in higher studies. The Department has experienced and dedicated faculty members with strong commitment to the engineering education and professional ethics.

Web Technology Lab

Assignment List

1. Installation of TOMCAT/ WebSphere/JBoss/GlassFish Server and configuration of it.
2. Write a program to design registration form for students by using HTML and CSS.
3. Write a program to design book catalog by using XML and CSS to display title, author, price and year of the book.
4. Write a program to design registration form for students by using HTML, CSS& Java Script and perform following validations: all fields mandatory, phone number and email address validation.
5. Design and Build Student Login Page using JSP, Servlet and MySQL.
6. Design and develop dynamic web application using PHP and MySQL as a back-end for employee data with insert, delete, view and update operations.
7. Design and develop dynamic web application using PHP, AJAX and MySQL as a back-end for employee data with insert and view operations.
8. Create a login module for the web application using struts framework.
9. Create an application for Bill Payment Record using AngularJS.
10. Design, Develop & Deploy web application using EJB.

Assignment No. : 1

TITLE

TOMCAT SERVER installation

OBJECTIVES

1. Understand how to install tomcat server and set up environment variables

PROBLEM STATEMENT

Installation of TOMCAT Server and Configuration of it.

OUTCOME

Students will be able to,

1. Install TOMCAT Server and Configuration of it .

SOFTWARE & HARDWARE REQUIREMENTS

1. Java 7 or Later
2. Apache Tomcat Server

THEORY-CONCEPT

Web Application:

A web application runs over the Internet. Ex. eBay,Amazon ,Google, facebooketc

A webapp contains five components:

1.HTTP Server: Examples are- Google Web Server , Apache HTTP Server, Apache Tomcat Server, Microsoft Internet Information Server (IIS) etc

2.HTTP Client (Web Browser): Examples are- Internet Explorer, Firefox, Google Chrome, Safari etc.

3.Database:Examples are- MySQL, Apache Derby, MS SQL Server, SQLite, PostgreSQL, Commercial Oracle, IBM DB2, SAP SyBase, MS Access etc

4.Client-Side Programs: It can be written in HTML Form,VBScript, JavaScript, Flash etc.

5.Server-Side Programs: could be written in Java Servlet/JSP, ASP, PHP, Perl, Python, CGI, and others.

A web app is 3-tier (or multi-tier) client-server database application which run over the Internet as shown in the following diagram,

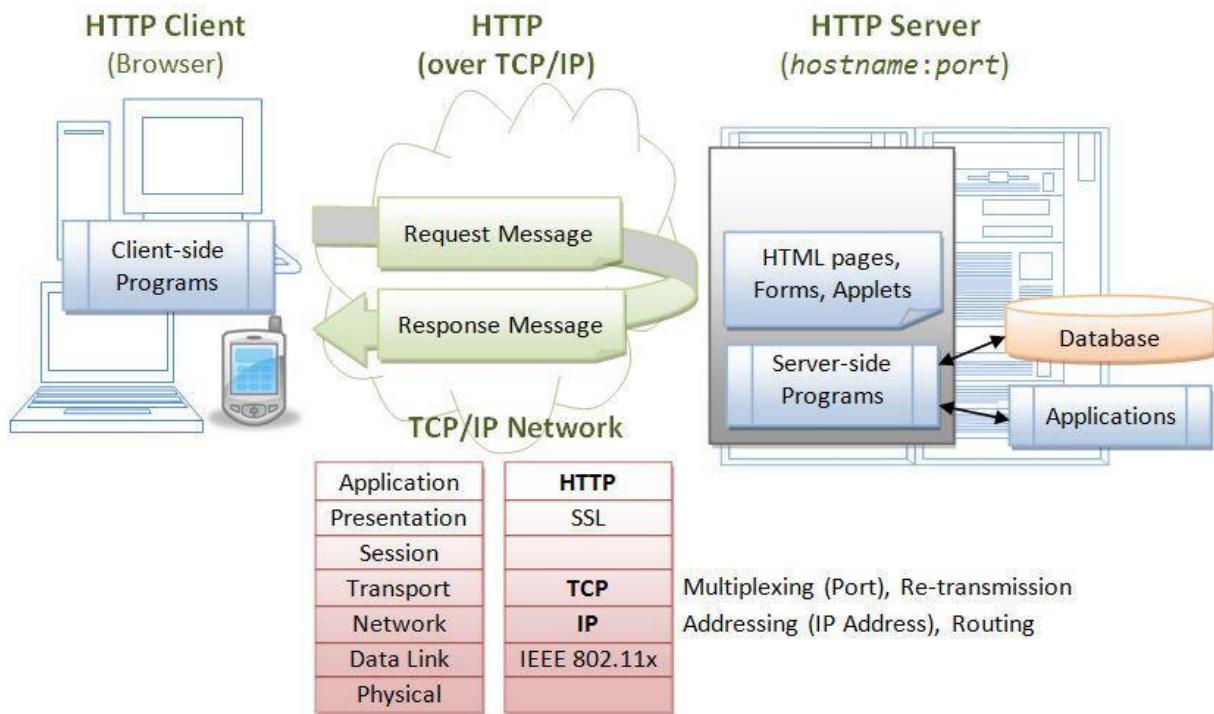


Figure.1: Three-Tier Client-Server Architecture

1. To start a webapp, A user, issues a URL request via a web browser (HTTP client), to HTTP server.
2. The HTTP server returns an HTML form (client-side program), which is loaded into the client's browser.
3. The user fills up the query data inside the form and submits that form.
4. The client-side program sends the query parameters to a server-side program.
5. The server-side program receives the query parameters, queries the database based on these parameters, and returns the query result to the client-side program.
6. The client-side program displays the query result on the browser.

7. The process repeats for the next request.

Apache tomcat

Tomcat is an *open-source* project, under the "Apache Software Foundation" (which also provides the most use, open-source, industrial-strength Apache HTTP Server). The mother site for Tomcat is <http://tomcat.apache.org>. Alternatively, you can find tomcat via the Apache mother site:<http://www.apache.org>.

EXECUTION STEPS

1. Goto <http://tomcat.apache.org>
2. Select tomcat version 8 from left side list under Download option (tomcat 8) or select a link [32-bit/64-bit Windows Service Installer \(pgp, md5, sha1, sha512\)](#)
3. Download and run ‘.exe ‘ file.
4. I preferred, installation of Tomcat at C:\Program Files (x86)\Apache Software Foundation
5. Use the default settings and provide a password that you will remember.
6. Now you will find the Tomcat installed at C:\Program Files (x86)\Apache Software Foundation\Tomcat 8.0

How to Run Tomcat

Following are the steps to check whether Tomcat has been installed successfully or not-

1. Find start program in the Programs Menu. Look under Apache Tomcat and select "Start Tomcat" or "startup.bat".
2. Open any web browser and type in the given URL:
 - o <http://localhost:8080/>

Now, you will see the Tomcat home page, which is provided by the Tomcat Web server running on your computer. To shut down your server and remove the Console window, select "Stop Tomcat" in the same menu of where you selected "Stop Tomcat".

Tomcat's Directories

It contains the following sub-directories:

- bin: It contains the binaries; startup script (startup.bat for Windows and startup.sh for Unix and Mac OS), shutdown script (shutdown.bat for Windows and shutdown.sh for Unix and Mac OS), as well as other binaries and scripts.
- conf: It contains the system-wide configuration files such as server.xml, web.xml, context.xml, and tomcat-users.xml.
- lib: It contains the Tomcat's system-wide JAR files, which can be accessible by all webapps. We can also place external JAR file (such as MySQL JDBC Driver).
- logs: It contains log files of Tomcat. Also, you can check error messages here.
- webapps: It contains the webapps to be deployed. You can also place the WAR (Webapp Archive) file for deployment here.
- work: Tomcat's working directory used by JSP, for JSP-to-Servlet conversion.
- temp: Temporary files are stored in this directory.

How to set a path for Windows

Assuming you have installed Java in C:\Program Files (x86)\Java\jdk 1.7.0\directory –

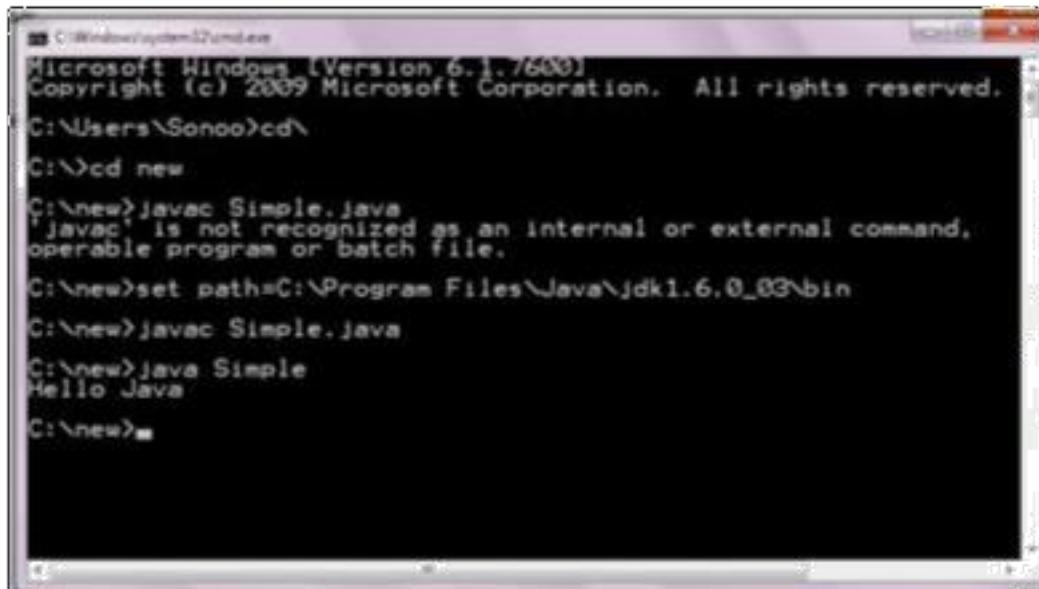
- Right-click on 'My Computer' and select 'Properties'.
- Click the 'Environment variables' button under the 'Advanced' tab.

Now, alter the 'Path' variable so that it also contains the path to the Java executable. Example, path set to C:\Program Files (x86) \Java\jdk1.7.0\bin

OR

through command prompt you can set the path by following command

```
set path=C:\Program Files (x86) \Java\jdk1.7.0\bin
```



A screenshot of a Windows Command Prompt window titled 'Windows\system32\cmd.exe'. The window shows the following command-line session:

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Sonoo>cd\
C:\>cd new
C:\new>javac Simple.java
'javac' is not recognized as an internal or external command,
operable program or batch file.
C:\new>set path=C:\Program Files\Java\jdk1.6.0_03\bin
C:\new>javac Simple.java
C:\new>java Simple
Hello Java
C:\new>
```

CONCLUSION / ANALYSIS

Hence, we have learned how to install and configure tomcat server.

ORAL QUESTIONS

1. What is Tomcat?
2. What is the tomcat default port?
3. What is the servlet container life cycle?
4. What services are provided by Tomcat?
5. Explain directory structure of tomcat.

Assignment No. : 2

TITLE

Title: HTML, CSS, XML

OBJECTIVES

1. Understand about basic concepts of html
2. Understand the basic concepts of XML
3. Understand the basic concepts of CSS

PROBLEM STATEMENT

Write a program to design registration form for students by using HTML and CSS

OUTCOME

Students will be able to,

1. Design static webpage using HTML.
2. Apply CSS to HTML pages.

SOFTWARE & HARDWARE REQUIREMENTS

Software: Notepad, Browser

THEORY-CONCEPT

HTML: HTML is the standard markup language for creating Web pages.

- HTML stands for Hyper Text Markup Language
- HTML describes the structure of Web pages using markup

- HTML elements are the building blocks of HTML pages
- HTML elements are represented by tags
- HTML tags label pieces of content such as "heading", "paragraph", "table", and so on
- Browsers do not display the HTML tags, but use them to render the content of the page
- HTML Versions:

HTML	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML 5	2014

Table.1: HTML Versions

CSS:

CSS stands for Cascading Style Sheet. It is nothing, but design language intended to simplify the process of making web pages presentable. CSS handles the feel and look part of a web page. By using CSS, one can control the color of text, style of fonts, spacing between paragraphs, layout designs.

CSS is easy to learn, easy to understand and it provides powerful control on presentation of an HTML document.

Advantages of CSS:

It saves the time, Pages load faster, Easy maintenance, Superior styles to HTML, Multiple Device Compatibility, Global web standards, Offline Browsing, Platform Independence.

CSS3 Modules:

CSS3 Modules are having old CSS specifications as well as extension features.

- Box Model
- Selectors
- Background
- Border
- Image Values and Replaced Content
- Text Effects

- Animations
- 2D/3D Transformations
- Multiple Column Layout
- User Interface

TECHNOLOGY / TOOL

- The <!DOCTYPE html> declaration defines this document to be HTML5
- The <html> element is the root element of an HTML page
- The <head> element contains meta information about the document
- The <title> element specifies a title for the document
- The <body> element contains the visible page content
- The <h1> element defines a large heading
- The <p> element defines a paragraph
- HTML tags are element names surrounded by angle brackets:

<tagname>content goes here...</tagname>

CSS can be added to HTML elements in 3 ways:

- Inline - by using the style attribute in HTML elements. An inline CSS is used to apply a unique style to a single HTML element.

Ex. <h1 style="color:blue;">This is a Blue Heading</h1>

- Internal - by using a <style> element in the <head> section. An internal CSS is used to define a style for a single HTML page. An internal CSS is defined in the <head> section of an HTML page, within a <style> element.

Example:<style>
body {background-color: powderblue; }
h1 {color: blue; }
p {color: red; }
</style>

- External - by using an external CSS file. An external style sheet is used to define the style for many HTML pages. With an external style sheet, you can change the look of an entire web site, by changing one file! To use an external style sheet, add a link to it in the <head> section of the HTML page.

Example: <link rel="stylesheet" href="styles.css">

- Use the HTML <head> element to store <style> and <link> elements
- Use the CSS **color** property for text colors
- Use the CSS **font-family** property for text fonts
- Use the CSS **font-size** property for text sizes
- Use the CSS **border** property for borders
- Use the CSS **padding** property for space inside the border
- Use the CSS **margin** property for space outside the border

DESIGN / EXECUTION STEPS

Following steps are used to Create and Execute web applications,

1. Write the HTML code in notepad and save with .html extension.
2. Write the CSS code in notepad and save with .css extension.
3. Import CSS file in HTML page.
4. Open HTML page in the browser.

TEST CASES

Manual testing is used to check whether CSS gets applied or not.

CONCLUSION/ANALYSIS

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

PROGRAM CODE & OUTPUT**Registration.html**

```

<html>
    <head>
        <title>regform</title>
        <link href="samp.css" rel="stylesheet"></link>
    </head>
<body>
    <div class="con">
        <form action="display.html" method="get" align="center">
            <div class="con2">
                </img>
            </div>
            <div class="main">
                <table align="center">
                    <tr>
                        <td><b><br>Welcome to Registration
                            Page<br><br></b></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
                                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>
                            </table>
                        </div>
                    </body>
                </html>
            
```

```
</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>
```

samp.css

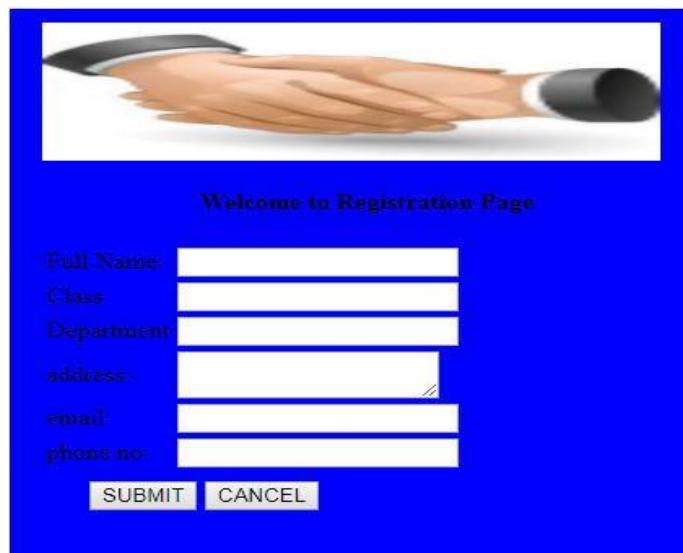
```
.con
{
width:400px;
height:400px;
background-color:blue;
color:blue;
padding:0px 10px 0px 10px;
margin:auto;
}
.main
{
width:400px;
height:80px;
float:left;
padding:60px 10px 10px 10px;
margin: auto;
}
.con1
{
width:200px;
height:30px;
float:left;
margin:auto;
padding:0px 0px 0px 0px;
}
.con2
{
width:150px;
```

```
height:30px;  
float:left;  
margin:10px 10px 10px10px;  
padding:0px 0px 0px0px;  
}
```

Display.css

```
<html>  
<head>  
<title> display</title>  
</head>  
<body>  
<h1 style="color:blue;margin-left:30px;"> You have submitted the form successfully....</h1>  
</body>  
</html>
```

Sample Output:



ORAL QUESTIONS

1. What is the difference between HTML and HTML5?
2. What is the difference between html elements and tags?
3. What is marquee?
4. What is the use of span tag? Give an example?
5. What is the use of 'required' attribute in HTML5?
6. What is External stylesheet? What are the advantages and disadvantages?
7. What is CSS selector?
8. What are the components of CSS style?
9. What are browser safe color?

Assignment No. : 3

TITLE

Title: XML and CSS

PROBLEM STATEMENT

Write a program to design book catalog by using XML and CSS to display title, author, price and year of the book.

OUTCOMES

Students will be able to,

1. Design static webpage using XML.
2. Apply CSS to XML pages.

SOFTWARE & HARDWARE REQUIREMENTS

Software: Notepad, Any Browser

THEORY-CONCEPT

XML stands for Extensible Markup Language. It is nothing but the text-based markup language which is derived from Standard Generalized Markup Language(SGML).

XML tags identify the data and are used to store and organize the data, rather than specifying how to display it like HTML tags, which are used to display the data. XML is not going to replace HTML in the near future, but it introduces new possibilities by adopting many successful features of HTML.

There are three important characteristics of XML that make it useful in a variety of systems and **solutions** –

- XML is extensible – **XML allows you to create your own self-descriptive tags**, or language, that suits your application.
- XML carries the data, does not present it – **XML allows you to store the data irrespective of how it will be presented**.

XML is a public standard – **XML was developed by an organization called the World Wide Web Consortium (W3C)** and is available as an open standard.

TECHNOLOGY/TOOL

The XML document have an XML declaration, but it is optional, and it is written as–

```
<? xml version = "1.0" encoding = "UTF-8"?>
```

Where version is nothing but the version of an XML document and UTF specifies the character-encoding used in the document.

Each XML-element needs to be closed either with start or with end elements as shown below –

```
<element>.....</element>
```

An XML document can have only one root element.

```
<root>
  <x>...</x>
  <y>...</y>
</root>
```

XML Attributes:

Using a name/value pair, an attribute specifies a single property for an element. An XML-element can have one or more attributes. For example –

```
<a href = "http://www.google.com/">XMLTutorial</a>
```

Here href is the attribute name and http://www.google.com/ is attribute value.

DESIGN/EXECUTION STEPS

Following steps are used to Create and Execute web applications,

1. Write the XML code in notepad and save with .xml extension.
2. Write the CSS code in notepad and save with .css extension.

3. Import CSS file in XML page.
4. Open XML page in the browser.

TEST CASES

Manual testing is used to check whether CSS gets applied or not.

CONCLUSION/ANALYSIS

Hence, we have designed static web pages using XML and CSS

PROGRAM CODE & OUTPUT

Book.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="book_catalog.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>
```

book_catalog.css

```
BOOK {
    Display: block;
```

```
margin-left:0;  
magin-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;  
}
```

Output:

Database Management System
Korth

500
1985

Computer Network
Tenenbaum

600
1985

Software Engineering and project Management
Roger Pressman

600
1985

ORAL QUESTIONS

1. Explain difference between HTML and XML?
2. What is XML DOM?
3. Explain difference between CDATA and PCDATA?
4. What is mean by simple element and complex element?
5. What is XPATH?
6. Explain XSL and XSLT?

Assignment No. - 4

TITLE

HTML, Java Script

OBJECTIVES

1. Understand about basic concepts of JavaScript.
2. Use JavaScript for validation of data.

PROBLEM STATEMENT

Write a program to design registration form for students by using HTML, CSS& Java Script and perform following validations: all fields mandatory, phone number and email address validation.

OUTCOMES

Students will be able to,

1. Design static webpage using HTML.
2. Apply JavaScript to HTML pages for validation of data.

SOFTWARE & HARDWARE REQUIREMENTS

Software's: Notepad, Any Browser

THEORY-CONCEPT

JavaScript is a programming language of HTML as well web. It is preferred for creating network-centric applications. It is integrated and complimentary with Java. As JavaScript is integrated with HTML it is very easy to implement. It is open as well as cross-platform.

Advantages:

The advantages **of using JavaScript are –**

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

Validation:

When client enters the all necessary data and press the submit button form validation is done at server side If data entered by a client is incorrect or missing, the server needs to send all data back to the client and request for resubmission of form with correct information. This is really a lengthy process which puts a lot of load(burden) on the server.

So, JavaScript provides a way to validate form's data on the client's side itself before sending it to the web server. Form validation performs two functions-

- Basic Validation – First of all the form must be checked to make sure all the mandatory fields are filled in. It would require just a loop through each field in the form and check for the data.
- Data Format Validation – **Secondly, the data that is entered must be checked for correct format and its value.** The code must include appropriate logic to test correctness of data.

TECHNOLOGY/TOOL

JavaScript can be implemented using JavaScript statements that are placed within the <script>.

You can place the <script> tags, containing your JavaScript, anywhere within your web page, but it is normally recommended that you should keep it within the <head> tags.

The script tag takes two important attributes:

- **Language – This attribute specifies what scripting language you are using.** Typically, its value will be JavaScript. Although recent versions of HTML (and XHTML, its successor) have phased out the use of this attribute.
- **Type – This attribute is what is now recommended to indicate the scripting language in use and its value should be set to "text/javascript".**

DESIGN/EXECUTION STEPS

Following steps are used to Create and Execute web applications,

1. Write an HTML code in notepad and save with .html extension.
2. Write the function for validation of email id and phone no and enclosed this function in script.
3. Call this function on ‘onClick’ event of submit button.
4. Open HTML page in the browser.

TEST CASES

Manual testing is used to check following validations

- All the fields like Full name, Class, Department, Address, Phone number and email id are mandatory fields.
- Phone no. should be numbers only.
- Email id should be in proper format like abc@abc.com

CONCLUSION/ANALYSIS

Hence, we applied validate the data using JavaScript.

PROGRAM CODE & OUTPUT

Registration.html

```
<html>
    <head>
        <title>regform</title>
        <link href="samp.css" rel="stylesheet"></link>

    <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 name="myForm" action="display.xml" method="get" align="center">
<div class="con2">
<imgsrc="D:/handshake.jpg" width=380px
height=100px></img>
</div>
<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>
</div>
</form>
</div>
</body>
</html>

```

Output:

The screenshots illustrate the validation process for a registration form. The first screenshot shows a required field for 'address' is empty. The second shows an invalid email ID ('aaa.com'). The third shows an invalid phone number ('fgf'). Each screenshot includes an 'OK' button at the bottom right.

Field	Value Entered	Error Message
address		Please fill out this field.
email ID	aaa@gmail.com	Please enter correct email ID
phone no.	8888855555	Please enter numeric only.

ORAL QUESTIONS

1. Name some Java Script features.
2. How to define anonymous function?
3. What is callback?
4. What is the difference between undefined and not-defined in JavaScript?
5. What is ‘closure’ in JavaScript?
6. What are JavaScript data types?
7. What are all the types of Pop up boxes available in JavaScript?

Assignment No. - 05

TITLE

JSP, Servlet and MySQL(Backend)

OBJECTIVES

1. Understand about basic concepts of html, CSS
2. Understand the basic functionalities of JSP
3. Having the knowledge of SQL query to create the database

PROBLEM STATEMENTS

1. Design and Build Student Login Page using JSP, Servlet and MySQL.
2. Design and Build Employee Login Page using JSP, Servlet and MySQL.

OUTCOMES

Students will be able to,

1. Develop a dynamic webpage using JSP, HTML and Servlet.
2. Write a server side java application called Servlet to catch the data sent from client, process it and store it on database (MySQL).
3. Write a server side java application called JSP to catch form data sent from client and store it on database (MySQL).

SOFTWARE NEEDED

1. Any Operating System
2. JDK 7 or later
3. Editors; Netbeans/Eclipse
4. Web browser
5. Tomcat 7 or later

THEORY - CONCEPT

Java Server Pages (JSP): It is a server side programming technology that is used to create dynamic web-based applications. JSP have right to use the complete Java APIs, including the JDBC API to access the databases.

It is a technology that helps software developers to create dynamic web pages based on HTML, XML and other document types. It was released in 1999 by Sun Microsystems. It is just like a PHP and ASP, but it uses the Java programming language.

A JSP element is a type of java servlet that is designed to accomplish the role of a user interface for a java web application. Web developers write JSPs as text files that combine HTML or XHTML code, XML elements, and rooted JSP actions and commands.

Using JSP, you can collect input from users through webpage forms, current records from a database or another source and create web pages dynamically.

JSP tags can be used for different purposes, such as retrieving information from a database or registering user preferences, accessing JavaBeans components, passing control between pages, and sharing information between requests, pages etc.

Why we need JSP?

JSP is used for the design of dynamic web page and servlet is used to code the logic that is present i.e. in the MVC (Model-View-Controller) architecture, the servlet is the controller and the JSP is the view.

Architecture of JSP

1. The request / response part of a JSP is defined in below architecture
2. The client initiated request for a JSP file using browser
3. Webs server (i.e, JSP Engine) invokes the JSP file and interpret the JSP file produce a java code. The created java code will be a Servlet.
4. Once Servlet is created, JSP engine compiles the servlet. Compilation errors will be detected in this phase.
5. Now servlet class is loaded by the container and executes it.
6. Engine sends the response back to the client.

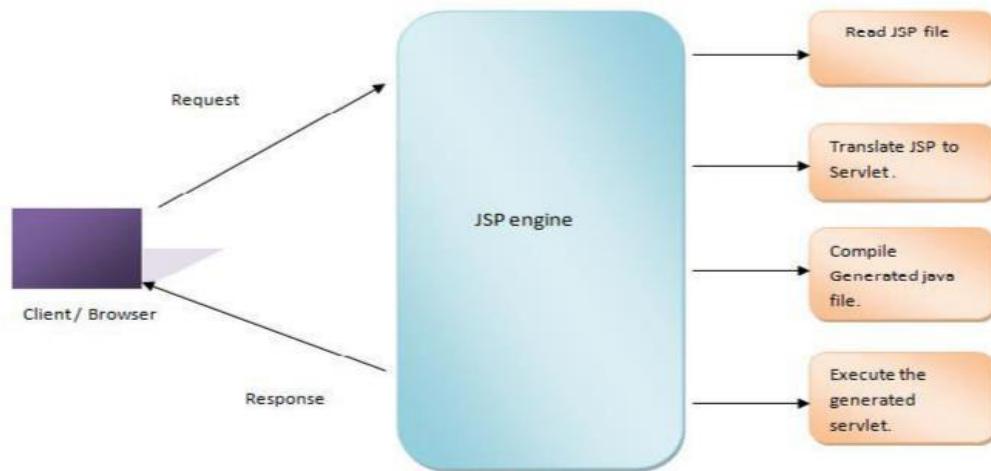


Figure.2: Architecture of JSP

Syntax of JSP:

JSP [declarations](#) is used to declare variables and methods as shown below,

```
<%      text      %>
```

Following is the simple and first example for JSP:

```
//Hello.jsp
<html>
  <head>
<title> JSP File</title>
  </head>

<body>
  <%
    out.println("Welcome to JSP Class");
  %>
</body>
</html>
```

Output:

Welcome to JSP Class

Servlet:

A Servlet is a server side program and written in Java. Servlet is a web component that is deployed on the server for creating the dynamic web pages. A Java servlet is a Java program that extends the capabilities of a server. Although servlets can respond to any types of requests, they most commonly execute applications hosted on Web servers.

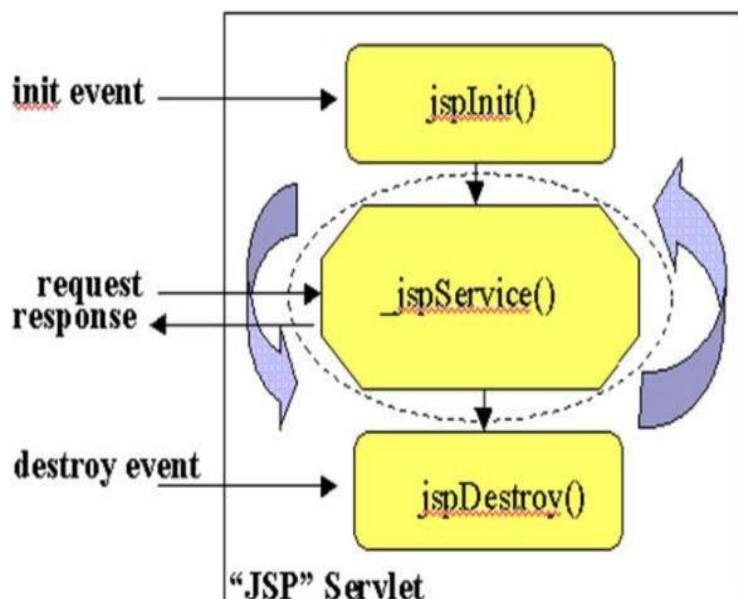


Figure.3: Functions of Servlet

TECHNOLOGY/TOOL IN BRIEF

1. JSP and Servlets
2. IDE: NetBeans 7.0 or Later
3. Databases: MySQL

NetBeans: NetBeans is an IDE, used for quickly and easily developing java desktop, mobile, and web applications, as well as HTML5 applications with HTML, JavaScript, and CSS. Also provides a huge set of tools for PHP and C/C++ developers. It is free and open source tool and has a great community of users and developers around the world.

MySQL: MySQL is a freely available open source Relational Database Management System (RDBMS). It uses the Structured Query Language (SQL).

SQL is the most popular language for adding, accessing and managing data in a database. It is most noted for its quick processing, proven reliability, ease and flexibility of use. MySQL is an essential part of almost every open source PHP application. Good examples for PHP & MySQL-based scripts are WordPress, Joomla, Magento and Drupal.

DESIGN / EXECUTION STEPS

Following steps are used to Create and Execute web applications,

1. Design html and jsp files with an extension of.html and .jsp
2. Write database connection page using servlet
3. Set MySQL username, password and database name in database connection page
4. Start the Tomcat Server with port number
5. Open the browser and type localhost:8084

TEST CASES

Manual testing is used to validate the fields like username, password, mobile number and email id's of the users entered by user with the database.

CONCLUSION / ANALYSIS

Hence, we have performed the dynamic web application using JSP, Servlet and MySQL.

PROGRAM CODE & OUTPUT

Following pages required to run this application:

1. index.jsp
2. SignUp.jsp
3. User.jsp
4. LoginSuccess.jsp
5. LoginFailure.jsp
6. LoginServlet.jsp
7. Login.jsp

```
// index.jsp
<%-- 
   Document : Login
   Created on : Dec 31, 2017, 2:07:34 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> 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>
              </td>
            </tr>
          </table>
        </td>
      </tr>
    </table>
  </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="LoginServelt.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>&ampnbsp&ampnbsp&ampnbsp&ampnbsp <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; <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; <input type="password" name="passs" class="b"></td>
</tr>
<tr>
<td><font face="Times New Roman" size="+1"><strong>Mobile</strong></font></td>
<td>&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; <input type="text" name="eidd" class="b"></td>
</tr>
<tr>
<td><font face="Times New Roman" size="+1"><strong>Date</strong></font></td>
<td>&nbsp;&nbsp;&nbsp;&nbsp; <input type="text" name="date" class="b" value="<%="strDateNew1%">"></td>
</tr>
<tr>
<td></td>
<td>&nbsp;&nbsp;&nbsp;&nbsp; <input type="submit" name="s" value="submit" class="b1" >
    &nbsp;&nbsp;&nbsp; <input type="reset" name="r" value="clear" class="b1"></td>
</tr>
</table>
</form></td>
</tr></table>
<center> <table><tr><td> <a href="index.jsp"><font color="#000000" size="2"><b>Login Here</b></font></a>
</td>
</tr></table></center>

</body>
</html>

```

//LoginSuccess.jsp

```

<%-->
Document : LoginSuccess
Created on : Dec 31, 2017, 2:40:04 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> 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>
              </tr>
            </table>
          </form>
        </td>
      </tr>
    </table>
  </body>
</html>
```

```
<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("mobb");
String d = request.getParameter("eidd");
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())
{
    id=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());
}
%>

```

//Database Name: loginpage

//Table Name: student

Table Data								
sid	sname	suserid	spass	mobile	email	date		
13	Naresh	6B	naresh	6B	na...	6B	8793762366	10B
14	Naresh	6B	naresh	6B	na...	6B	8793762366	10B
15	Anand	5B	anand	5B	anand	5B	9988776655	10B
16	veena	5B	veena	5B	veena	5B	7385910048	10B
17	vani	4B	vani	4B	vani	4B	7385910048	10B
18	abc	3B	abc	3B	abc	3B	7385910048	10B
*	(NULL)	OK	(NULL)	OK	(N...)	OK	(NULL)	OK

Output:

Step 1: Run index.jsp page

The screenshot shows a web browser window with the title bar "Login". The address bar displays "localhost:8084/Assignment3/index.jsp". The main content area is titled "User Login" in green. It contains a form with three fields: "User Id" (text input), "Password" (text input), and a "Login" button. Below the form is a link labeled "New User?".

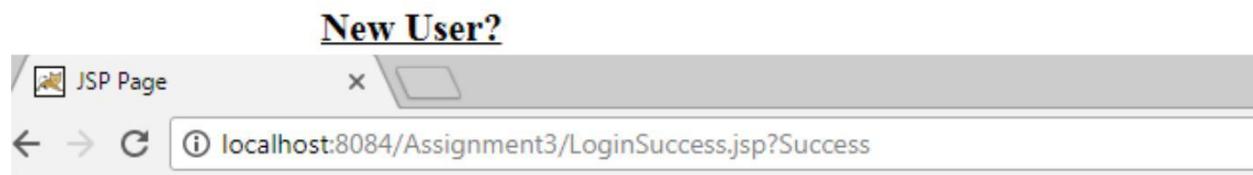
Step 2: If user is already registered then use own credentials else click New User for registration

The screenshot shows a web browser window with the address bar displaying "localhost:8084/Assignment3/SignUp.jsp". The main content area is titled "User Registration" in red. It contains a form with six fields: "Name" (text input), "User ID" (text input), "Password" (text input), "Mobile" (text input), "Email ID" (text input), and "Date" (text input with value "31/12/2017"). At the bottom are two buttons: "submit" and "clear".

Step 3: Once registered click Login directly

User Login

User Id	anand
Password
<input type="button" value="Login"/>	



User Logged In Successfully!...

Step 4: If user gives wrong username and password, it will show error message



User Id	<input type="text"/>
Password	<input type="password"/>
<input type="button" value="Login"/>	

ORAL QUESTIONS

1. What is JSP?
2. What is Servlet?
3. What is the purpose of MySQL?
4. What is database?
5. What is the syntax of JSP?
6. How do we connect JSP file to database?

Assignments no: 06

TITLE

Add dynamic web application essence using PHP, HTML and MySQL.

OBJECTIVES

To understand the principles and methodologies of PHP web based applications development process,

PROBLEM STATEMENT

Design and develop dynamic web application using PHP and MySQL as a back-end for employee data with insert, delete, view and update operations.

OUTCOMES

Students should be able to,

1. Develop web based application using suitable client side and server side web technologies.
2. Develop solution to complex problems using appropriate method, technologies, frameworks, web services and content management.

SOFTWARE & HARDWARE REQUIREMENTS

Software (Minimum Requirement):

1. Ubuntu 64 bit / Windows XP.
2. XAMPP Server

Hardware (Minimum Requirement):

Intel p4 Machine with 1GB RAM and 32GB HDD.

THEORY-CONCEPT

1. PHP:

The PHP Hypertext Preprocessor (PHP) began as a little open source venture that advanced as an ever increasing number of individuals discovered how valuable it was. Rasmus Lerdorf released the principal form of PHP route in 1994. PHP is a recursive acronym for "PHP: Hypertext Preprocessor".

PHP is a server side scripting dialect that is installed in HTML. It is utilized to oversee dynamic substance, databases, session following, even form whole internet business locales. It is incorporated with various prevalent databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.

PHP is pleasingly zippy in its execution, particularly when gathered as an Apache module on the Unix side. The MySQL server, once began, executes even extremely complex questions with colossal outcome sets in record-setting time.

PHP bolsters a substantial number of real conventions, for example, POP3, IMAP, and LDAP. PHP4 included help for Java and conveyed question designs (COM and CORBA), making n-level improvement a plausibility out of the blue. PHP is excusing: PHP dialect tries to be as pardoning as would be prudent. PHP Syntax is C-Like.

PHP performs framework capacities, i.e. from documents on a framework it can make, open, read, compose, and close them. PHP can deal with frames, i.e. accumulate information from records, spare information to a document; through email you can send information, return information to the client.

You include, erase, adjust components inside your database through PHP. Access treats factors and set treats. Utilizing PHP, you can confine clients to get to a few pages of your site. It can encode information.

Example:

"Hello World" Script in PHP

To get a feel for PHP, first start with simple PHP scripts. Since "Hello, World!" is an essential example, first we will create a friendly little "Hello, World!" script.

As mentioned earlier, PHP is embedded in HTML. That means that in amongst your normal HTML (or XHTML if you're cutting-edge) you'll have PHP statements like this –

```
<html>

<head>

<title> Hello World</title>

</head>

<body>

<?php echo ("Hello Php"); ?>

</body>

</html>
```

To create and run PHP Web pages three fundamental parts should be introduced on your PC framework.

Web Server – PHP will work with for all intents and purposes all Web Server programming, including Microsoft's Internet Information Server (IIS) however then regularly utilized is unreservedly accessible Apache Server. Download Apache **for nothing here** – <https://httpd.apache.org/download.cgi>

Database – PHP will work with for all intents and purposes all database programming, including Oracle and Sybase yet most regularly utilized is uninhibitedly accessible MySQL database. Download MySQL **for nothing here** – <https://www.mysql.com/downloads/>

PHP Parser – keeping in mind the end goal to process PHP content directions a parser must be introduced to create HTML yield that can be sent to the Web Browser. This instructional exercise will manage you how to introduce PHP parser on your PC.

2. MySQL:

MySQL is the most famous Open Source Relational SQL Database Management System. MySQL is outstanding amongst other RDBMS being utilized for creating different online programming applications. MySQL is created, advertised and upheld by MySQL AB, which is a Swedish organization. This instructional exercise will give you a fast begin to MySQL and make you OK with MySQL programming.

What is a Database?

A database is a different application that stores a gathering of information. Every database has at least one unmistakable APIs for making, getting to, overseeing, seeking and recreating the information it holds.

Different sorts of information stores can likewise be utilized, for example, records on the document framework or vast hash tables in memory yet information getting and composing would not be so quick and simple with those kind of frameworks.

These days, we utilize social database administration frameworks (RDBMS) to store and oversee tremendous volume of information. This is called social database since every one of the information is put away into various tables and relations are set up utilizing essential keys or different keys known as Foreign Keys.

A Relational DataBase Management System (RDBMS) is a product that:

- Empowers you to execute a database with tables, segments and records.
- Ensures the Referential Integrity between columns of different tables.
- Updates the lists naturally.
- Deciphers a SQL inquiry and consolidates data from different tables.

RDBMS Terminology

Before we continue to clarify the MySQL database framework, let us modify a couple of definitions identified with the database.

- Database: A database is a gathering of tables, with related information.
- Table: A table is a grid with information. A table in a database resembles a basic spreadsheet.
- Column: One section (information component) contains information of one and a similar kind, for instance the segment postcode.
- Row: A line (= tuple, passage or record) is a gathering of related information, for instance the information of one membership.
- Redundancy: Storing information twice, needlessly to make the framework quicker.
- Essential Key: An essential key is exceptional. A key esteem can not happen twice in one table. With a key, you can just discover one column.
- Outside Key: A remote key is the connecting pin between two tables.
- Compound Key: A compound key (composite key) is a key that comprises of numerous sections, since one segment isn't adequately exceptional.
- Index: A file in a database looks like a file at the back of a book.
- Referential Integrity: Referential Integrity ensures that an outside key esteem dependably indicates a current column.

MySQL Database

MySQL is a quick, simple to-utilize RDBMS being utilized for some little and huge organizations. MySQL is produced, showcased and upheld by MySQL AB, which is a Swedish organization. MySQL is winding up so famous as a result of numerous great reasons:

- MySQL is discharged under an open-source permit. So you don't have anything to pay to utilize it.
- MySQL is a capable program in its own particular right. It handles a huge subset of the usefulness of the most costly and intense database bundles.
- MySQL utilizes a standard type of the outstanding SQL information dialect.
- MySQL takes a shot at many working frameworks and with numerous dialects including PHP, PERL, C, C++, JAVA, and so forth.
- MySQL works rapidly and functions admirably even with extensive informational indexes.
- MySQL is amicable to PHP, the most refreshing dialect for web advancement.
- MySQL underpins huge databases, up to 50 million lines or more in a table. The default document measure restrain for a table is 4GB, yet you can expand this (if your working framework can deal with it) to a hypothetical utmost of 8 million terabytes (TB).

- MySQL is adaptable. The open-source GPL permit enables developers to alter the MySQL programming to fit their own particular surroundings.

TECHNOLOGY/TOOL

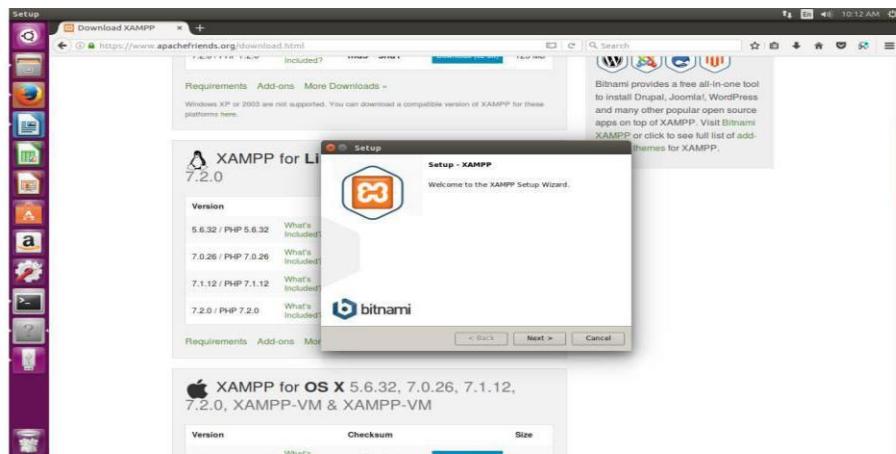
1. Technology is to be used is PHP (PHP Hypertext Preprocessor) and tool XAMPP server is to be used to execute PHP web application.
2. XAMPP server embeds the PHP, MySQL and phpmyadmin, these three tools must be required to run php web application.

DESIGN/EXECUTION STEPS

For the design purpose html and CSS is to be used. For this design part contains the GUI of web applications, how its looks like? When users going to use the web application.

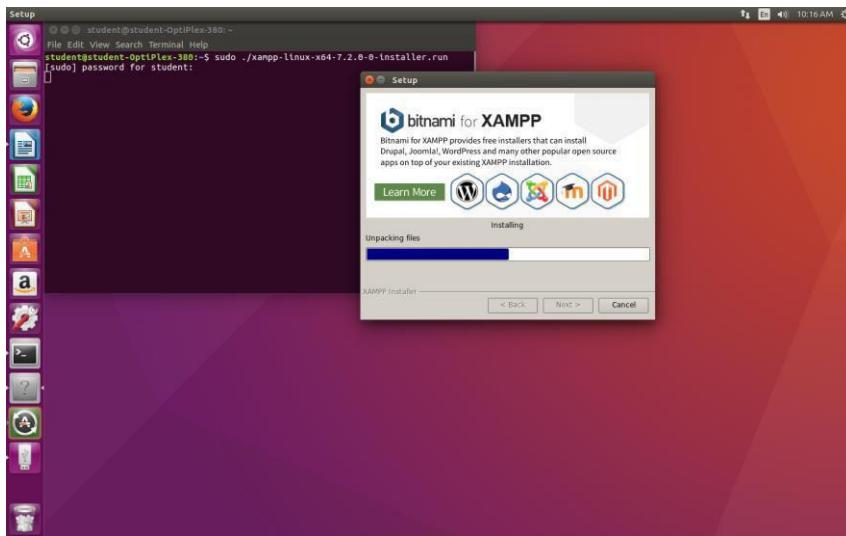
Steps to install XAMPP and configure the PHP, MYSQL server.

1. Download the XAMPP using following link (download latest version as per your Operating system Windows/ Linux). Here we are using Linux Ubuntu Systems. Copy and paste downloaded XAMPP into home location. <https://www.apachefriends.org/download.html>
2. Install XAMPP with following command in terminal(copy paste the command).
`sudo ./xampp-linux-x64-7.2.0-0-installer.run`
3. After above command following installation window will appear.
4. ove command following installation window will appear.

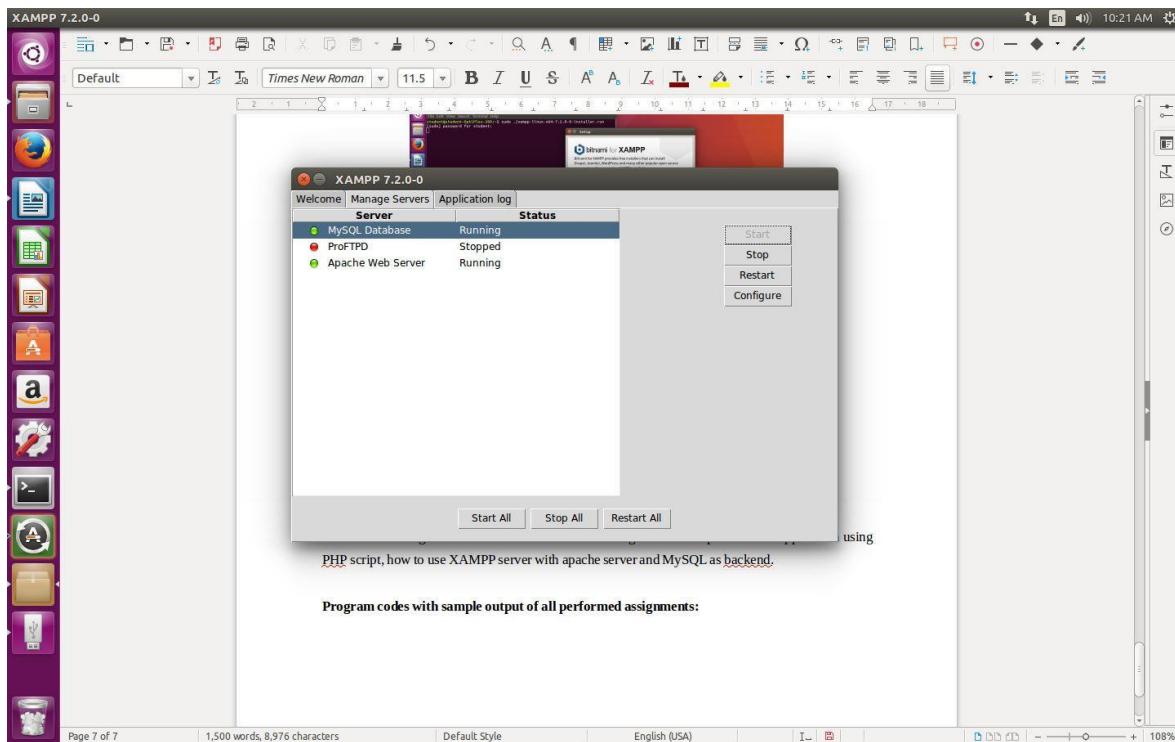


Click on next until all installation windows are completed.

During installation XAMPP look like as follows:

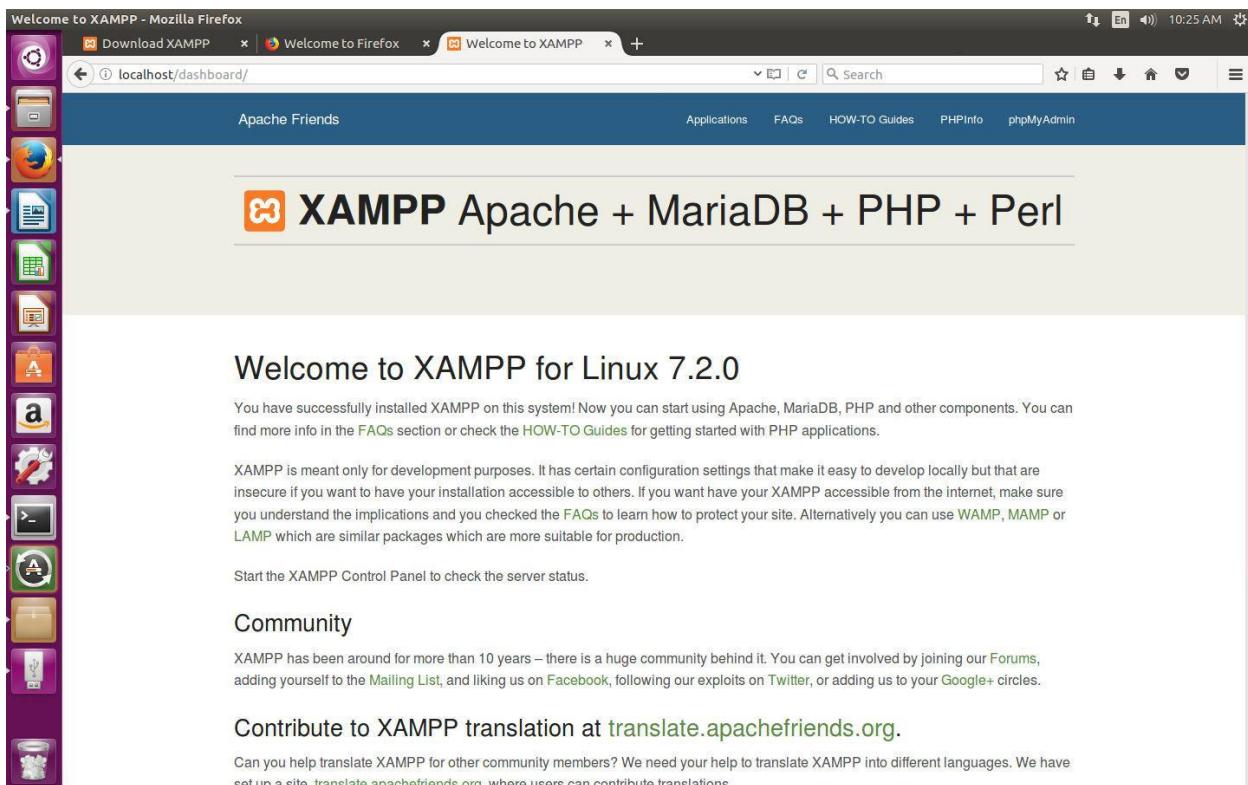


after successful completion of installation. It will ask to run XAMPP and here say to run. After run the MySQL and Apache web server should be in running state. See below image.



As per above image you are ready with Apache Web Server and MySQL Database.

4. Now Open browser and type “localhost” in URL with quotes. Following page should be displayed.



5. the Default directory is with following path

To become root just open the terminal and type the following command `sudo -i`

after you become the root just type the following command to enter the root directory

```
cd /
```

“opt” is the directory in which XAMPP is installed by default.

6. As you have successfully installed and started XAMPP now just navigate to htdocs by typing the following command into terminal:

Note: to edit or create any file in htdocs you need to be root : `sudo -i`

7. To navigate to root folder

```
cd /
```

8. To navigate to htdocs

cd opt/lampp/htdocs

9. To create a file hello.php

gedit hello.php

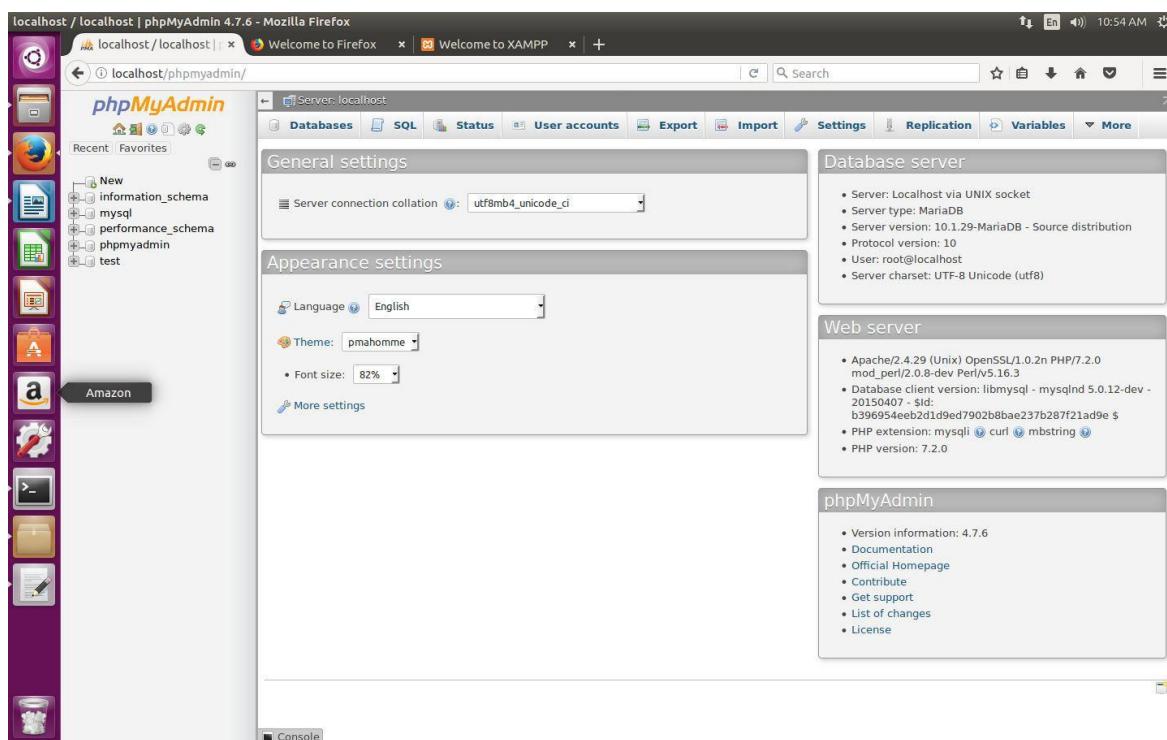
Code for Sample Hello Script

```
<?php  
echo("Hello, Welcome to First PHP Application");  
?>
```

10. Now go to your browser then type localhost/filename.php

11. Open phpmyadmin to create MySQL database

type in browser URL “localhost/phpmyadmin” you will see following window.



Here you can create the database and table inside database as per your need. Here we created “test” database with “users” table contain four columns.

To start XAMPP Manually simply call this command on root:

sudo /opt/lampp/lampp start

To stop XAMPP manually simply call this command on root:

sudo /opt/lampp/lampp stop

CONCLUSION/ANALYSIS

In this assignment, we have studied how to design and develop small web application using PHP script, XAMPP server with apache server and MySQL as backend.

PROGRAM CODE WITH OUTPUT

Here we have created “emp” directory in htdocs directory contain following files

1. index.php
2. add.html
3. add.php
4. edit.php
5. config.php
6. database.sql(optional)

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
```

```
?>

<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>

        </table>

    </form>

</body>

</html>
```

```
</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'])) {

    $name = mysqli_real_escape_string($mysql, $_POST['name']);

    $age = mysqli_real_escape_string($mysql, $_POST['age']);

    $email = mysqli_real_escape_string($mysql, $_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.</font>";
```

```
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';
```

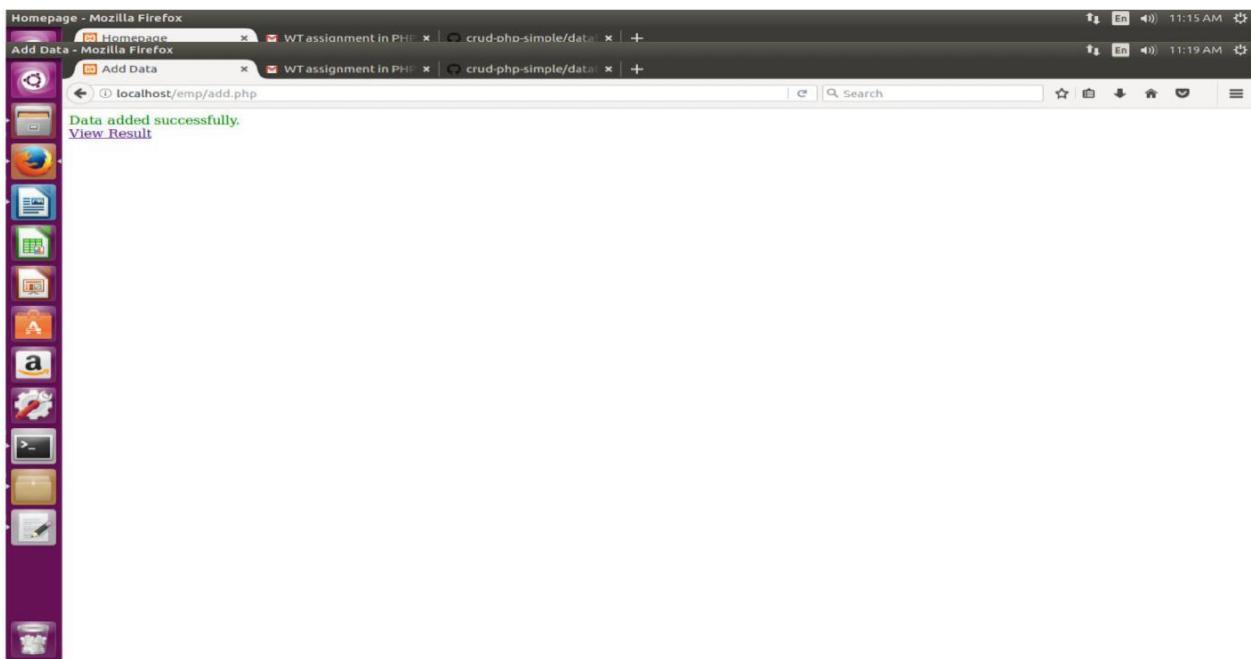
```
$dbName = 'test';  
$dbUser = 'root';  
$dbPass = " ";  
$mysqli = mysqli_connect($database, $dbUser, $dbPass, $dbName);  
  
?>
```

7. database.sql

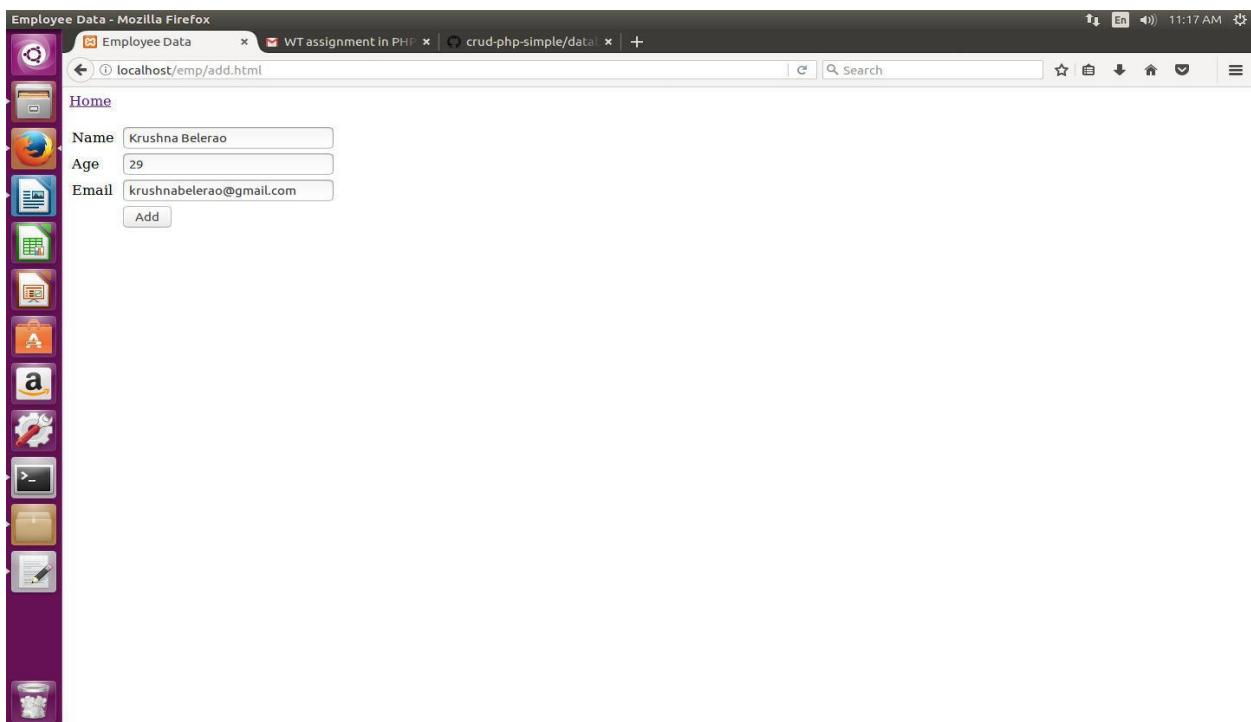
- a. create database emp;
- b. use emp;
- c. CREATE TABLE `users` (
 `id` int(11) NOT NULL auto_increment,
 `name` varchar(100) NOT NULL,
 `age` int(3) NOT NULL,
 `email` varchar(100) NOT NULL,
 PRIMARY KEY (`id`)
);

OUTPUT

1. First Window

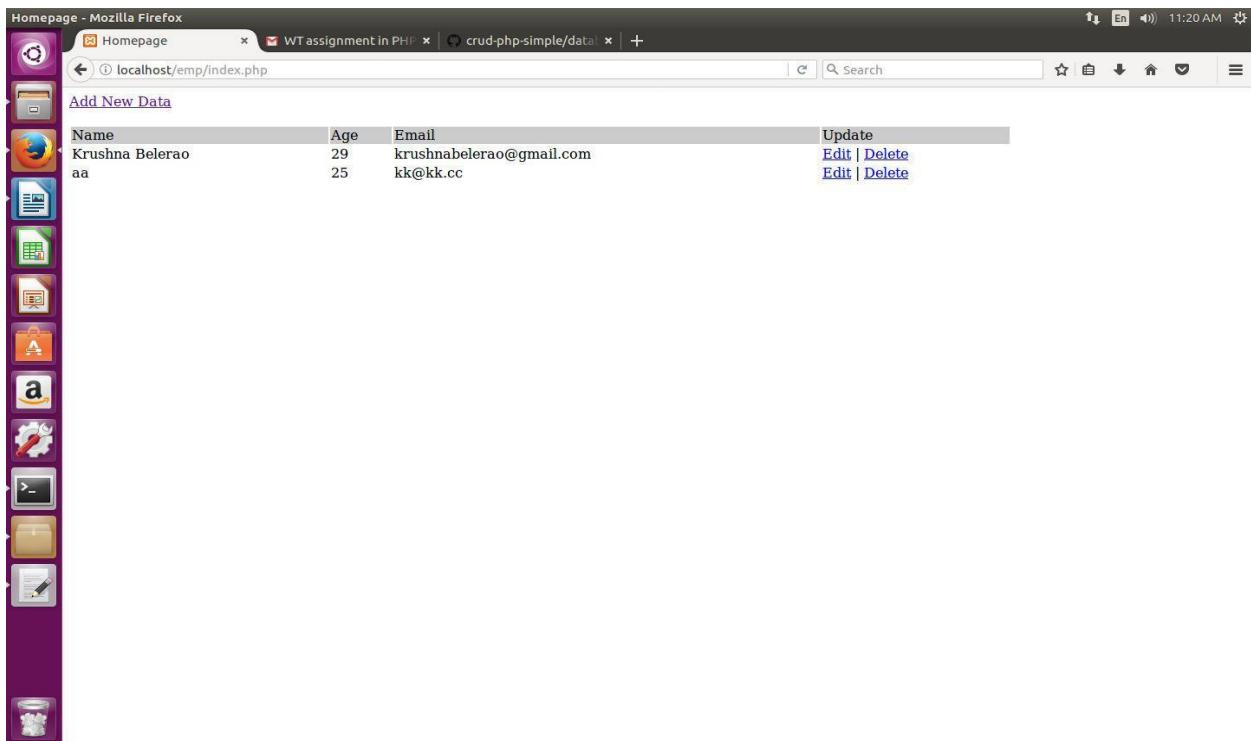


2. Second Window click on “Add New Data” see top left corner link



3. click “add” to add data in database and it will display below window

4. Click on “View Result ” below image will be displayed with all details in database.



The screenshot shows a Mozilla Firefox browser window with three tabs open: "Homepage", "WT Assignment in PHP", and "crud-php-simple/data". The main content area displays a table with two rows of employee data:

Name	Age	Email	Update
Krushna Belerao	29	krushnabelerao@gmail.com	Edit Delete
aa	25	kk@kk.cc	Edit Delete

5. for delete and edit, use Update link from the right side of above web page.

ORAL QUESTIONS

1. What is the use of "echo" in php?
2. How to include a file to a php page?
3. Differences between GET and POST methods ?
4. What is the use of 'print' in php?
5. What is the difference between Session and Cookie?
6. What are the different errors in PHP?
7. How to print current date and time?
8. What is the difference between sql and Mysql?
9. Why do we use GROUP BY and ORDER BY function in mysql?
10. What is JOIN in MySQL? What are the different types of join?

Assignments no: 07

TITLE

Add dynamic web application essence using PHP, AJAX and MySQL.

OBJECTIVES

To understand the principles and methodologies of web based applications development process,

PROBLEM STATEMENT

Design and develop dynamic web application using PHP, AJAX and MySQL as a back-end for employee data with insert and view operations.

OUTCOMES

Students should be able to,

1. Develop web based application using suitable client side and server side web technologies.
2. Develop solution to complex problems using appropriate method, technologies, frameworks, web services and content management.

SOFTWARE & HARDWARE REQUIREMENTS

Software (Minimum Requirement):

1. Ubuntu 64 bit / Windows XP.
2. XAMPP Server

Hardware (Minimum Requirement):

Intel p4 Machine with 1GB RAM and 32GB HDD.

THEORY-CONCEPT

AJAX remains for Asynchronous JavaScript and XML. AJAX is another procedure for making better, speedier, and more intelligent dynamic web applications with the assistance of XML, HTML, CSS, and Java Script. Ajax utilizes XHTML for content, CSS for introduction, alongside Document Object Model and JavaScript for dynamic substance show.

Customary web applications transmit data to and from the server utilizing synchronous solicitations. It implies you round out a frame, hit submit, and get coordinated to another page with new data from the server. With AJAX, when you hit submit, JavaScript will influence a demand to the server, to decipher the outcomes, and refresh the present screen. In the purest sense, the client could never realize that anything was even transmitted to the server.

AJAX instructional exercise covers ideas and cases of AJAX innovation for apprentices and experts.

AJAX is an acronym for Asynchronous JavaScript and XML. It is a gathering of between related innovations like JavaScript, DOM, XML, HTML, CSS and so forth.

AJAX enables you to send and get information nonconcurrently without reloading the page. So it is quick.

AJAX enables you to send just essential data to the server not the whole page. So just profitable information from the customer side is steered to the server side. It makes your application intuitive and quicker.

AJAX speaks with the server utilizing XMLHttpRequest question. How about we endeavor to comprehend the stream of ajax or how ajax functions by the picture showed beneath.

AJAX communicates with the server using XML Http Request object. Let's try to understand the flow of ajax or how ajax works by the image displayed below:

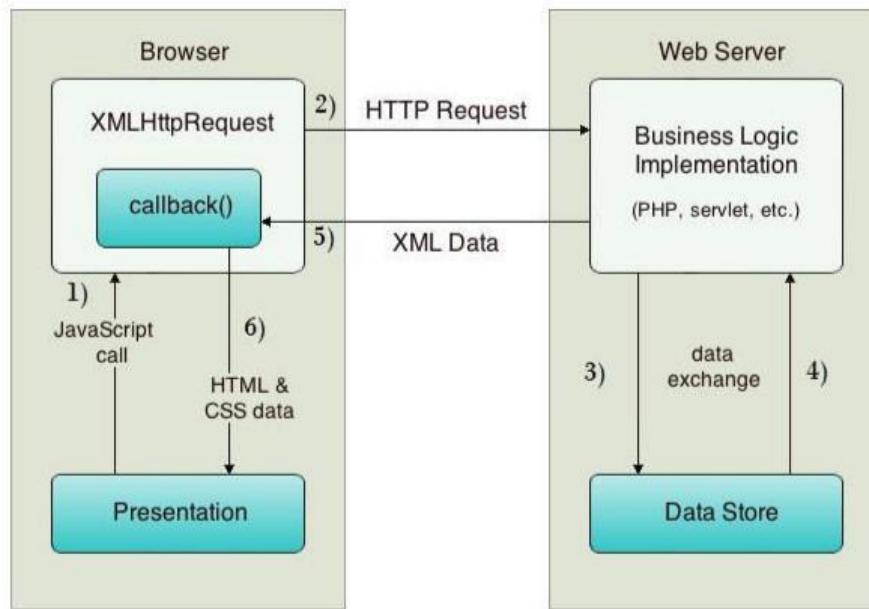


Figure.4: AJAX communication with the server using XML Http Request object

As you can see in the above example, XML Http Request object plays an important role.

TECHNOLOGY/TOOL

AJAX, PHP and MySQL

DESIGN/EXECUTION STEPS

For the design purpose html and CSS is to be used. For this design part contains the GUI of web applications, how its looks like? When users going to use the web application.

Steps to install XAMPP and configure the PHP, MYSQL server.

Follow same steps from the Assignment No. 4-A

CONCLUSION/ANALYSIS

In this assignment, we have studied how to design and develop small web application using PHP java script, ajax, XAMPP server with apache server and MySQL as backend.

PROGRAM CODE WITH OUTPUT

Here we have created “emp” directory in htdocs directory contain following files

1. index.html
2. insert.php
3. display.php
4. config.php

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-hwg4gsxgFZhOsEEamOYGBf13FyQuiTwIaQgxVSNgt4="
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>
```

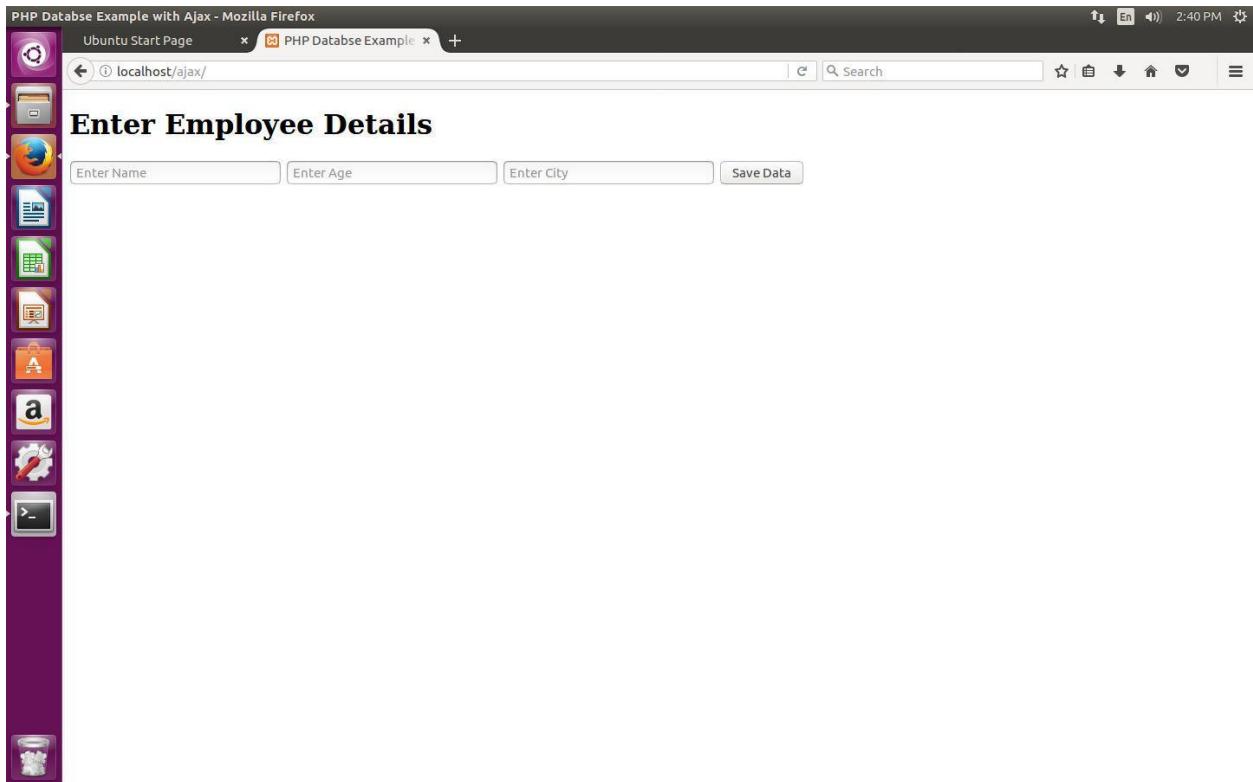
```
<a href="display.php">Display</a>

<!--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
```

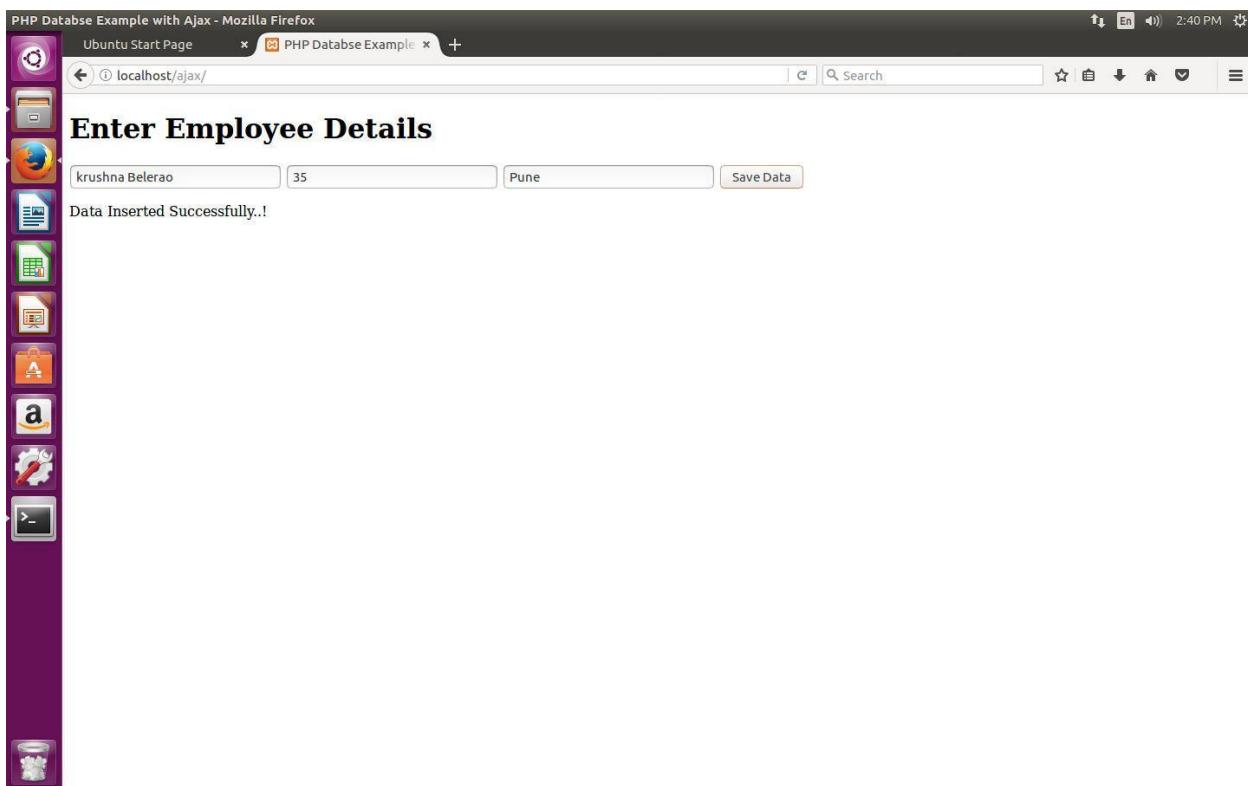
```
{
```

```
echo("fail");
```

```
}
```

```
?>
```

in below image URL you can see that without refreshing whole page only part of page refreshed.



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/>

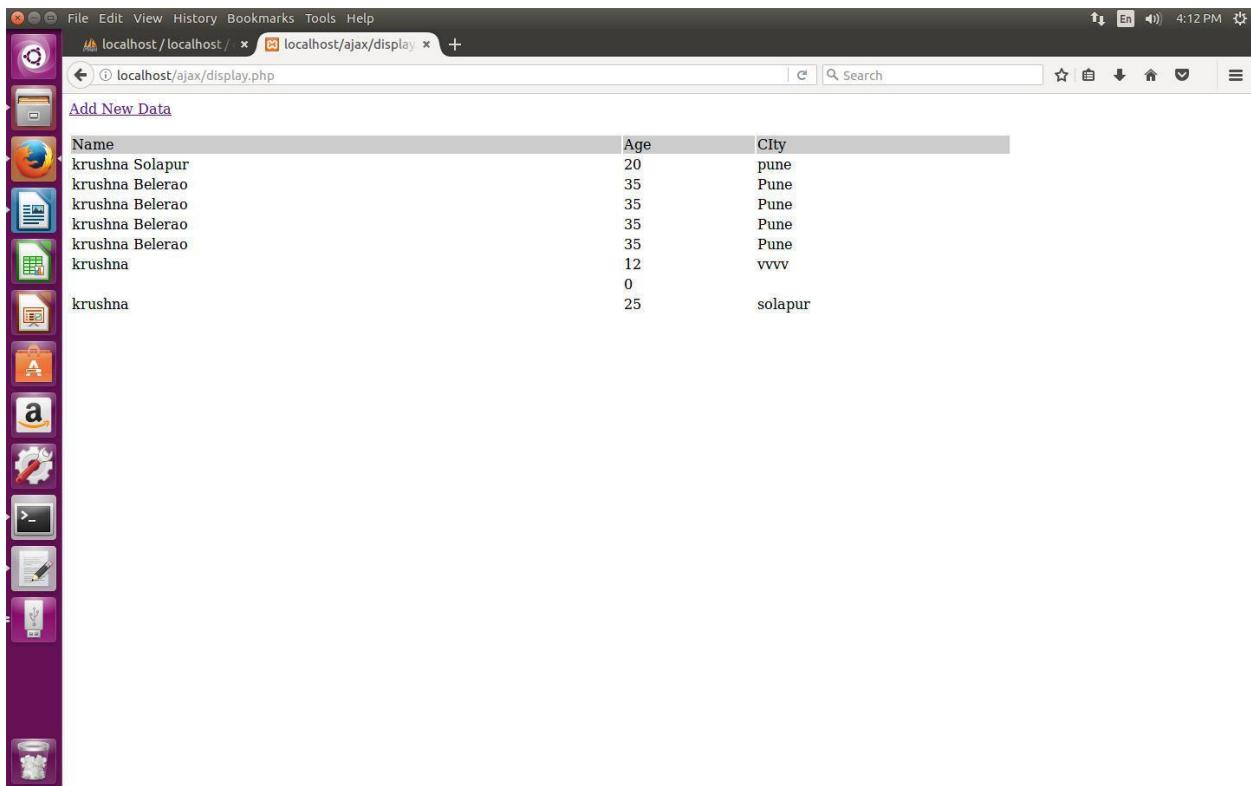
<table width='80%' border=0>

<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);  
?>
```

ORAL QUESTIONS

1. What is AJAX?
2. What is jQuery?
3. How many TRIGGERS allows per table in mysql?
4. What is difference between COMMIT and ROLLBACK?
5. What is Ajax?
6. Whether jQuery HTML work for both HTML and XML documents?
7. What is the use of jQuery.ajax method ()?
8. What are Ajax applications?
9. How to control the duration of an Ajax request?
10. What are the advantages and disadvantages of Ajax?
11. Which are the two methods used for cross domain Ajax calls?
12. What are all the technologies used by Ajax?
13. What is JSON in Ajax?
14. What are the difference between AJAX and Javascript?
15. How Ajax objects can be created?

Assignment No: 08

TITLE

Design and develop any web application using Struts framework.

OBJECTIVES

1. To impart the efficient and available client side and server side technologies.
2. To implement the communication between computing nodes using client side and server side technologies.
3. To design and implement the web services with content management.

PROBLEM STATEMENT

Create a login module for the web application using struts framework.

OUTCOMES

Students should be able to,

1. Implement the effective client side and server side technologies using struts framework.
2. Solve the complex problem of development using MVC framework.

SOFTWARE & HARDWARE REQUIREMENTS

Software's: Java 1.7 or Higher, Apache Tomcat 7 or higher, Struts API's, Eclipse IDE.

THEORY

The frameworks plays a vital role in industries for manageable and well designed application development as well as enterprise application development. The core of the Struts framework is a flexible control layer based on standard technologies like Java Servlets, JavaBeans, Resource Bundles, and XML, as well as various Jakarta Commons packages. Struts encourages application architectures based on the Model 2 approach, a variation of the classic Model-View-Controller(MVC)

Struts gives its own particular Controller segment and incorporates with different advancements to give the Model and the View. For the Model, Struts can collaborate with standard information get to advances, as JDBC and EJB, and also most any outsider bundles, as Hibernate, iBATIS, or Object Relational Bridge. For the View, Struts functions admirably with Java Server Pages, including JSTL and JSF, and in addition Velocity Templates, XSLT, and other introduction frameworks.

The Struts system gives the undetectable underpinnings each expert web application needs to survive. Struts causes you make an extensible advancement condition for your application, in view of distributed guidelines and demonstrated outline designs.

ORM remains for Object/Relational mapping. It is the customized and translucent constancy of items in a Java application in to the tables of a social database utilizing the metadata that portrays the mapping between the articles and the database. It works by changing the information starting with one portrayal then onto the next.

The Model-View-Controller Architecture

"Model-View-Controller" is a way to build applications that promotes complete separation between business logic and presentation. It is not specific to web applications, or Java, or J2EE (it predates all of these by many years), but it can be applied to building J2EE web applications.

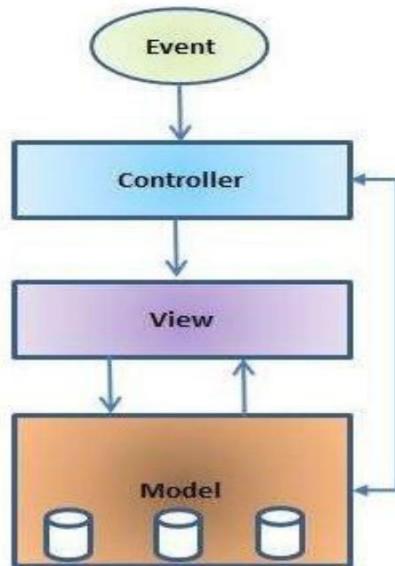


Figure.5: Basic MVC Architecture.

The "view" is the user interface, the screens that the end user of the application actually sees and interacts with. In a J2EE web application, views are JSP files. For collecting user input, you will have a JSP that generates an HTML page that contains one or more HTML forms. For displaying output (like a report), you will have a JSP generates an HTML page that probably contains one or more HTML tables. Each of these is a view: a way for the end user to interact with the system, putting data in, and getting data out.

What is Struts?

Struts is a framework that advances the utilization of the Model-View-Controller engineering for planning substantial scale applications. The structure incorporates an arrangement of custom label libraries and their related Java classes, alongside different utility classes. The most intense part of the Struts system is its help for making and preparing electronic structures. We will perceive how this functions later in this section.

Struts Tags

Common Attributes

Almost all tags provided by the Struts framework use the following attributes:

Attribute	Used for
Id	the name of a bean for temporary use by the tag
name	the name of a pre-existing bean for use with the tag
property	the property of the bean named in the name attribute for use with the tag
scope	the scope to search for the bean named in the name attribute

Table.2: Struts Framework Attributes

Creating Beans

Beans are created by Java code or tags.

Here is an example of bean creation with Java code:

```
// Creating a Plumber bean in the request scope
Plumber aPlumber = new Plumber();
request.setAttribute("plumber", aPlumber);
```

Beans can be created with the <jsp:useBean></jsp:useBean> tag:

```
<!-- If we want to do <jsp:setProperty ...></jsp:setProperty> or --
> <!-- <jsp:getProperty ... ></jsp:getProperty> -->
<!-- we first need to do a <jsp:useBean ... ></jsp:useBean> -->

<jsp:useBean id="aBean" scope="session" class="java.lang.String">
creating/using a bean in session scope of type java.lang.String
</jsp:useBean>
```

Most useful is the creation of beans with Struts tags:

```
<!-- Constant string bean -->
<bean:define id="greenBean" value="Here is a new constant string bean; pun intended."/>

<!-- Copying an already existent bean, frijole, to a new bean, lima -->
<bean:define id="lima" name="frijole"/>

<!-- Copying an already existent bean, while specifying the class -->
<bean:define id="lima" name="frijole" class="com.SomePackageName.Bean.LimaBean"/>

<!-- Copying a bean property to a different scope -->
<bean:define id="goo" name="foo" property="geeWhiz" scope="request" toScope="application"/>
```

Other Bean Tags

The Struts framework provides other tags for dealing with issues concerning copying cookies, request headers, JSP implicit defined objects, request parameters, web application resources,

Struts configuration objects, and including the dynamic response data from an action. These tags are not discussed here, but it is important to be aware of their existence.

```
<bean:cookie ... >
<bean:header ... >
<bean:page ... >
<bean:parameter ... >
<bean:header ... >
<bean:resource ... >
<bean:struts ... >
```

Bean Output

The `<bean:message>` and `<bean:write>` tags from the Struts framework will write bean and application resources properties into the current `HttpServletResponse` object.

This tag allows locale specific messages to be displayed by looking up the message in the application resources .properties file.

```
<!-- looks up the error.divisionByZero resource -->
<!-- and writes it to the HttpServletResponse object -->
<bean:message key="error.divisionByZero"/>

<!-- looks up the prompt.name resource -->
<!-- and writes it to the HttpServletResponse object; -->
<!-- failing that, it writes the string -->
<!-- contained in the attribute arg0-->
<bean:message key="prompt.name" arg0='Enter a name:'/>
```

This tag writes the string equivalent of the specified bean or bean property to the current `HttpServletResponse` object.

```
<bean:write ... >
    <!-- writes the value of customer.getStreetAddress().toString() -->
    <!-- to the HttpServletResponse object -->
    <bean:write name="customer" property="streetAddress"/>
```

Creating HTML Forms

Frequently data should be gathered from a client and handled. Without the capacity to gather client input, a web application would be futile. So as to get the clients data, a html shape is utilized. Client information can originate from a few gadgets, for example, content fields, content boxes, check takes care of, pop menus, and radio catches. The information comparing to the

client input is put away in an `ActionForm` class. A design document called `struts-config.xml` is utilized to characterize precisely how the client input are handled. The following diagram roughly depicts the use of Struts for using forms.

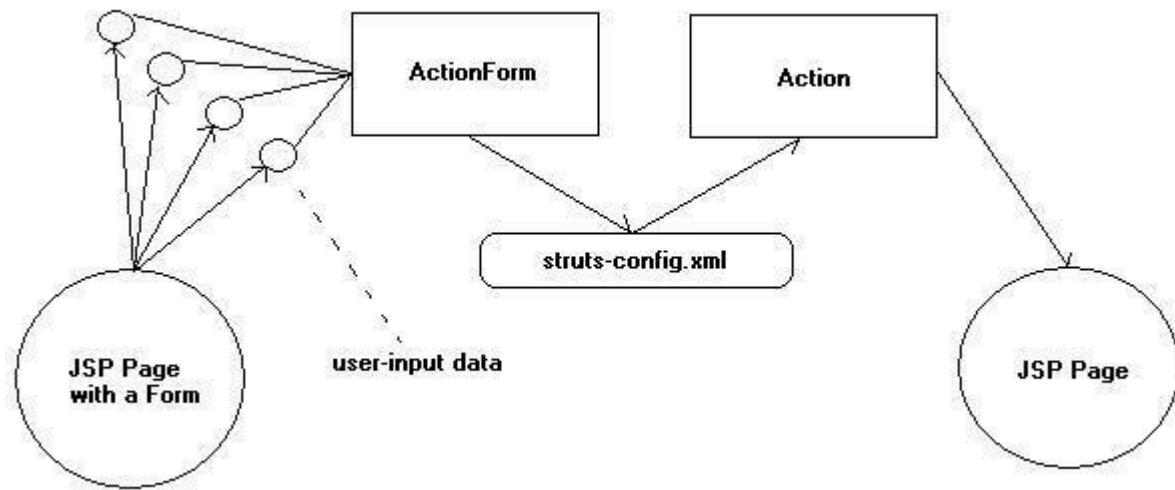


Figure.6: Struts for Using Forms

The Struts html tags are used to generate the widgets in the html that will be used in gathering the user's data. There are also tags to create a form element, html body elements, links, images, and other common html elements as well as displaying errors. Below are the tags provided by html section of the Struts framework and a short description of each.

`<html:base>` Generates a `<base>` tag. This tag should be used inside of a `<head>` tag.

`<html:button>` Generates an `<input type="button">` tag. This tag should be used inside a `<form>` element.

`<html:cancel>` Generates an `<input type="submit">` tag and causes the Action servlet not to invoke its `validate()` method. This tag should be used inside a `<form>` element.

Wheat Wood Clay

Stone Sheep

`<html:checkbox>` Generates an `<input type="checkbox">`.

`<html:multibox>` Generates an `<input type="checkbox">`. "Checkedness"

<html:errors>	depends upon whether the property array specified contains a corresponding value as the one specified for the multibox.
<html:file>	Generates html to display any errors that may have occurred during invocation of the validate() method.
<html:form>	Generates <form>.
<html:hidden>	<i>There is a hidden element here which is invisible. :-)</i> Generates <input type="hidden">.

The ActionForm class

The purpose of the ActionForm class is to contain and provide validation of the user-input data. This class is subclassed for application specific customization.

Here is a template for a customized ActionForm class with markers denoting where special items should be located in the class with \$ symbols.

```
package com.akurdi.action;

import com.opensymphony.xwork2.ActionSupport;

public class LoginAction extends ActionSupport {

    private String username;

    private String password;

    public String execute() {

        if(this.username == null || this.password == null)

        {

            return "error";

        }

        if (this.username.equals("admin"))

            && this.password.equals("admin123")) {

            return "success";
        }
    }
}
```

```
    } else {  
        addActionError(getText("error.login"));  
        return "error";  
    }  
}  
  
public String getUsername() {  
    return username;  
}  
  
public void setUsername(String username) {  
    this.username = username;  
}  
  
public String getPassword()  
{  
    return password;  
}  
  
public void setPassword(String password) {  
    this.password = password;  
}
```

Do this:

1. Create the directory structure. The root directory is SimpleStruts, and it has the standard WEB-INF directory with classes inside, and com.akurdi.action inside that. It also has a lib directory within WEB-INF, which is something we haven't seen before; we'll see in a minute what goes there.

2. Copy the Struts tag library descriptor files into WEB-INF. The files struts.tld, struts-bean.tld, struts-form.tld, struts-html.tld, struts-logic.tld, and struts-template.tld are available in the lib directory of your Struts installation.
3. Copy the Struts parser, struts.jar, into WEB-INF/lib/. This file is available in the lib directory of your Struts installation
4. Create the tag descriptor library file for any custom tags you may use beyond the Struts tags. In this case, the file defines no custom tags, but it's good practice to have it in place, in case you need to add your own tags later.
5. Create the struts-config.xml file.

There are three main sections to a struts-config.xml configuration file. They are the "Form Bean Definitions" section, the "Global Forward Definitions" section, and the "Action Mapping Definitions" section defines a forward called "success".

6. Create the web.xml file.

The web.xml web application configuration file will need to define the servlet ActionServlet, to which control will be transferred whenever an appropriate URL pattern is accessed. The servlet is defined just as any other servlet will be defined. The URL pattern is specified by a servlet mapping. For this application, the URL pattern is any requested resource that ends with a .do extension.

In order to use the Struts tags, the .tld files describing the tags will need to be included in the configuration file. The references to these tags are made just as they were for our own custom tags in the previous chapter. The Struts framework is simply a complex set of tag libraries (struts*.tld), with associated code (struts.jar).

7. The ApplicationResources.properties file provides resources that will be used by any subclassed Struts classes (for example, SetNameAction). This resources file provides a place to define prompts, labels that will display on buttons, and other information that may change. By placing this information in the ApplicationResources.properties file, recompiling any servlets used in the application can be avoided, as well as encouraging separation of logic and presentation.
8. Instances of the Name class are placed in the user sessions. Only one will exist in any particular user session. It provides methods for accessing and mutating a name.
9. The view of the application is done with the JSP index.jsp. It represents the user interface and allows the user to interact with the application.
10. Compile, create .war file and place it into the webapps folder of tomcat.

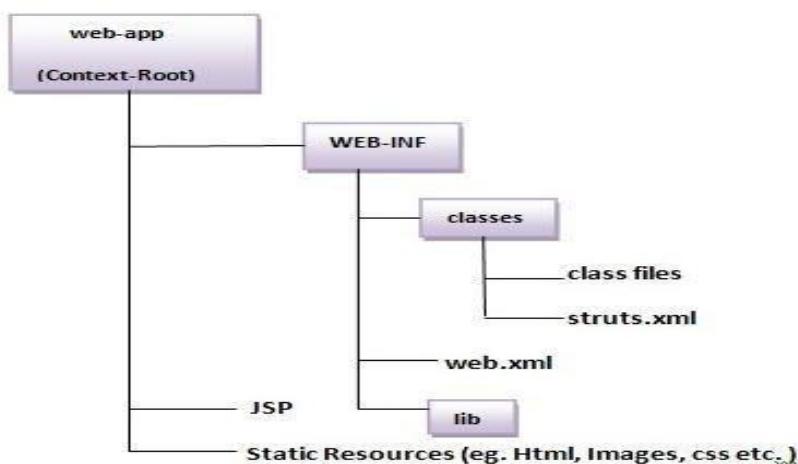
11. Go to <http://localhost:8080/Sturtsdemo/example/Login.jsp> to test your application

TECHNOLOGY/TOOL

- 1) Eclipse IDE
- 2) Apache Tomcat 7.0 or higher

DESIGN/EXECUTION STEPS

Step 1) Create the directory structure as



Step 2) Create input page as below.

Step 3) Provide the entry of Controller in (web.xml) file as given above

Step 4) Create the action class (LoginAction.java)

Step 5) Map the request in (struts-config.xml) file and define the view components.

Step 6) Load the jar files

Step 7) start server and deploy the project or create .war file paste it in webapps folder and run from manager-app.

TEST CASES

1. Manual test cases need to be performed on struts application.
2. Check whether the tomcat server is running.
3. Check whether the application running status in manager app list is true.
4. Run the application by selecting the app in the list.
5. Check whether the application giving desired results.

CONCLUSION/ANALYSIS

Hence we have successfully tested the Struts framework and tested the results.

PROGRAM CODE: INPUT & OUTPUT

- 1) Deploy the application from tomcat manager.
- 2) Provide the login username and password for login.
- 3) Test the result for correct and incorrect credentials.

ORAL QUESTIONS

- 1) What are the components of Struts Framework?
- 2) What's the role of a handler in MVC based applications?
- 3) What's the flow of requests in Struts based applications?
- 4) Which file is used by controller to get mapping information for request routing?
- 5) What's the role of Action Class in Struts?
- 6) How an actionForm bean is created? And Its uses.
- 7) How validation is performed in struts application?
- 8) What's the purpose of Execute method of action class?
- 9) How can we display all validation errors to user on JSP page?
- 10) What are the benefits of Struts framework?

Assignment No: 09

TITLE

Design and develop any web application using AngularJS.

OBJECTIVES

1. Understand the design of single-page applications and how AngularJS facilitates their development
2. Properly separate the model, view, and controller layers of your application and implement them using AngularJS
3. Master AngularJS expressions, filters, and scopes
4. Build Angular forms
5. Elegantly implement Ajax in your AngularJS applications
6. Write AngularJS directives

PROBLEM STATEMENT

Create an application for Bill Payment Record using AngularJS

OUTCOMES

Students can able to,

1. Implement the effective client side implementation.
2. Solve the complex problem of development using MVC framework.

SOFTWARE & HARDWARE REQUIREMENTS

Software's: Eclipse IDE/ Notepad/ Notepad++, Modern Web browser

THEORY-CONCEPT

AngularJS is an open-source web application framework. It was initially created in 2009 by MiskoHevery and Adam Abrons. It is presently kept up by Google. Its most recent adaptation is 1.2.21. "AngularJS is an auxiliary system for dynamic web applications. It gives you a chance to

utilize HTML as your layout dialect and gives you a chance to stretch out HTML's linguistic structure to express your application parts plainly and compactly. Its information official and reliance infusion take out a significant part of the code you as of now need to compose. Also, everything occurs inside the program, making it a perfect band together with any server innovation".

General Features

- AngularJS is a productive system that can make Rich Internet Applications (RIA).
- AngularJS gives designers a choices to compose customer side applications utilizing JavaScript in a spotless Model View Controller (MVC) way.
- Applications written in AngularJS are cross-program agreeable. AngularJS consequently handles JavaScript code reasonable for every program.
- AngularJS is open source, totally free, and utilized by a great many engineers the world over. It is authorized under the Apache permit version2.0.
- By and large, AngularJS is a system to assemble expansive scale, elite, and simple to-keep up web applications.

Core Features:

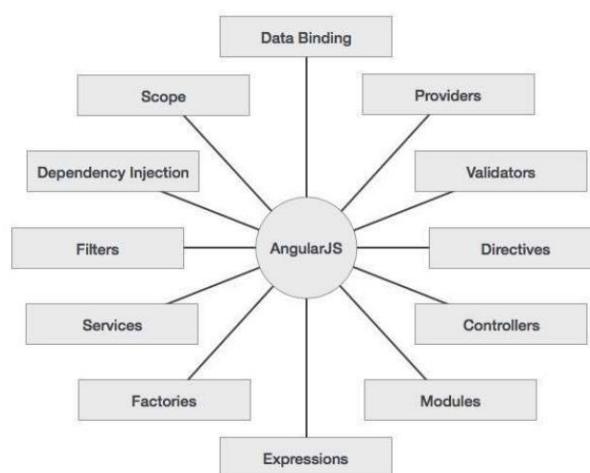


Figure.7: Architecture of AngularJS

1. **Data-authoritative:** It is the programmed synchronization of information amongst model and view parts.

2. **Scope:** These are objects that allude to the model. They go about as paste amongst controller and view.
3. **Controller:** These are JavaScript capacities bound to a specific degree.
4. **Services:** AngularJS accompanies a few implicit administrations, for example, \$http to make aXMLHttpRequests. These are singleton objects which are instantiated just once in application.
5. **Filters:** These select a subset of things from a cluster and restore another exhibit.
6. **Directives:** Directives are markers on DOM components, for example, components, characteristics, css, and that's only the tip of the iceberg. These can be utilized to make custom HTML labels that fill in as new, custom gadgets. AngularJS has worked in mandates, for example, ngBind, ngModel, and so on.
7. **Templates:** These are the rendered see with data from the controller and model. These can be a solitary record, (for example, index.html) or different perspectives in a single page utilizing partials.
8. **Routing:** It is idea of exchanging sees.
9. **Model View Whatever:** MVW is an outline design for isolating an application into various parts called Model, View, and Controller, each with unmistakable obligations. AngularJS does not actualize MVC in the conventional sense, yet rather something nearer to MVVM (Model-View-ViewModel). The Angular JS group alludes it cleverly as Model View Whatever.
10. **Deep Linking:** Deep connecting permits to encode the condition of use in the URL with the goal that it can be bookmarked. The application would then be able to be re-established from the URL to a similar state.

11. Dependency Injection: AngularJS has a worked in reliance infusion subsystem that encourages the designer to make, comprehend, and test the applications effectively.

Advantages of AngularJS

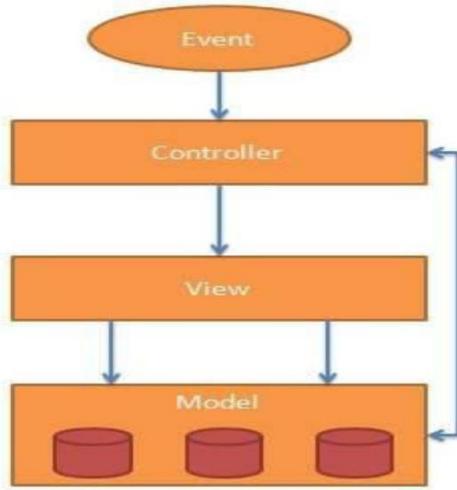
- It gives the ability to make Single Page Application in a spotless and viable way.
- It gives information restricting ability to HTML. Along these lines, it gives client a rich and responsive experience.
- AngularJS code is unit testable.
- AngularJS utilizations reliance infusion and make utilization of partition of concerns.
- AngularJS gives reusable segments.
- With AngularJS, the engineers can accomplish greater usefulness with short code.
- In AngularJS, sees are unadulterated html pages, and controllers written in JavaScript do the business handling.

Model View Controller

Model View Controller or MVC as it is famously called, is a product configuration design for creating web applications. A Model View Controller design is comprised of the accompanying three sections.

- **Model – It is the most minimal level of the example in charge of looking after information.**
- **View – It is in charge of showing all or a part of the information to the client.**
- **Controller – It is a product Code that controls the connections between the Model and View.**

MVC is mainstream since it secludes the application rationale from the UI layer and backings detachment of concerns. The controller gets all solicitations for the application and afterward works with the model to set up any information required by the view.

**Figure. 8: Model View Controller**

Model

The model is in charge of overseeing application information. It reacts to the demand from view and to the directions from controller to refresh itself.

The View

An introduction of information in a specific arrangement, activated by the controller's choice to exhibit the information. They are content based layout frameworks, for example, JSP, ASP, PHP and simple to incorporate with AJAX innovation.

The Controller

The controller reacts to client enter and performs communications on the information show objects. The controller gets input, approves it, and afterward performs business operations that alter the condition of the information demonstrate.

AngularJS is a MVC based structure.

- **An AngularJS application comprises of following three essential parts –ng-app –**
- This directive defines and links an AngularJS application to HTML.**

- **ng-model** – This directive binds the values of AngularJS application data to HTML input controls.
- **ng-bind** – This directive binds the AngularJS Application data to HTML tags.

DESIGN/EXECUTION STEPS

Steps for AngularJS

1. When a link <https://angularjs.org/> is opened, there are two options to download AngularJS library –



- **View on GitHub** – Click on this button to go to GitHub and get all of the latest scripts.
- **Download AngularJS 1** – Or click on this button, a screen as below would be seen –

Download AngularJS

This screenshot shows the "Download" configuration page for AngularJS. It includes several tabs for different deployment methods:

- Branch:** 1.5.x (stable) (selected), 1.2.x (legacy)
- Build:** Minified, Uncompressed, Zip
- CDN:** <https://ajax.googleapis.com/ajax/libs/angularjs/1.5.2/angular.min.js>
- Bower:** bower install angular#1.5.2
- npm:** npm install angular@1.5.2
- Extras:** Browse additional modules

At the bottom, there are links for "Previous Versions" and a large blue "Download" button with a downward arrow icon.

- This screen gives various options of using Angular JS as follows:

- **Downloading and hosting files locally**
 1. There are two different options **legacy** and **latest**. The names itself are self-descriptive. **Legacy** has version less than 1.2.x and **latest** has 1.5.x version.
 2. We can also go with the minified, uncompressed or zipped version.
- **CDN access – You** also have access to a CDN. The CDN will give you access around the world to regional data centers that in this case, Google host. This means using CDN moves the responsibility of hosting files from your own servers to a series of external ones. This also offers an advantage that if the visitor to your webpage has already downloaded a copy of AngularJS from the same CDN, it won't have to be re-downloaded.

Example:

Now let us write a simple example using AngularJS library. Let us create an HTML file *myfirstexample.html* as below –

```
<!doctype html>
<html>

<head>
<scriptsrc="https://ajax.googleapis.com/ajax/libs/angularjs/1.5.2/angular.min.js"></script>
</head>

<bodyng-app="myapp">

<divng-controller="HelloController">
<h2>Welcome {{helloTo.title}} to the world of
Tutorialspoint!</h2> </div>

<script>
angular.module("myapp",[])
.controller("HelloController",function($scope){
```

```

$scope.helloTo={ };
$scope.helloTo.title="AngularJS";
});
</script>
</body>
</html>

```

Following sections describe the above code in detail:

1. Include AngularJS

We have included the AngularJS JavaScript file in the HTML page so we can use AngularJS –

```

<head>
<scriptsrc="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
</head>

```

To update into latest version of Angular JS, use the following script source.

```

<head>
<scriptsrc="https://ajax.googleapis.com/ajax/libs/angularjs/1.5.2/angular.min.js"></script>
</head>

```

2. Point to AngularJS app

Next we tell what part of the HTML contains the AngularJS app. This done by adding the *ng-app* attribute to the root HTML element of the AngularJS app. You can either add it to *html* element or *body* element as shown **below** –

```

<bodyng-app="myapp">
</body>

```

3. View

The view is this part –

```

<divng-controller="HelloController">
<h2>Welcome {{helloTo.title}} to the world of Tutorialspoint!</h2>

```

```
</div>
```

ng-controller tells AngularJS what controller to use with this view. *helloTo.title* tells AngularJS to write the "model" value named *helloTo.title* to the HTML at this location.

4. Controller

The controller part is –

```
<script>
angular.module("myapp",[])
.controller("HelloController",function($scope){
    $scope.helloTo={};
    $scope.helloTo.title="AngularJS";
});
</script>
```

This code registers a controller function named *HelloController* in the angular module

named *myapp*. The controller function is registered in angular via the `angular.module(...).controller(...)` function call.

The `$scope` parameter passed to the controller function is the *model*. The controller function adds a *helloTo* JavaScript object, and in that object it adds a *title* field.

5. Execution

Save the above code as *myfirstexample.html* and open it in any browser.

Output as below:

```
Welcome AngularJS to the world of Tutorialspoint!
```

- At the point when the page is stacked in the program, following things happen –
- HTML archive is stacked into the program, and assessed by the program. AngularJS JavaScript document is stacked, the precise worldwide question is made. Next, JavaScript which registers controller capacities is executed.

- Next AngularJS look over the HTML to search for AngularJS applications and perspectives. When see is found, it associates that view to the comparing controller work.
- Next, AngularJS executes the controller capacities. It at that point renders the perspectives with information from the model populated by the controller. The page is presently prepared.

6. How AngularJS integrates with HTML

- ng-app directive indicates the start of AngularJS application.
- ng-model directive then creates a model variable named "name" which can be used with the html page and within the div having ng-app directive.
- ng-bind then uses the name model to be displayed in the html span tag whenever user input something in the text box.
- Closing</div> tag indicates the end of AngularJS application.

AngularJS directives are used to extend HTML. These are special attributes starting with ng-prefix. We're going to discuss following directives –

- **ng-app – This directive starts an AngularJS Application.**
- **ng-init – This directive initializes application data.**
- **ng-model – This directive binds the values of AngularJS application data to HTML input controls.**
- **ng-repeat – This directive repeats html elements for each item in a collection.**

TECHNOLOGY/TOOL

Any IDE or you can use web browser.

TEST CASES

Deploy the Html program run test the result for dynamic implementation of AngularJS.

CONCLUSION/ANALYSIS

With the help of this assignment it is helpful to understand features of AngularJS. MVC model structure and its use in advanced web programming is studied.

PROGRAM CODE: INPUT & OUTPUT

```
<html ng-app="billpayApp">

<!-- SCRIPTS TO BE ADDED IN HEAD TAG --
&gt; &lt;head&gt;
&lt;title&gt;Bill Payment Record using angular and bootstrap
framework&lt;/title&gt;

&lt;meta http-equiv="content-type" content="text/html; charset=utf-8" /&gt;

<!-- ACCESSING ANGULARJS BY CDN METHOD--&gt;
&lt;script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.4/angular.min
.js"&gt;&lt;/script&gt;

<!-- ACCESSING STYLESHEET FOR DESIGN [OPTIONAL PART CAN BE SKIP]--&gt;
&lt;link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.mi
n.css"&gt;

<!-- MODEL PART--&gt;
&lt;script&gt;
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", []);</pre>
```

```

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</b></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" ng-
click="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>

<tbody ng-model="rmvBills">
<tr ng-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>
```



Student's Bill's remained to Be Paid - 3

Add extra biller fields if any

Bill Name	Status	Status	Close
Electricity	<input type="checkbox"/>	false	<input type="button" value="x"/>
Internet(Wi-fi)	<input type="checkbox"/>	false	<input type="button" value="x"/>
Parking Charges	<input type="checkbox"/>	false	<input type="button" value="x"/>
Phone	<input checked="" type="checkbox"/>	true	<input type="button" value="x"/>
House Tax	<input checked="" type="checkbox"/>	true	<input type="button" value="x"/>



Student's Bill's remained to Be Paid -

Add extra biller fields if any

Bill Name	Status	Status	Close
Internet(Wi-fi)	<input checked="" type="checkbox"/>	true	<input type="button" value="x"/>
Phone	<input checked="" type="checkbox"/>	true	<input type="button" value="x"/>
House Tax	<input checked="" type="checkbox"/>	true	<input type="button" value="x"/>



ORAL QUESTIONS

1. What is AngularJS and what are some of its advantages?
2. What is the Model View Controller (MVC)?
3. What is data binding in AngularJS? How does it relate to the MVC architecture?
4. Explain the concept of scope. How does scope inheritance work in AngularJS?
5. Explain the difference between a factory and a service in AngularJS.
6. Explain why there are two “destroy” events associated with the termination of a scope in AngularJS.
7. What is dependency injection and how does it work?
8. What are directives? Can you explain the functions of the following directives?
9. Explain the role of \$routeProvider in AngularJS.

Assignment No. - 10

TITLE

Web Application using EJB

OBJECTIVES

1. Understand about basic concepts of java beans.
2. Understand the basic functionalities of JSP, HTML.
3. Having the knowledge of JBOSS server to deploy web application.

PROBLEM STATEMENTS

Design, Develop & Deploy web application using EJB.

OUTCOMES

Students will be able to,

1. Develop a dynamic webpage using Java Beans, HTML and JSP.
2. To understand the concepts and method of web based applications development Process using EJB.
3. Create a simple EJB 3 stateless session bean and a local Java application client which will call/invoke the bean to develop for addition of two numbers.

SOFTWARE NEEDED

1. Ubuntu 64 bit / Windows 7.
2. JDK 7 (Java SE 7)
3. EJB 3.0 (stateless session bean)
4. Eclipse luna
5. JBoss Application Server (AS) 7.1.1

THEORY – CONCEPT

Java Beans :

J2EE application container contains the components that can be used by the clients for executing the business logic .These components are known as Enterprise Java Beans (EJB) .

J2EE platform has component based architecture to provide multi-tiered, distributed and highly transactional features to enterprise level applications.

EJB mainly contains the business logic & business data. EJB component is an EJB class. It is a java class written by EJB developer & this class implements business logic.

It is used for developing very much scalable and robust enterprise level applications to be deployed Application Server such as JBOSS, Web Logic etc.

EJB 3.0 is being a large shift from EJB 2.0 and makes development of EJB based applications relatively easy.

Features of EJBs:

Some of the features of an application server include the following:

- **Client Communication:** The client, which is often a user interface, must be able to call the methods of objects on the application server via agreed-upon protocols.
- **State Management:** You'll recall our discussions on this topic in the context of JSP (JavaServer Pages) and servlet development back in Chapter 6.
- **Transaction Management:** Some operations, for example, when updating data, must occur as a unit of work. If one update fails, they all should fail.
- **Database Connection Management:** An application server must connect to a database, often using pools of database connections for optimizing resources.
- **User Authentication and Role-Based Authorization:** Users of an application must often log in for security purposes. The functionality of an application to which a user is allowed access is often based on the role associated with a user ID.
- **Asynchronous Messaging:** Applications often need to communicate with other systems in an asynchronous manner; that is, without waiting for the other system to respond. This requires an underlying messaging system that provides guaranteed delivery of these asynchronous messages.
- **Application Server Administration:** Application servers must be administered. For example, they need to be monitored and tuned.

Types of Enterprise Java Beans (EJB):

There are three types of Enterprise Java Beans namely:

1. Session Beans
2. Entity Beans
3. Message driven beans

Session Beans

- Session beans are intended to allow the application author to easily implement portions of application code in middleware and to simplify access to this code.
- Represents a single client inside the server
- The client calls the session bean to invoke methods of an application on the server
- Perform works for its client, hiding the complexity of interaction with other objects in the server
- Is not shared
- Is not persistent

When the client stops the session, the bean can be assigned to another client from the server
Session beans are divided into two types:

1. Stateless Session Bean:

Stateless Session Bean is intended to be simple and “light weight” components. The client, thereby making the server highly scalable, if required, maintains any state. Since no state is maintained in this bean type, stateless session beans are not tied to any specific client, hence any available instance of a stateless session bean can be used to service a client.

- values only for the duration of the single invocation
- Except during method invocation, all instances of stateless session bean are equivalent

Stateless Session Bean's Life Cycle:

- The client invoke the create method
- The EJB container :
 - Instantiates the bean
 - Invokes the setSessionContext
 - Invokes ejbCreate
- The bean is ready
- While in the ready state

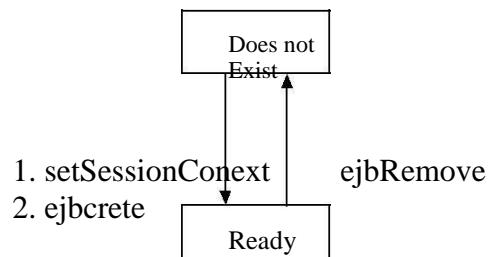


Figure.9: Stateless Session Bean's Life Cycle

client may invoke a business method :

A client may invoke the remove method and the container calls the bean's ejbRemove method
It's never passivate .

2. Stateful Session Bean:

State ful Session Bean provides easy and transparent state management on the server side.
Because state is maintained in this bean type, the application server manages client/bean pairs.

Stateful session beans can access persistent resources on behalf of the client, but unlike entity beans, they do not actually represent the data.

Stateful Session Beans Life Cycle:

- The client invoke the create method
- The EJB container :
Instantiates the bean
- The bean is ready
- While in the ready state
container may passivate the bean moving it from memory to secondary storage
- - A client may invoke a business method

EJB container may activate a bean, moving it back to the ready stage, and then calls the bean's ejbActivate method.

A client may invoke the remove method and the container calls the bean's ejbRemove method

Difference Between Stateless and State Full EJB Are as Follows

Stateless:

1. Normally data members are not put in stateless session bean
2. Stateless beans are pooled

3. No effort for keeping client specific data
4. No Activation/Passivation in stateless session bean

Stateful:

1. Data members that represent state are present in stateful session bean
2. Stateful beans are cached
3. Setting the tag idle-timeout-seconds determines how long data is maintained in stateful session bean
4. Activation – Passivation used

An Entity Bean

- An entity bean is an object representation of persistent data maintained in a permanent data store such as a database. A primary key identifies each instance of an entity bean. Entity beans are transactional and are recoverable in the event of a system crash.
- Entity beans are representations of explicit data or collections of data, such as a row in a relational database. Entity bean methods provide procedures for acting on the data representation of the bean. An entity bean is persistent and survives if its data remains in the database.
- An entity bean can implement either bean-managed or container-managed persistence. In the case of bean-managed persistence, the implementer of an entity bean stores and retrieves the information managed by the bean through direct database calls. The bean may utilize either Java Database Connectivity (JDBC) or SQL-Java (SQLJ) for this method.
- In the case of container-managed persistence, the container provider may implement access to the database using standard APIs. The container provider can offer tools to map instance variables of an entity bean to calls to an underlying database. The container saves the data. There is no code in the bean for access the database. The container handles all database access required for the bean which create links between beans are created using a structure called abstract schema.

Enterprise Java Beans (EJB) Architecture

The EJB architecture is an extension of Web architecture. It has an additional tier. The clients of an enterprise bean can be a traditional java application lie, applet, JSP or Servlet.

Like in a web application, client browser has to go all the way to web container to use a servlet or JSP, the communication between beans and clients is performed by the EJB container.

The following are the flows of the EJB architecture.

- The client is working on a web browser.
- There is a database server that hosts a database, like MySQL /Oracle.

- The J2EE server machine is running on an application server
- The client interface is provided with JSP/Servlet. The enterprise beans reside in the business tier providing to the client tier.
- The Application Server manages the relationships between the client and database machines.

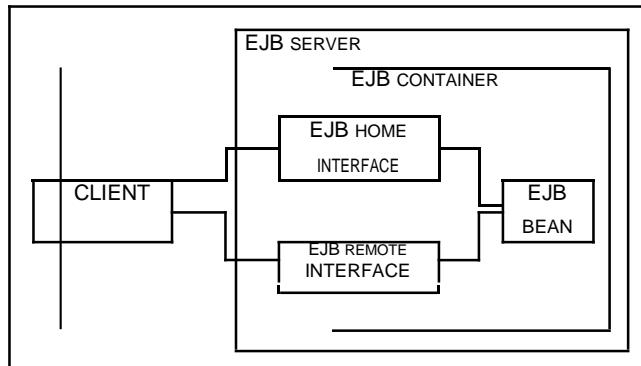


Figure.10: EJB Architecture

- In a diagram, an enterprise bean is a non-visual component of a distributed, transaction-oriented enterprise application. Enterprise beans are typically deployed in EJB containers and run on EJB servers.

There are three types through which two or more activities may interfere:

1. Dirty read
 2. Non-Repeatable read
 3. Phantom read
- **Clustering and Load-Balancing:** Clustering is the process of combining the multiple peripherals, computers and other resources into a single unit.
 - A clustered system then works as load balanced system. In a distributed system when a request is send to the server, an algorithm running on the server decides which server has less load and sends the request to that server. EJB container encapsulates these features to provide smooth and efficient service.
 - **Deployment Descriptor:** A deployment descriptor is an XML file packaged with the enterprise beans in an EJB JAR file or an EAR file. It contains metadata describing the contents and structure of the enterprise beans, and runtime transaction and security information for the EJB container.
 - **EJB Server:** An EJB server is a high-level process or application that provides a runtime environment to support the execution of server applications that use enterprise beans. An EJB server provides a JNDI-accessible naming service. It manages and coordinates the allocation of resources to client applications, provides access to system resources and provides a transaction service.

DESIGN / EXECUTION STEPS

Following steps are used to Create and Execute web applications,

1. Design EJB project.
2. Start JBOSS & Deploy it on JBOSS server.
3. Design html and jsp files with an extension of.html and .jsp
4. Run the application in browser and get the result

TEST CASES

Manual testing is used to check the application is running properly in JBOSS server

CONCLUSION / ANALYSIS

Hence, we have created a simple EJB 3 stateless session bean and a local Java application client which will call/invoke the bean to develop for performing addition of two numbers.

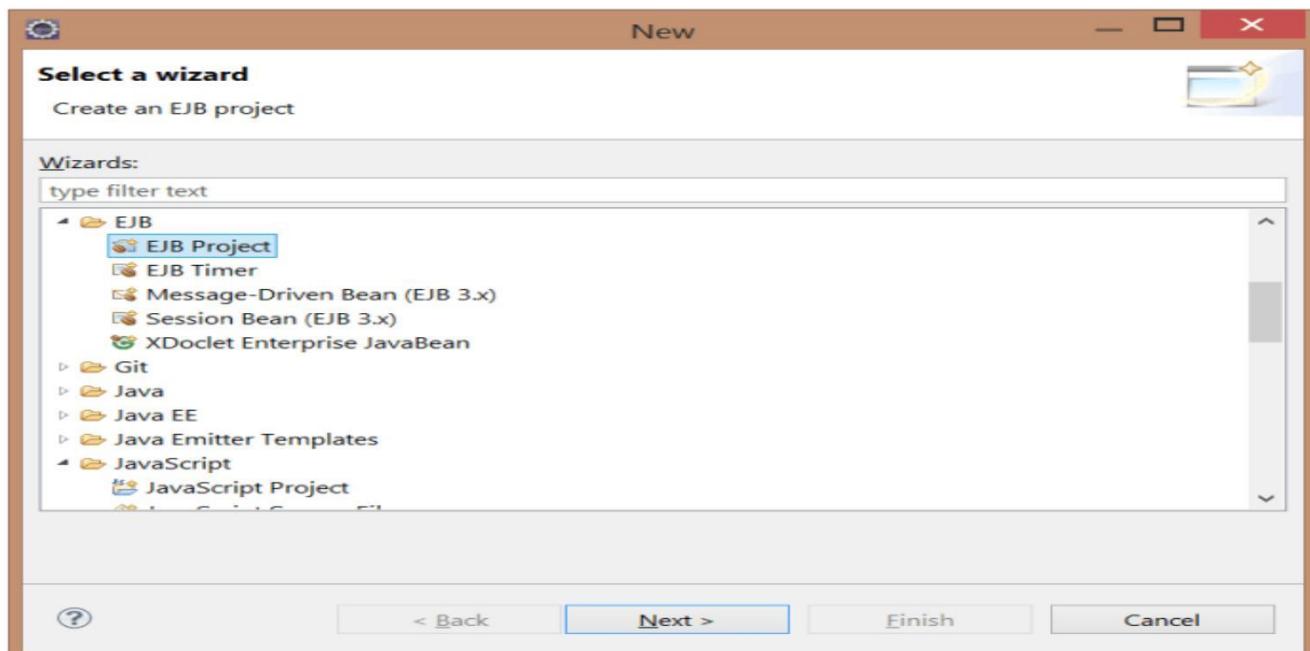
PROGRAM CODE: INPUT & OUTPUT

Create a new EJB Project :

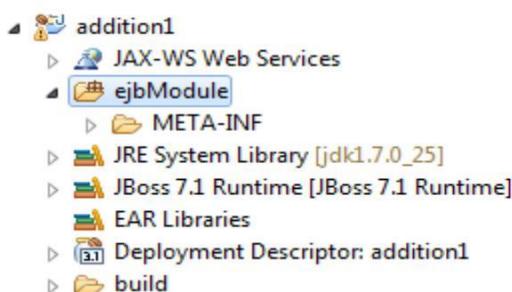
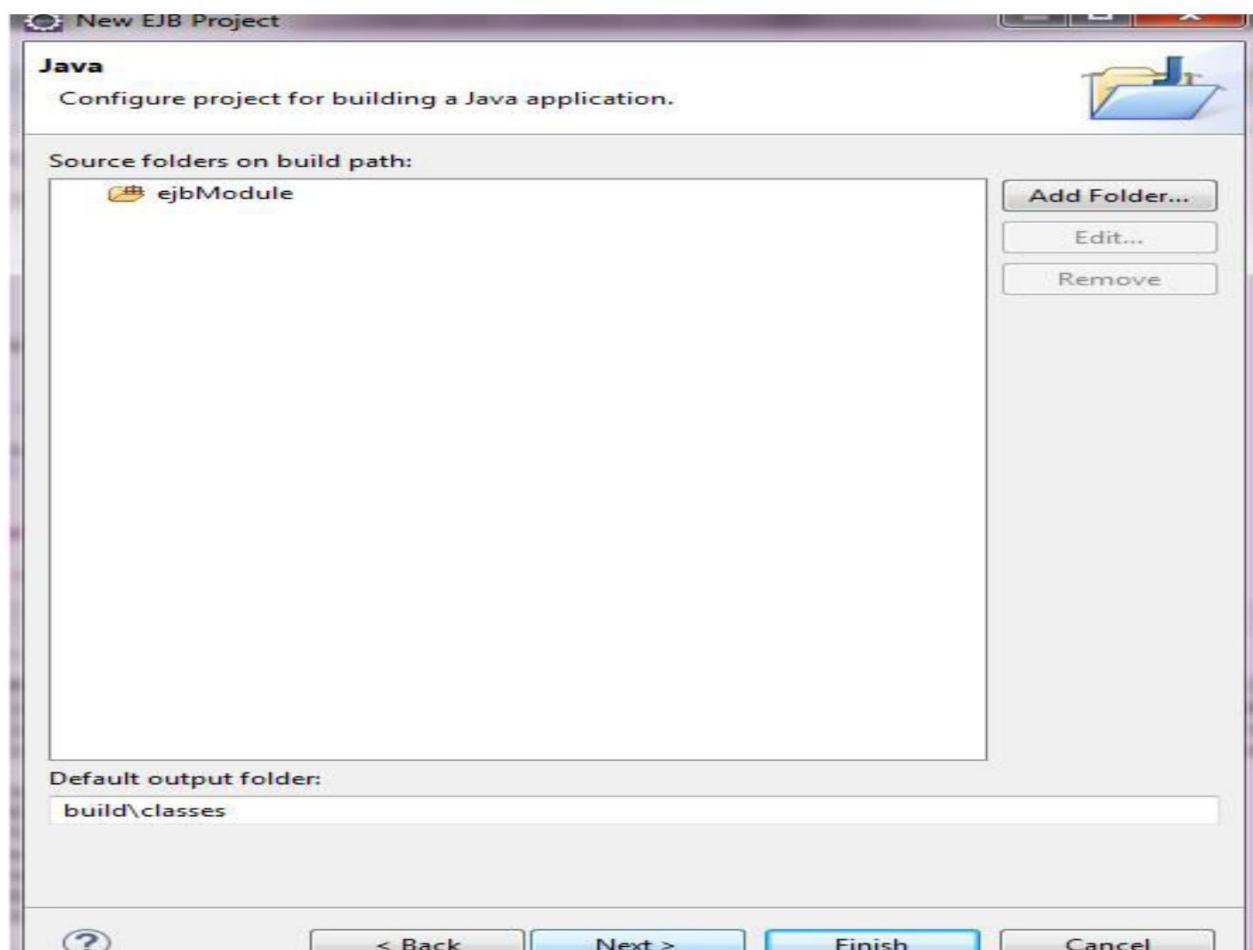
Open Eclipse IDE and create a new EJB project which can be done by clicking on, File menu -> New -> EJB Project

Step 1:

- Create EJB project addition
- Click File -> New -> Other -> EJB -> EJB Project -> Next

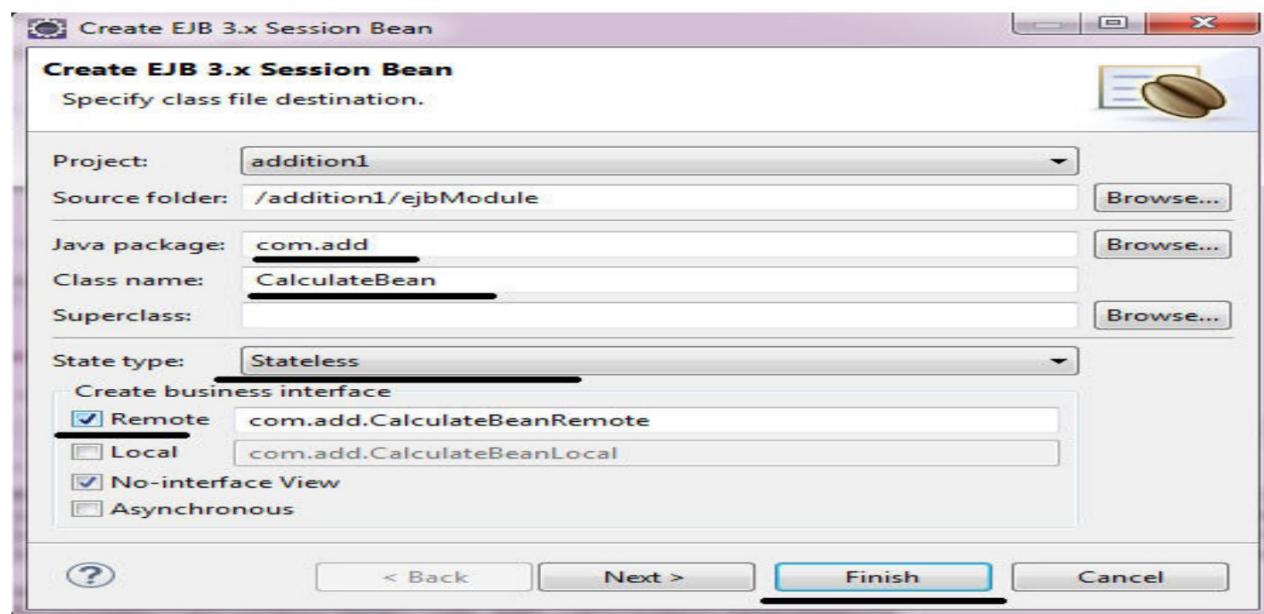
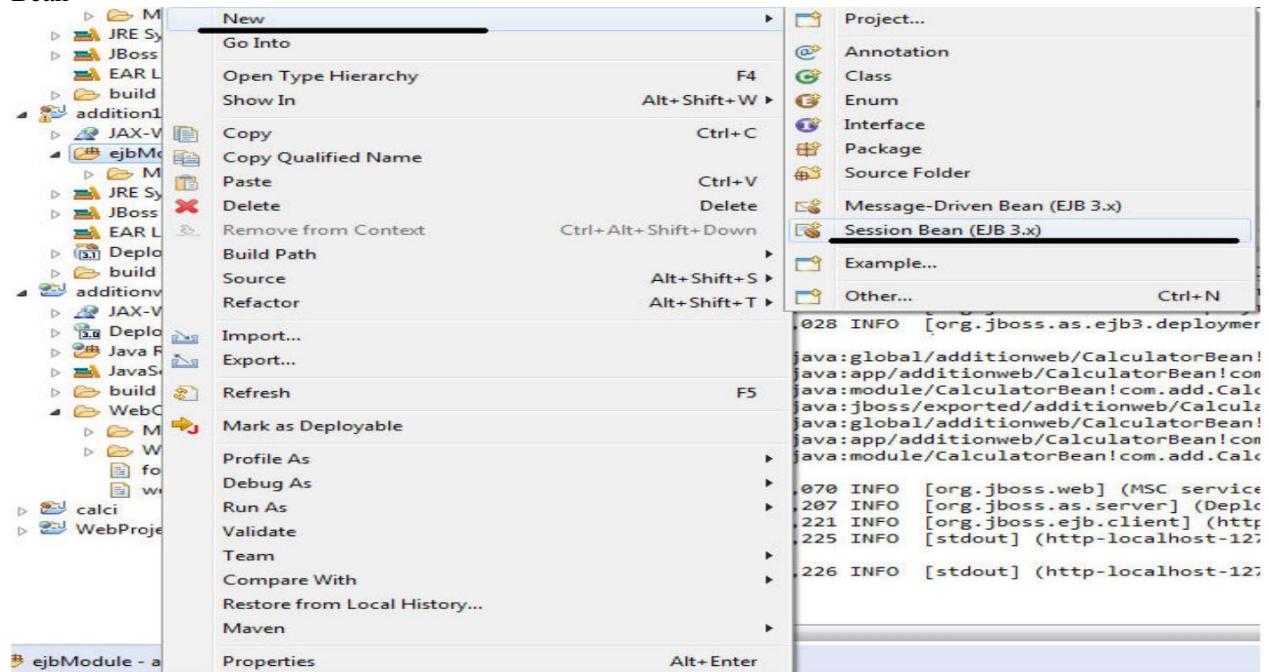


The screenshot shows the 'EJB Project' configuration dialog box. The title bar says 'EJB Project' and has a sub-instruction 'Create an EJB Project and add it to a new or existing Enterprise Application.' Below the title are several configuration sections: 'Project name' (set to 'addition1'), 'Project location' (with a checked 'Use default location' checkbox and a 'Location' field set to 'D:\workplace\addition1'), 'Target runtime' (set to 'JBoss 7.1 Runtime'), 'EJB module version' (set to '3.1'), 'Configuration' (with a dropdown set to 'Default Configuration for JBoss 7.1 Runtime' and a 'Modify...' button), 'EAR membership' (with an unchecked 'Add project to an EAR' checkbox and a 'EAR project name' field set to 'EAR'), and 'Working sets' (with an unchecked 'Add project to working sets' checkbox and a 'Working sets:' dropdown). At the bottom are buttons for '?', '< Back' (disabled), 'Next >', 'Finish' (highlighted in blue), and 'Cancel'.

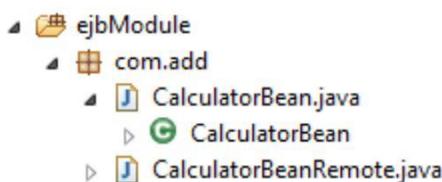


Step 2 :

Now create Stateless session bean with its remote interface. Expand project → expand ejbModule → Right click Session Bean → New → Session

Bean

In ejbModule 2 java files are going to create after Finish button.



Write following code in CalculatorBean.java

```

1 package com.add;
2
3 import javax.ejb.LocalBean;
4
5 /**
6  * Session Bean implementation class CalculatorBean
7  */
8 @Stateless
9 @LocalBean
10 public class CalculatorBean implements CalculatorBeanRemote {
11
12     /**
13      * Default constructor.
14      */
15     public CalculatorBean() {
16         // TODO Auto-generated constructor stub
17     }
18
19     public float add(float a, float b)
20     {
21         return a+b;
22     }
23 }

```

Write Following code in CalculatorBeanRemote.java

```

1 package com.add;
2
3 import javax.ejb.Remote;
4
5 @Remote
6 public interface CalculatorBeanRemote {
7
8     public float add(float a, float b);
9
10 }
11

```

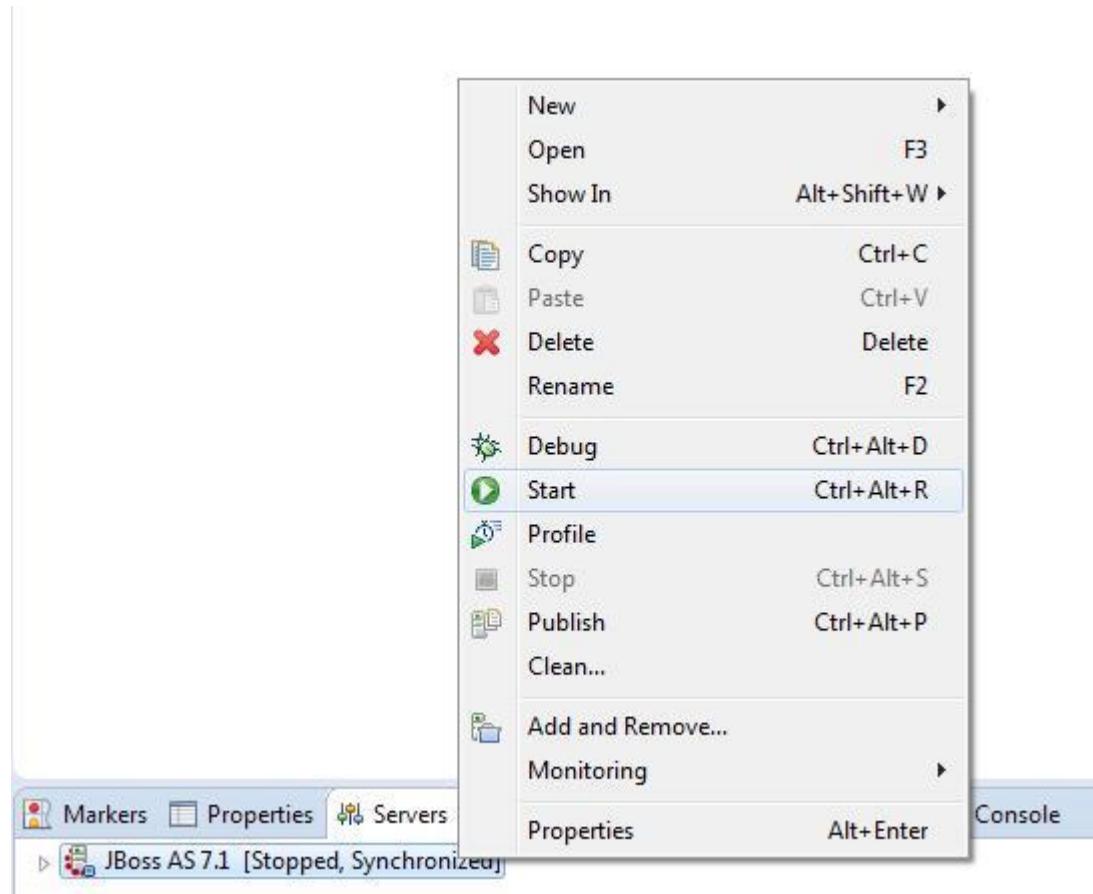
Step 3:

Deploying the project :

Now we need to deploy the our EJB "addition" on server. Follow the steps mentioned bellow to deploy this project on server.

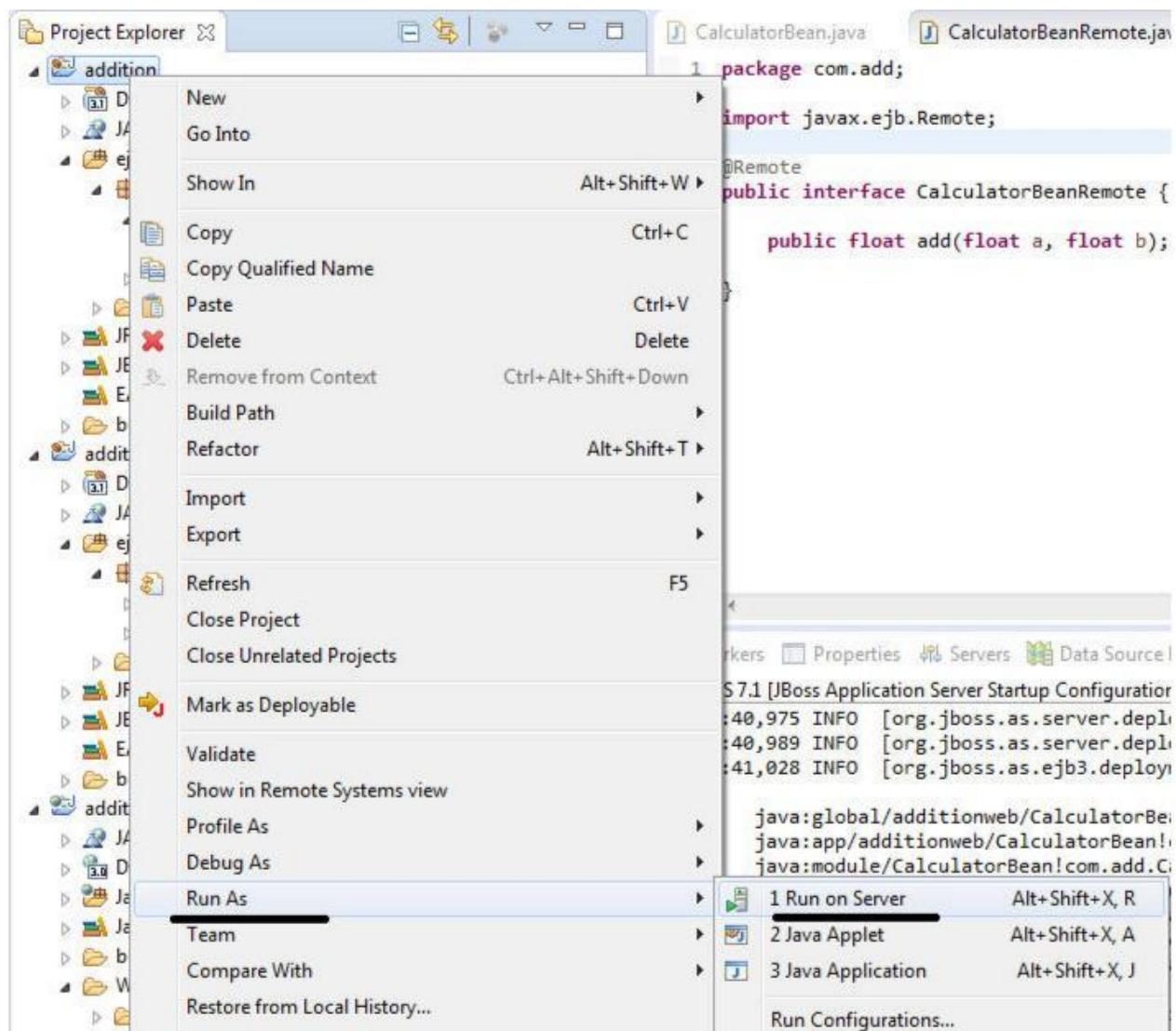
Strat the server

Right click on "JBoss 7.1 Runtime Server" from Servers view and click on Start.



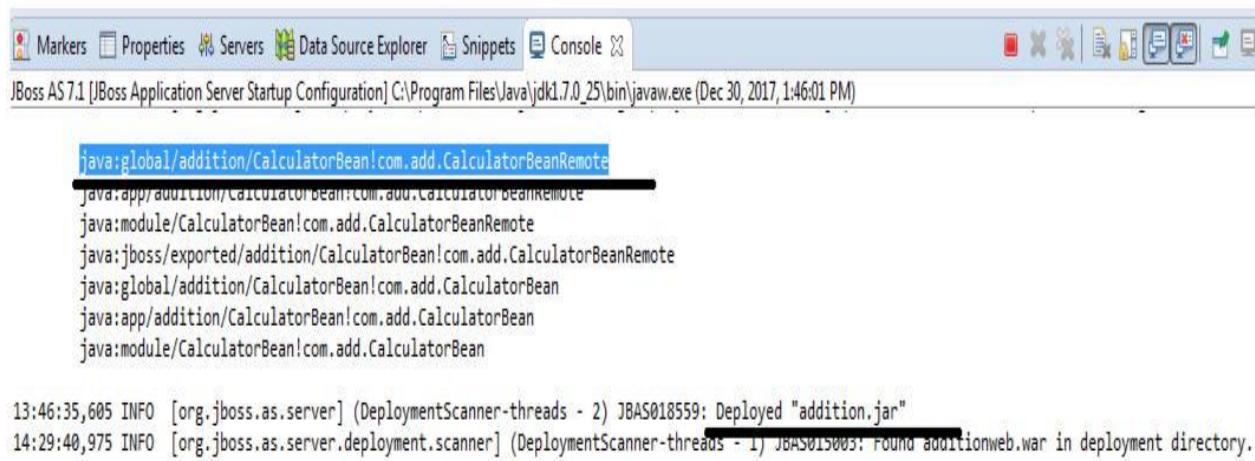
Step 4:

Now next step Go to Project-> addition -> right click -> run-> Run on server



Step 5:

After running the program you can see following message on console



The screenshot shows the JBoss AS 7.1 [JBoss Application Server Startup Configuration] console window. The logs display the deployment process of 'addition.jar' and 'additionweb.war'. Key log entries include:

```

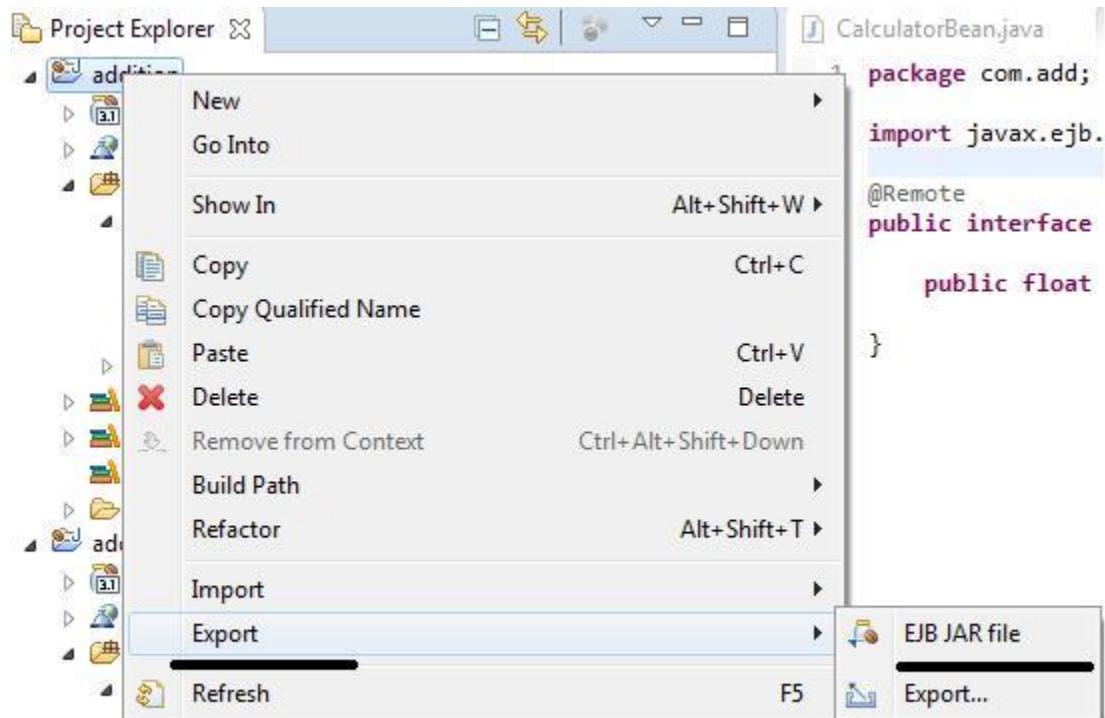
java:global/addition/CalculatorBean!com.add.CalculatorBeanRemote
java:app/addition/CalculatorBean!com.add.CalculatorBeanRemote
java:module/CalculatorBean!com.add.CalculatorBeanRemote
java:jboss/exported/addition/CalculatorBean!com.add.CalculatorBeanRemote
java:global/addition/CalculatorBean!com.add.CalculatorBean
java:app/addition/CalculatorBean!com.add.CalculatorBean
java:module/CalculatorBean!com.add.CalculatorBean

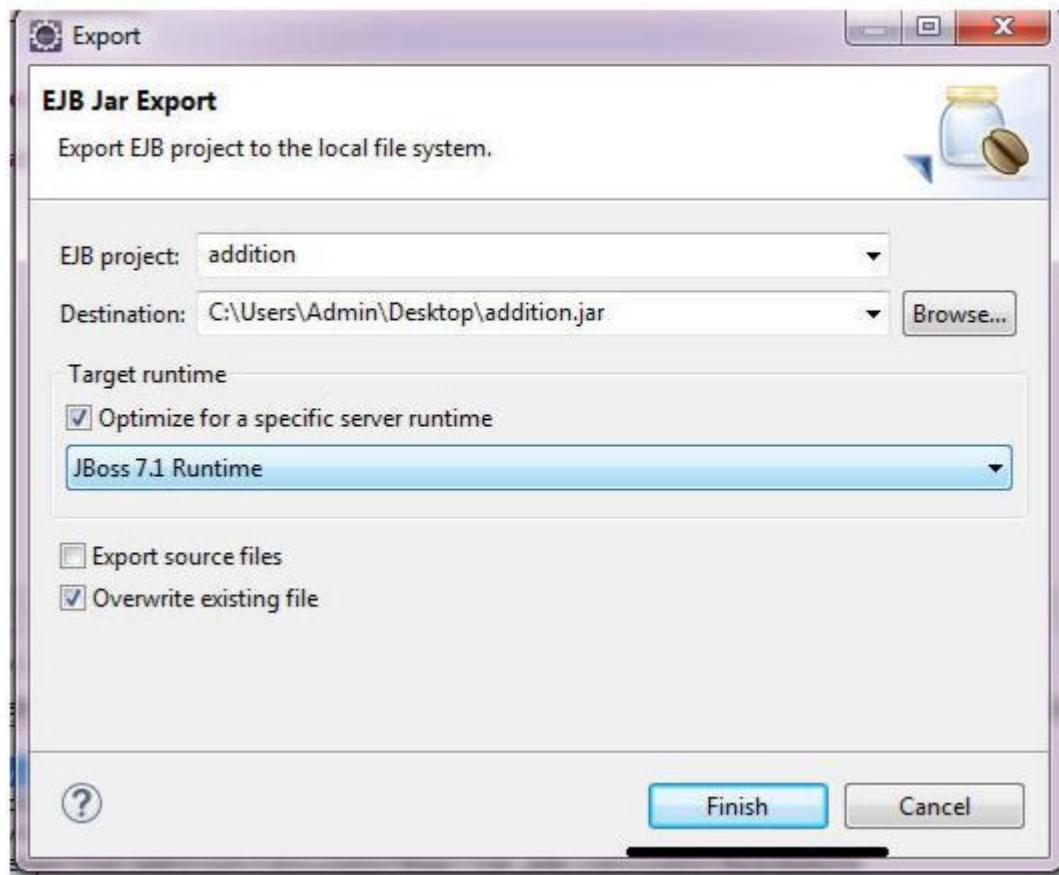
13:46:35,605 INFO [org.jboss.as.server] (DeploymentScanner-threads - 2) JBAS018559: Deployed "addition.jar"
14:29:40,975 INFO [org.jboss.as.server.deployment.scanner] (DeploymentScanner-threads - 1) JBAS015003: Found additionweb.war in deployment directory.

```

Step 6:

Once this jar file is deployed to server now export EJB jar file save it in desktop -> Finish.



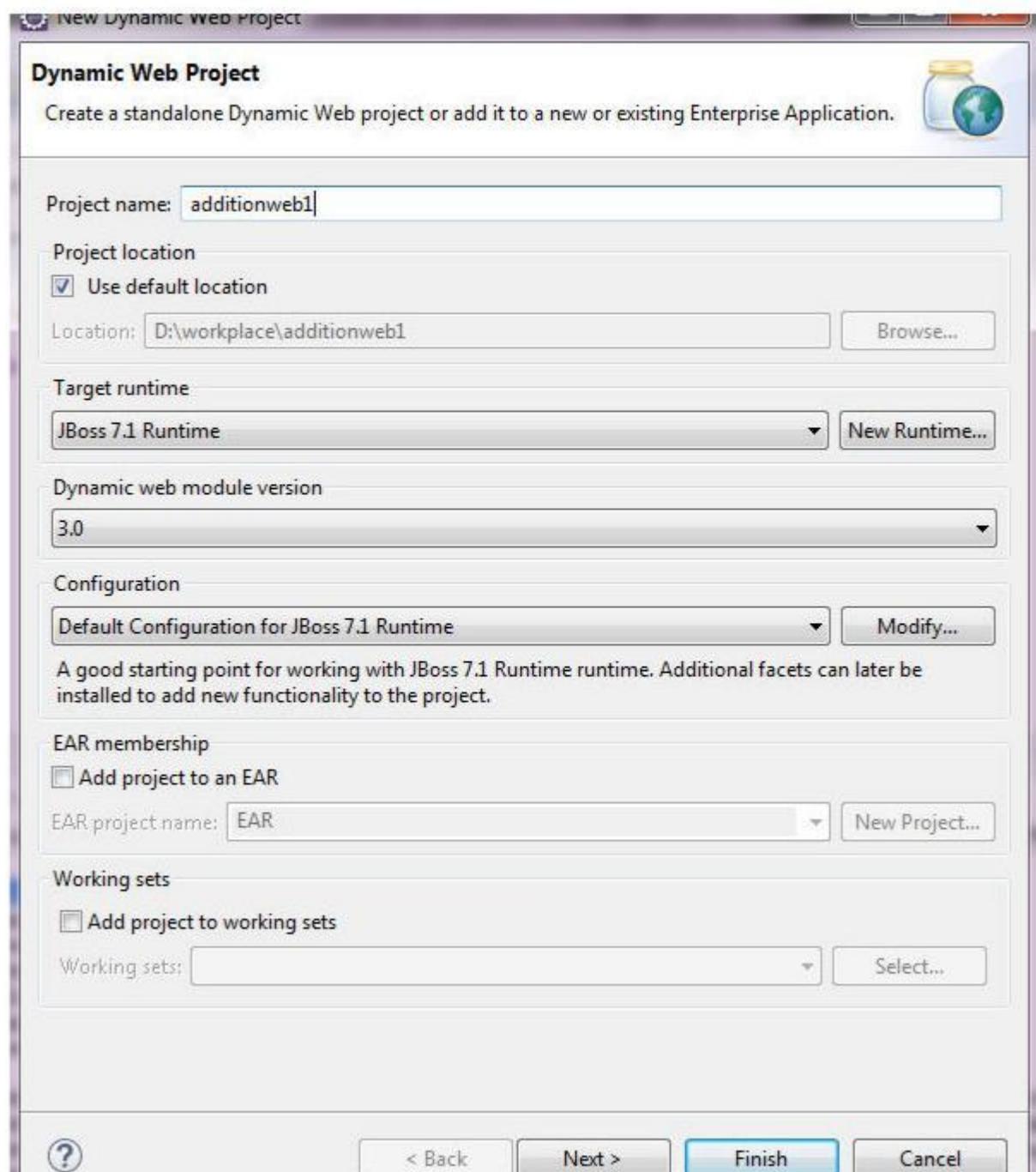


Step 7:

Now create another project



Write project name-> as additionweb -> finish

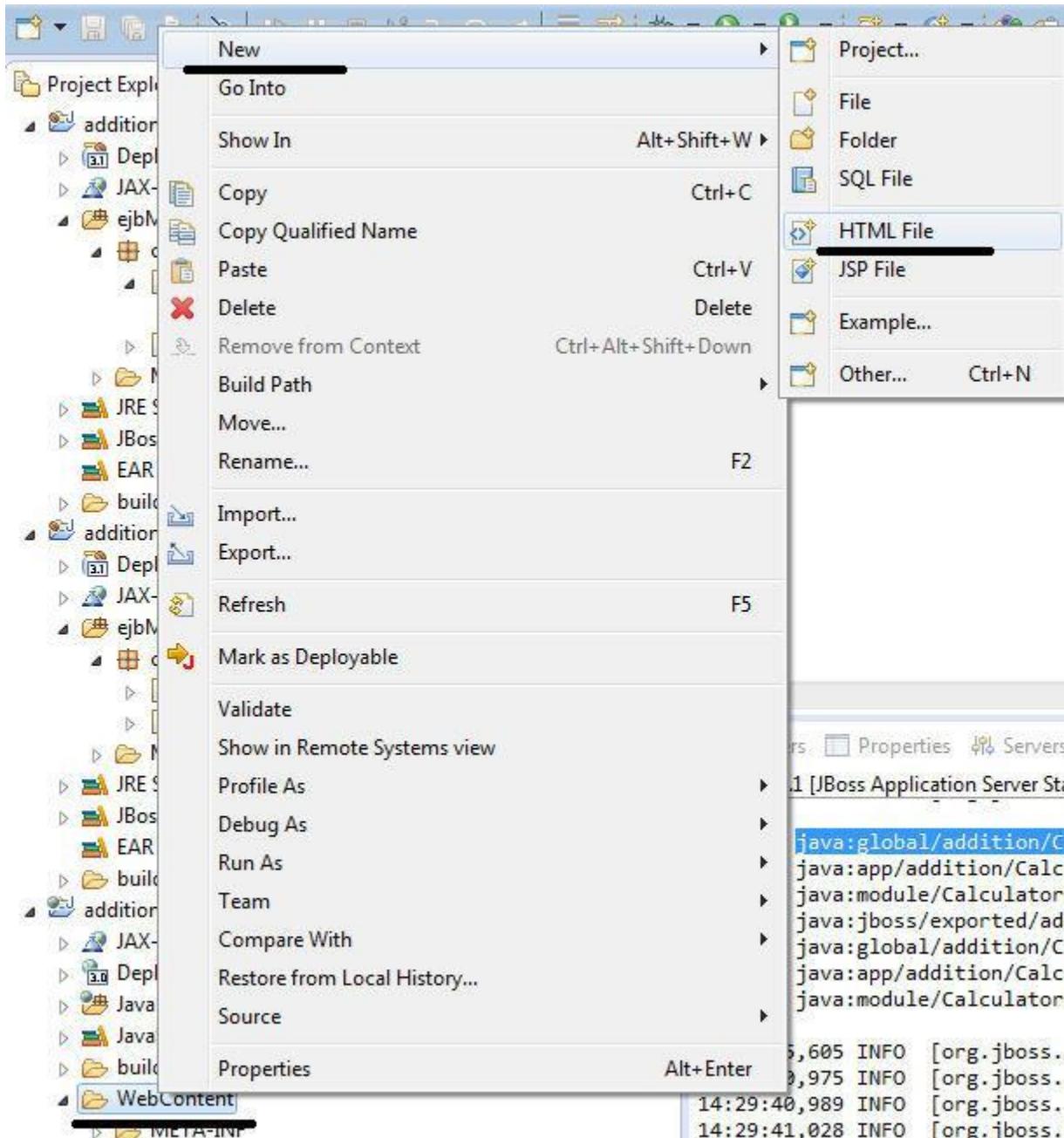


Get a file structure as follow



Step 8:

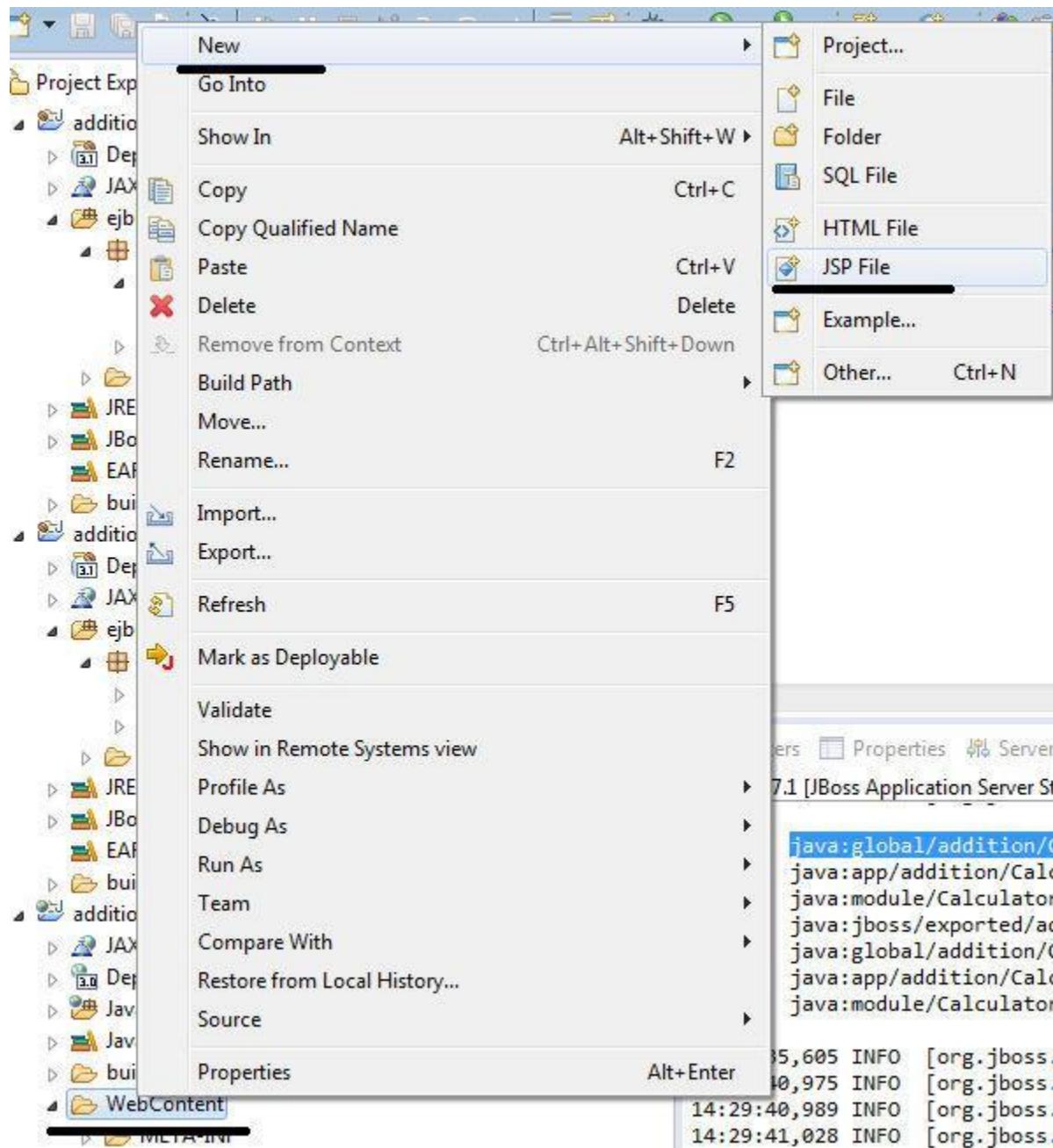
WebContent -> right click->new -> html page



Write file name -> form.html->Finish

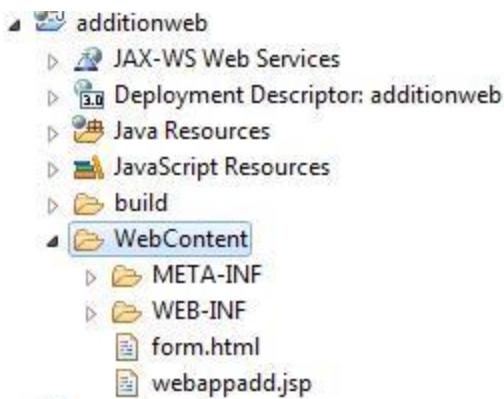
Step 9:

WebContent -> right click->new -> jsp page



Write file name -> webappadd.jsp->Finish

Get the file structure in project window as follows



Write the following code in form.html

//form.html

```

1① <html>
2②   <head>
3     <title>Calculator</title>
4   </head>
5
6③   <body bgcolor="blue">
7     <h1>Calculator</h1>
8     <hr>
9
10④    <form action="webappadd.jsp" method="POST">
11⑤    <p>Enter first value:
12      <input type="text" name="num1" size="25"></p>
13      <br>
14⑥    <p>Enter second value:
15      <input type="text" name="num2" size="25"></p>
16      <br>
17
18      <b>Select your choice:</b><br>
19      <input type="radio" name="group1" value = "add">Addition<br>
20
21⑦    <p>
22      <input type="submit" value="Submit">
23      <input type="reset" value="Reset"></p>
24
25    </form>
26
27
28  </body>
29 </html>
  
```

Write following code in webappadd.jsp

```

CalculatorBean.java   CalculatorBeanRemote.java   webappadd.jsp   form.html   webclient.jsp   http://localhost8080/additionweb/w...   CalculateBean.java
1 <%@ page contentType="text/html; charset=UTF-8" %>
2 <%@ page import="com.add.*, javax.naming.*, javax.ejb.EJB"%>
3
4<%>
5
6 float result=0;
7 // CalculatorBeanRemote calculator=null;
8
9
10 try {
11     InitialContext ic = new InitialContext();
12
13
14     CalculatorBeanRemote calculator = (CalculatorBeanRemote) ic.lookup("java:global/addition/CalculatorBean!com.add.CalculatorBeanRemote");
15
16     System.out.println("Loaded Calculator Bean");
17 //CalculatorBean
18
19
20
21
22     String s1 = request.getParameter("num1");
23     String s2 = request.getParameter("num2");
24     String s3 = request.getParameter("group1");
25
26 System.out.println(s3);
27
28     if ( s1 != null && s2 != null ) {
29         Float num1 = new Float(s1);
30         Float num2 = new Float(s2);
31
32         if(s3.equals("add"))
33             result=calculator.add(num1.floatValue(),num2.floatValue());
34     }
35
36
37 %>
<p>

38     <b>The result is:</b> <%= result %>
39     <p>
40
41<%>
42     }
43     }// end of try
44     catch (Exception e) {
45     e.printStackTrace ();
46     //result = "Not valid";
47 }
48
49 %>

```

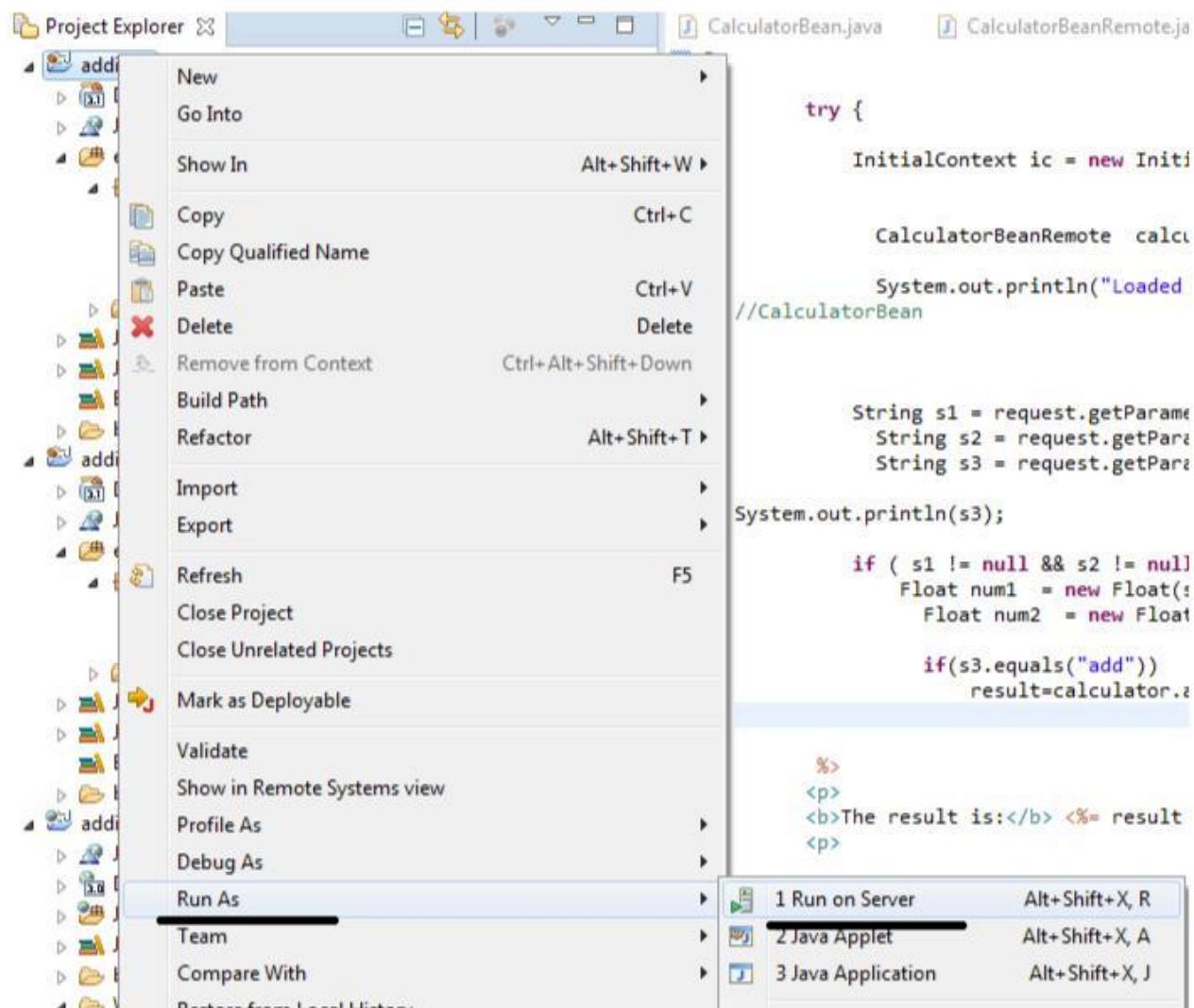
Step 10 :

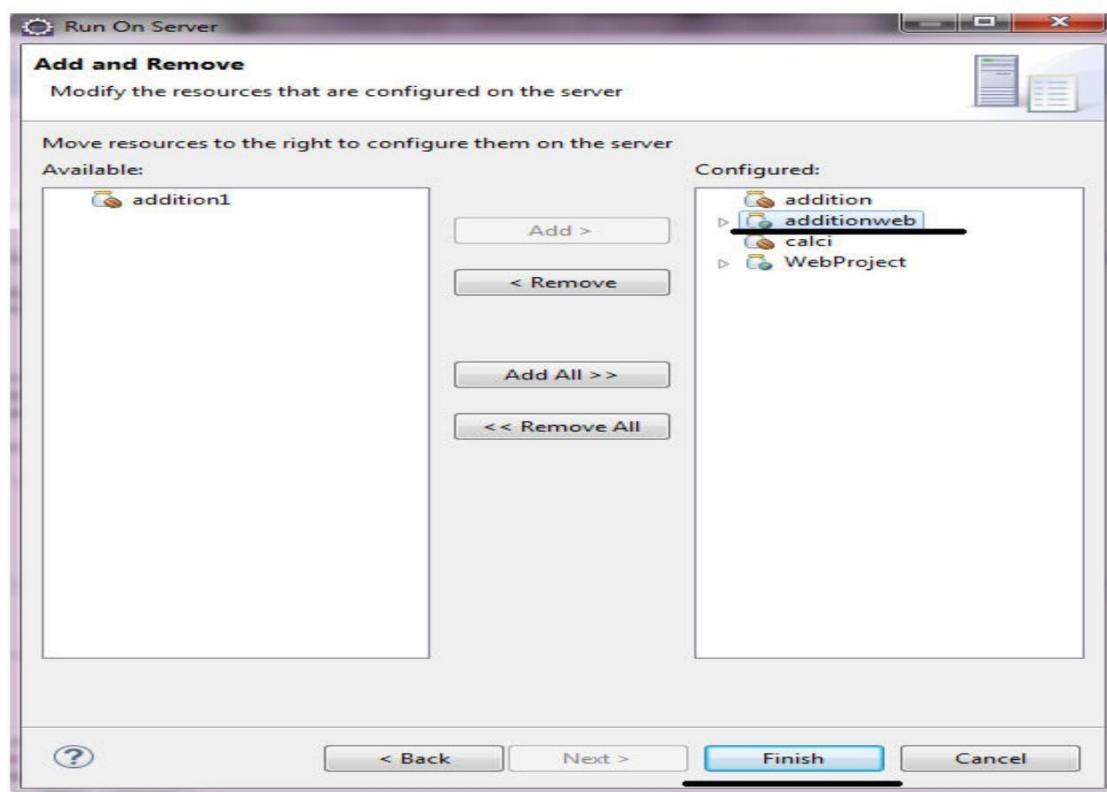
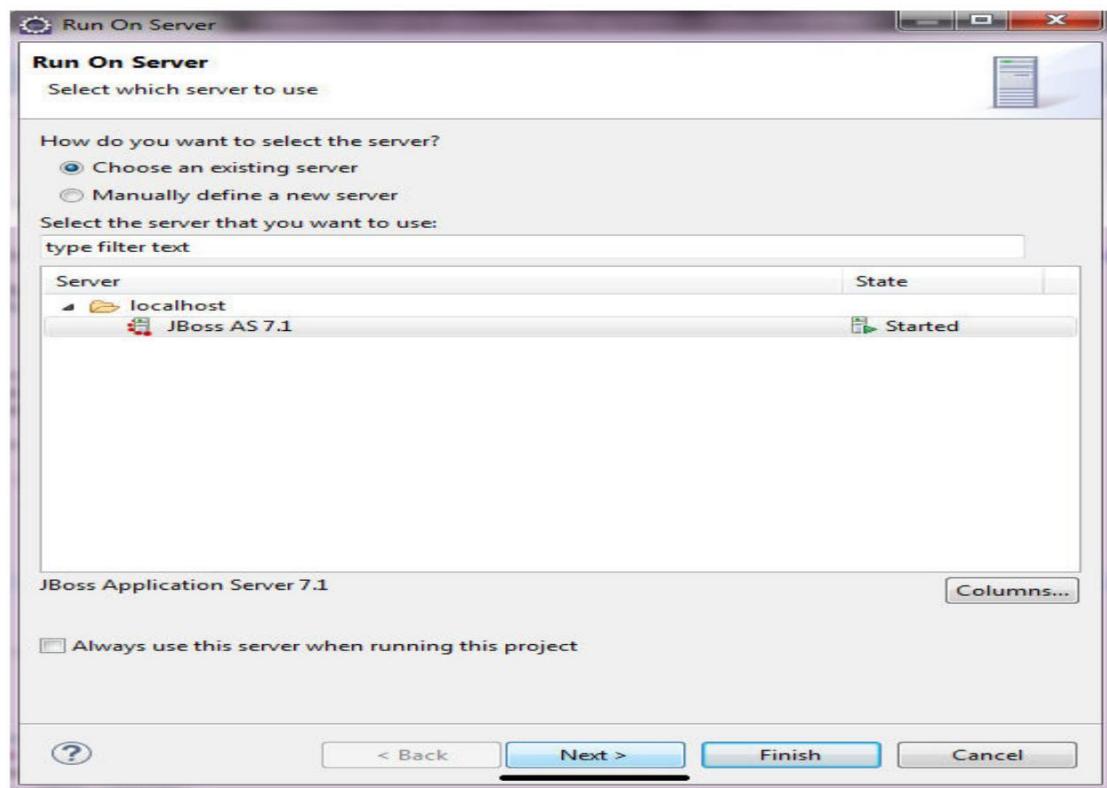
Copy the url from step 5 and add that url in wepappadd code as given above.

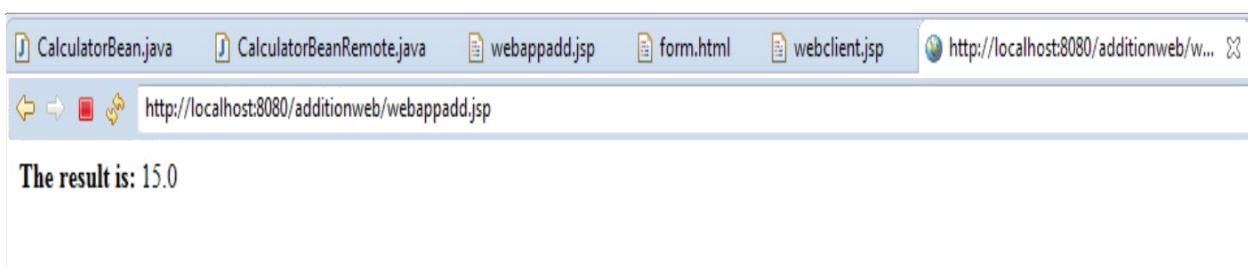
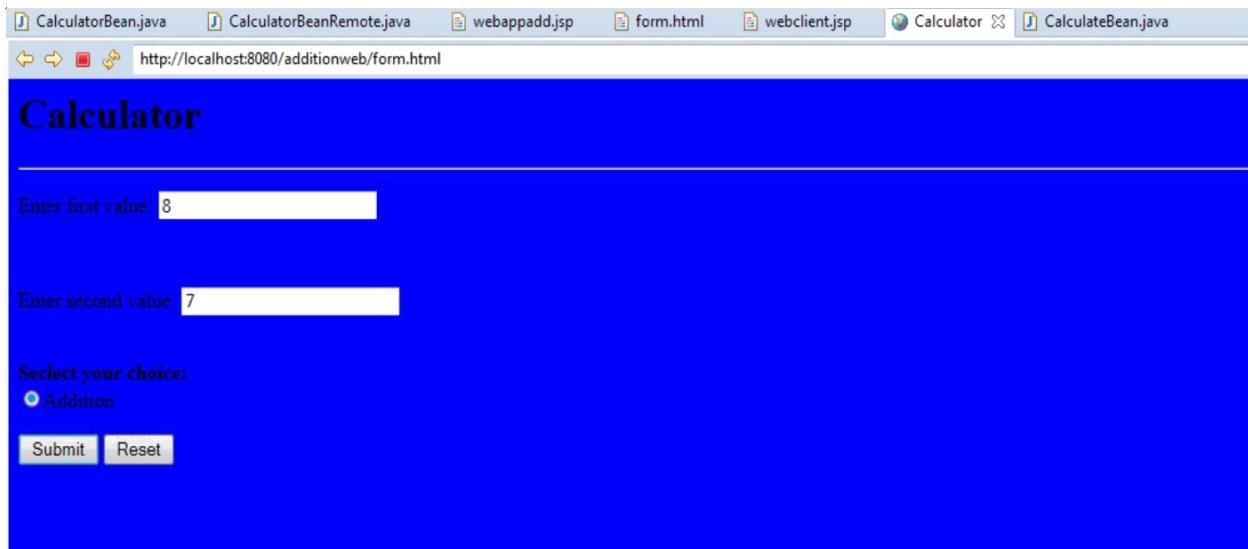
Step 11 :

Running the application :

Right click on project addition-> run as -> run on server







ORAL QUESTIONS

1. What is EJB?
2. What is JSP?
3. What is the purpose of JBOSS?
4. What is the syntax of JSP?
5. How to deploy java beans to server?