SRTTC's Suman Ramesh Tulsiani Technical Campus - Faculty of Engineering

DEPARTMENT OF COMPUTER ENGINEERING

Course 310257

Web Technology Laboratory

DEPARTMENT OF COMPUTER ENGINEERING

Laboratory manual

Course 310257

Web Technology

Prepared by: Prof.V.Y.Bhamare

SRTCT's Suman Ramesh Tulsiani Technical Campus

Faculty of Engineering

Old Mumbai - Pune Hwy, Khamshet

Pune,Maharashtra 410405

DEPARTMENT OF COMPUTER ENGINEERING

Certificate

This is to certify that the Laboratory Manual entitled

Web Technology

Submitted by

(Candidate Name) (Seat no.)

Is approved by Prof. V.Y.Bhamare for submission. It is certified further that, to the best ofmy knowledge, the practical's represents work carried out by my students as the fulfillment for T.E. Computer Engineering (Semester II) Web Technology Work as prescribed by the University of Pune for the academic year 2024-25

[Prof. V.Y.Bhamare]	**[Prof.A.V.Surana]**	**Prof. (Dr.) J. B.Sankpal**
Subject Coordinator	Head of Department	**Principal, SRTTC-FoE**

INDEX

Sr. No.	Title of Assignment	Date of Conduction	Date of checking	Page No.	Faculty Signature
1	Case study: Before coding of the website, planning is important, students should visit different websites (Min. 5) for the different client projects and note down the evaluation results for these websites, either good website or bad website				
2	Implement a web page index.htm for any client website (e.g., a restaurant website project) using following: a. HTML syntax: heading tags, basic tags and attributes, frames, tables, images, lists, links for text and images, forms etc. b. Use of Internal CSS, Inline CSS, External CSS				
3	Design the XML document to store the information of the employees of any business organization and demonstrate the use of: a) DTD b) XML Schema And display the content in (e.g., tabular format) by using CSS/XSL.				
4	Implement an application in Java Script using following: a) Design UI of application using HTML, CSS etc. b) Include Java script validation c) Use of prompt and alert window using Java Script e.g., Design and implement a simple calculator using Java Script for operations like addition, multiplication, subtraction, division, square of number etc. a) Design calculator interface like text field for input and output, buttons for numbers and operators etc. b) Validate input values c) Prompt/alerts for invalid values etc				
5	Implement the sample program demonstrating the use of Servlet. e.g., Create a database table ebookshop (book_id, book_title, book_author, book_price, quantity) using database like Oracle/MySQL etc. and display (use SQL select query) the table content using servlet.				
6	Implement the program demonstrating the use of JSP. e.g., Create a database table students_info (stud_id, stud_name, class, division, city) using database like				

	Oracle/MySQL etc. and display (use SQL select query) the table content using JSP.				
7	Build a dynamic web application using PHP and MySQL. a. Create database tables in MySQL and create connection with PHP. b. Create the add, update, delete and retrieve functions in the PHP web app interacting with MySQL database				
8	Design a login page with entries for name, mobile number email id and login button. Use struts and perform following validations a. Validation for correct names b. Validation for mobile numbers c. Validation for email id d. Validation if no entered any value e. Re-display for wrongly entered values with message f. Congratulations and welcome page upon successful entries				
9	Design an application using Angular JS. e.g., Design registration (first name, last name, username and password) and login page using Angular JS.				
10	Design and implement a business interface with necessary business logic for any web application using EJB. e.g., Design and implement the web application logic for deposit and withdraw amount transactions using EJB.				
11	**Mini Project**: Design and implement a dynamic web application for any business functionality by using web development technologies that you have learnt in the above given assignments.				

Assignment No. : 1

TECHNNOLOGY

Case study: Before coding of the website, planning is important

OBJECTIVES

1. Understand how to develop web application
2. before coding you do the planning of web application

PROBLEM STATEMENT

check the performance of different (min. 5 web site) on the basis of speed page loading, content etc

Sr. No.	Website URL	Purpose of Website	Things liked in the website	Things disliked in the website	Overall evaluation of the website (Good/Bad)

use this format of analysis the performance of web application

OUTCOME

Students will be able to, planned and develop good web application considering all facts.

SOFTWARE & HARDWARE REQUIREMENTS

1. normal browser

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,

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,

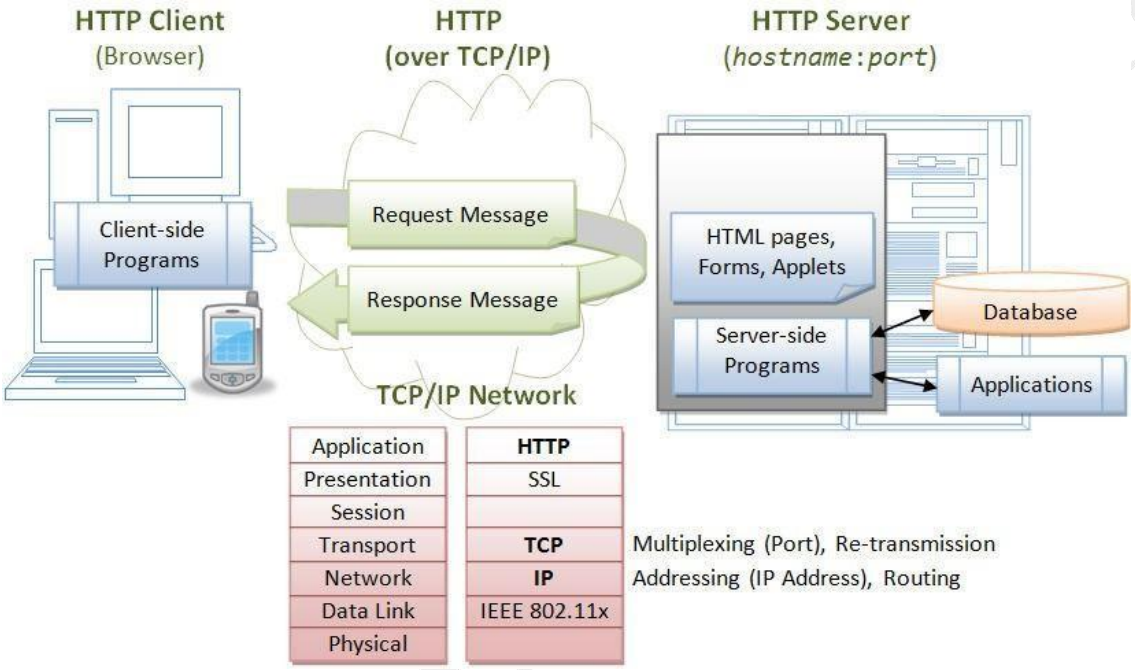


Figure.1: Three-Tier Client-Server Architecture

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.

EXECUTION STEPS

Note:-do it for all web site(Minimum 5)

Sr. No.	Website URL	Purpose of Website	Things liked in the website	Things disliked in the website	Overall evaluation of the website (Good/Bad)

1. Go to any website: like: www.sit.com
2. Mention the purpose
3. What you like on that web site, animation graphics etc
4. mention things which you don't like
5. on the basis of these points check the performance

Assignment 2

Title: HTML, CSS, XML

Objectives:-

- Understand about basic concepts of html
- Understand the basic concepts of XML
- Understand the basic concepts of CSS

Problem Statement : - Implement a web page index.htm for any client using following:
a. HTML syntax: heading tags, basic tags and attributes, frames, tables, images, lists, links for text and images, forms etc.
b. Use of Internal CSS, Inline CSS, External CSS
S/W Required : CPP

Theory Concept:-

- HTML stands for Hyper Text Markup Language
- HTML describes the structure of Web pages using markup
- 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 elements are the building blocks of HTML pages
- HTML elements are represented by tags

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

Program:-

```
<!DOCTYPE html>

<html>

<head>
	<title>Best Learning Platform</title>

	<style>
		* {
			margin: 0;
			padding: 0;
		}

		.navbar {
			display: flex;
			align-items: center;
			justify-content: center;
			position: sticky;
			top: 0;
			cursor: pointer;
		}

		.background {
			background: black;
			background-blend-mode: darken;
			background-size: cover;
		}

		.nav-list {
			width: 70%;
			display: flex;
			align-items: center;
		}

		.logo {
			display: flex;
			justify-content: center;
			align-items: center;
```

```
}

.logo img {
	width: 180px;
	border-radius: 50px;
}

.nav-list li {
	list-style: none;
	padding: 26px 30px;
}

.nav-list li a {
	text-decoration: none;
	color: white;
}

.nav-list li a:hover {
	color: grey;
}

.rightnav {
	width: 30%;
	text-align: right;
}

#search {
	padding: 5px;
	font-size: 17px;
	border: 2px solid grey;
	border-radius: 9px;
}

.firstsection {
	background-color: #131414d6;
	height: 400px;
	color: white;
}

.secondsection {
	background-color: #ff862f;
	height: 400px;
}

.box-main {
	display: flex;
	justify-content: center;
	align-items: center;
	color: white;
	max-width: 80%;
	margin: auto;
	height: 80%;
}
```

```
.firsthalf {
	width: 100%;
	display: flex;
	flex-direction: column;
	justify-content: center;
}

.secondhalf {
	width: 30%;
}

.secondhalf img {
	width: 70%;
	border: 4px solid white;
	border-radius: 150px;
	display: block;
	margin: auto;
}

.text-big {
	font-family: 'Piazzolla', serif;
	font-weight: bold;
	font-size: 35px;
}

.text-small {
	font-size: 18px;
}

.btn {
	padding: 8px 20px;
	margin: 7px 0;
	border: 2px solid white;
	border-radius: 8px;
	background: none;
	color: white;
	cursor: pointer;
}

.btn-sm {
	padding: 6px 10px;
	vertical-align: middle;
}

.section {
	height: 400px;
	display: flex;
	align-items: center;
	justify-content: center;
	max-width: 90%;
	margin: auto;
}

.section-Left {
```

```
                flex-direction: row-reverse;
            }

            .paras {
                padding: 0px 65px;
            }

            .thumbnail img {
                width: 250px;
                border: 2px solid black;
                border-radius: 26px;
                margin-top: 19px;
            }

            .center {
                text-align: center;
            }

            .text-footer {
                text-align: center;
                padding: 30px 0;
                font-family: 'Ubuntu', sans-serif;
                display: flex;
                justify-content: center;
                color: white;
            }
        </style>
</head>

<body>
    <nav class="navbar background">
        <ul class="nav-list">
            <div class="logo">
                <img src= "E:/SRTTC DATA/WT notes and
prac/Practicals/edulogo.png"  style="height: 35px; width: 35px;">
            </div>
            <li><a href="#web">Web Technology</a></li>
            <li><a href="#program">C Programming</a></li>
            <li><a href="#course">Other Courses</a></li>
        </ul>

        <div class="rightNav">
            <input type="text" name="search" id="search">
            <button class="btn btn-sm">Search</button>
        </div>
    </nav>

    <section class="firstsection">
        <div class="box-main">
            <div class="firstHalf">
                <h1 class="text-big" id="web">Web Technology</h1><br>
                <p class="text-small">
                    HTML stands for HyperText Markup Language.
                    It is used to design web pages using a markup
```

```
                    language. HTML is the combination of Hypertext
                    and Markup language. Hypertext defines the
                    link between the web pages. A markup language
                    is used to define the text document within tag
                    which defines the structure of web pages.
                    HTML is a markup language that is used by the
                    browser to manipulate text, images, and other
                    content to display it in the required format.
                </p>


            </div>
        </div>
</section>

 <section class="secondsection">
        <div class="box-main">
            <div class="firstHalf">
                <h1 class="text-big" id="program">
                    C Programming
                </h1><br>
                <p class="text-small">
                    C is a procedural programming language. It
                    was initially developed by Dennis Ritchie
                    as a system programming language to write
                    operating system. The main features of C
                    language include low-level access to memory,
                    simple set of keywords, and clean style,
                    these features make C language suitable for
                    system programming like operating system or
                    compiler development.
                </p>


            </div>
        </div>
</section>


        <section class="firstsection">
        <div class="box-main">
            <div class="firstHalf">
                <h1 class="text-big" id="web">Java</h1><br>
                <p class="text-small">
                    Java has been one of the most popular
                programming language for many years.
                Java is Object Oriented. However it is
                not considered as pure object oriented
                as it provides support for primitive
                data types (like int, char, etc) The
                Java codes are first compiled into byte
                code (machine independent code). Then
                the byte code is run on Java Virtual
                Machine (JVM) regardless of the
```

```
                                underlying architecture.
                                </p>


                        </div>
                </div>
        </section>



        <footer class="background">
                <p class="text-footer">
                        Copyright ©-All rights are reserved
                </p>


        </footer>
</body>

</html>
```

Output : -

Conclusion: -

Assignment 3

Title: XML and CSS

PROBLEM STATEMENT

Write a program to design book catalog by using XML and CSS to display tile, 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 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

- XML is extensible XML allows you to create your own self-descriptive tags, orlanguage, that suits your application
- XML carries the data, does not present it **XML allows you to store the datairrespective** 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>**……**</el

ement> 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

XMLTutorial

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

TEST CASES

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

PROGRAM CODE & OUTPUT

Emp.xml :-

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE note SYSTEM "note.dtd">
<?xml-stylesheet type="text/xsl" href="rule.xsl"?>
<employee>
<details>
<empId>101</empId>
<name>Pramod Kumar</name>
<age>45</age>
<post>Manager</post>
<salary>60000</salary>
</details>
<details>
<empId>102</empId>
<name>Rakesh Chauhan</name>
<age>30</age>
<post>HR</post>
<salary>50000</salary>
</details>
<details>
<empId>103</empId>
<name>Vishal Anand</name>
<age>34</age>
<post>Library Clerk</post>
<salary>35000</salary>
</details>
<details>
<empId>104</empId>
<name>Anil Dixit</name>
<age>30</age>
<post>Developer</post>
<salary>45000</salary>
</details>
<details>
<empId>105</empId>
<name>Raj Bahadur</name>
<age>26</age>
<post>Marketing</post>
<salary>30000</salary>
</details>
        </employee>
```

Rule.css

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<body>
<h1 align="center">Employee Detail Table </h1>
<table border="3" align="center">
<tr>
<th>empId</th>
<th>Name</th>
<th>Age</th>
<th>Post</th>
<th>Salary</th>
</tr>
<xsl:for-each select="employee/details">
<tr>
<td><xsl:value-of select="empId"/></td>
<td><xsl:value-of select="name"/></td>
<td><xsl:value-of select="age"/></td>
<td><xsl:value-of select="post"/></td>
<td><xsl:value-of select="salary"/></td>
</tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Note.dtd:-

```
<!ELEMENT employee (details)>
<!ELEMENT details (empId,name,age,post,salary)>
<!ELEMENT empId (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT age (#PCDATA)>
<!ELEMENT post (#PCDATA)>
<!ELEMENT salary (#PCDATA)>
```

Output:-

Employee Detail Table

empId	Name	Age	Post	Salary
101	Pramod Kumar	45	Manager	60000
102	Rakesh Chauhan	30	HR	50000
103	Vishal Anand	34	Library Clerk	35000
104	Anil Dixit	30	Developer	45000
105	Raj Bahadur	26	Marketing	30000

CONCLUSION/ANALYSIS

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

Assignment 4

Title: HTML ,CSS AND JAVASCRIPT

PROBLEM STATEMENT

Design and implement a simple calculator using Java Script for operations like addition, multiplication, subtraction, division, square of number etc.
a) Design calculator interface like text field for input and output, buttons for numbers and operators etc.
b) Validate input values
c) Prompt/alerts for invalid values etc.

OUTCOMES

Students will be able to,

1. Design calculator using javascript
2. Understand how to use javascript and apply CSS

SOFTWARE & HARDWARE REQUIREMENTS

Software: Notepad, Any Browser

THEORY-CONCEPT

JavaScript (js) is a light-weight object-oriented programming language which is used by several websites for scripting the webpages. It is an interpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document. It was introduced in the year 1995 for adding programs to the webpages in the Netscape Navigator browser. Since then, it has been adopted by all other graphical web browsers. With JavaScript, users can build modern web applications to interact directly without reloading the page every time.

Features of JavaScript

There are following features of JavaScript:

1. All popular web browsers support JavaScript as they provide built-in execution environments.

2. JavaScript follows the syntax and structure of the C programming language. Thus, it is a structured programming language.
3. JavaScript is a weakly typed language, where certain types are implicitly cast (depending on the operation).
4. JavaScript is an object-oriented programming language that uses prototypes rather than using classes for inheritance.
5. It is a light-weighted and interpreted language.
6. It is a case-sensitive language.
7. JavaScript is supportable in several operating systems including, Windows, macOS, etc.
8. It provides good control to the users over the web browsers.

Application of JavaScript

JavaScript is used to create interactive websites. It is mainly used for:

- Client-side validation,
- Dynamic drop-down menus,
- Displaying date and time,
- Displaying pop-up windows and dialog boxes (like an alert dialog box, confirm dialog box and prompt dialog box),
- Displaying clocks etc.

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. Write the script code in notepad and save with .js extension.
4. Import CSS files and js in html page.
5. Open page in the browser

TEST CASES

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

PROGRAM CODE & OUTPUT

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width">
<title>Calculate me! - A calculator made my me</title>
<link href="style.css" rel="stylesheet" type="text/css" />
<link href="util.css" rel="stylesheet" type="text/css" />
</head>
<body>
<h1 class="text-center">Welcome to Calculate me!</h1>
<div class="box">
<div class="container flex flex-col items-center mx-auto m-w-20">
<div class="row">
<input class="input" type="text"/>
</div>
<div class="row">
<button class="button">C</button>
<button class="button">%</button>
<button class="button">M+</button>
<button class="button">M-</button>
</div>
<div class="row">
<button class="button">7</button>
<button class="button">8</button>
<button class="button">9</button>
<button class="button">*</button>
</div>
<div class="row">
<button class="button">4</button>
<button class="button">5</button>
<button class="button">6</button>
<button class="button">/</button>
</div>
<div class="row">
<button class="button">1</button>
<button class="button">2</button>
<button class="button">3</button>
<button class="button">+</button>
</div>
<div class="row">
<button class="button">0</button>
```

```
<button class="button">.</button>
<button class="button">=</button>
<button class="button">-</button>
</div>
</div>
</div>
<script src="script.js"></script>
</body>
</html>
```

style.css :-

```
@import
url('https://fonts.googleapis.com/css2?family=Roboto:wght@300&family=Ubuntu:wght@30
0&display=swap');
html, body
{ height:
100%;
width: 100%;
font-family: 'Roboto', sans-serif;
background-color: black;
}
.box {
margin: 45px 580px;
width: 300px;
min-height: 399px;
background-color: white;
border-radius: 5px;
padding: 15px;
background-color: slategrey;
}
h1{
margin: 30px 557px;
font-family: fangsong;
font-size: 35px;
color: darkcyan;
}
.button{  widt
h: 66px;
padding: 20px;
margin: 0 3px;
border: 2px solid black;
border-radius: 9px;
cursor: pointer;
font-weight: 800;
font-size: 18px;
}
```

```
.row{
margin: 8px 0;
}
.row
input{ width:
291px; font-
size: 20px;
margin: 0;
padding: 10px 0px;
border: 2px solid black;
border-radius: 5px;
}
```

Util.css:

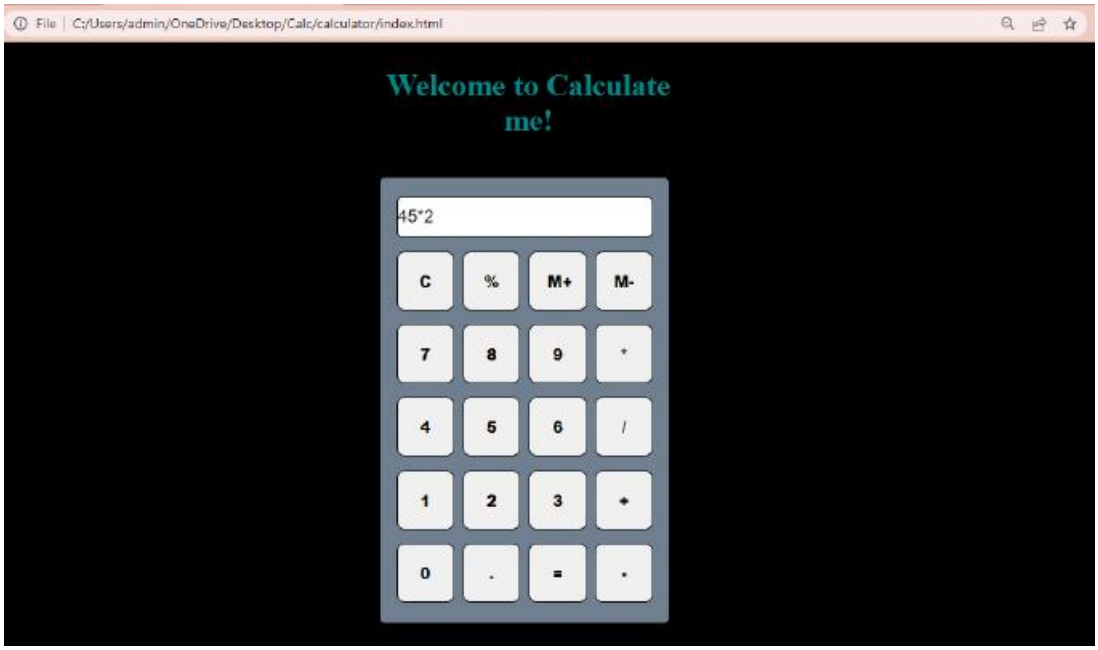
```
.text-center{
text-align: center; }
.bg-
red{ background:
red;}
.mx-
auto{ margin:
auto;}
.flex{ display:
flex;}
.flex-col{
flex-direction: column;}
.items-
center{ align-items:
center;
}
```

Script.js:

```
// Todo: Make M+ M- and MC functional
let string = "";
let buttons = document.querySelectorAll('.button');
Array.from(buttons).forEach((button)=>{ button.addE
ventListener('click', (e)=>{ if(e.target.innerHTML ==
'='){
string = eval(string);
document.querySelector('input').value = string;
}
else if(e.target.innerHTML ==
'C'){ string = ""
document.querySelector('input').value = string;
}
```

```
else{ console.log(e.tar
get)
string = string + e.target.innerHTML;
document.querySelector('input').value = string; } }) })
```

Output:-



CONCLUSION/ANALYSIS

Hence, we have designed and implemented a simple calculator using Java Script

Assignment 5

Title: Servlet

PROBLEM STATEMENT

Implement the sample program demonstrating the use of Servlet.

e.g., Create a database table ebookshop (book_id, book_title, book_author, book_price, quantity) using database like Oracle/MySQL etc. and display (use SQL select query) the table content using servlet.

OUTCOMES

Students will be able to,

1. Design webpage using Servlet .

SOFTWARE & HARDWARE REQUIREMENTS

Software: Eclipse ,SQL, Any Browser

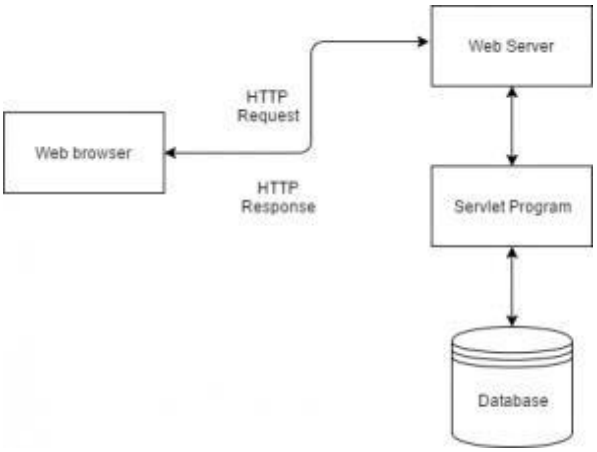
THEORY-CONCEPT

Servlets are the Java programs that run on the Java-enabled web server or application server. They are used to handle the request obtained from the webserver, process the request, produce the response, then send a response back to the webserver.

Properties of Servlets are as follows:

- Servlets work on the server-side.
- Servlets are capable of handling complex requests obtained from the webserver.

Servlet Architecture is can be depicted from the image itself as provided below as follows:



DESIGN/EXECUTION STEPS

1. The clients send the request to the webserver.
2. The web server receives the request.
3. The web server passes the request to the corresponding servlet.
4. The servlet processes the request and generates the response in the form of output.
5. The servlet sends the response back to the webserver.
6. The web server sends the response back to the client and the client browser displays it on the screen.

TEST CASES

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

PROGRAM CODE & OUTPUT

Servlet1.java :-

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
public class Servlet1 extends HttpServlet
{
public void doGet(HttpServletRequest req, HttpServletResponse res) throws IOException,
ServletException
{
PrintWriter out = res.getWriter();
res.setContentType("text/html");
out.println("<html><body>");
try
{
Class.forName("com.mysql.jdbc.Driver");
Connection con = DriverManager.getConnection("jdbc:mysql:///ebookshop","root","aarati");
// Here dsnname- mydsn,user id- system(for oracle 10g),password is pintu.
```

```
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery("select * from book");
out.println("<table border=1 width=50% height=50%>");
out.println("<tr><th>book_id</th><th>book_title</th><th>book_author</th><th>book_price</t
h><th>book_quantity</th><tr>");
while (rs.next())
{ int i = rs.getInt("book_id");
String t = rs.getString("book_title");
String nm = rs.getString("book_author");
int s = rs.getInt("book_price");
int q= rs.getInt("book_quantity");
out.println("<tr><td>" + i + "</td><td>" + t + "</td><td>" + nm + "</td><td>" + s +
"</td><td>" + q + "</td></tr>");
}
out.println("</table>");
out.println("</html></body>");
con.close();
}
catch (Exception e)
{
out.println("error");
}
}
}
```

Database :-

```
mysql> create database ebookshop;
Query OK, 1 row affected (0.02 sec)
mysql> create table book(book_id int,book_title varchar(30),book_author varchar(30),book_price
int,book_quantity int);
Query OK, 0 rows affected (0.06 sec)
mysql> insert into book(book_id,book_title,book_author,book_price ,book_quantity)
-> values(102,'Vahana Masterclass',’Alfredo Covlli’,1000,10);
Query OK, 1 row affected (0.01 sec)
mysql> insert into book(book_id,book_title,book_author,book_price ,book_quantity)
```

-> values(101,'A Passage to India','E.M. Foster',100,20);

Query OK, 1 row affected (0.01 sec)

mysql> insert into book(book_id,book_title,book_author,book_price ,book_quantity)

-> values(103,'A Revenue Stamp','Amrita Pritam', 200,30);

Query OK, 1 row affected (0.01 sec)

mysql> insert into book(book_id,book_title,book_author,book_price ,book_quantity)

-> values(104,'Pinjar Amrita',' Pritam',250, 15);

Query OK, 1 row affected (0.01 sec)

mysql> insert into book(book_id,book_title,book_author,book_price ,book_quantity)

-> values(105,'A Suitable Boy','Vikram Seth',350, 15);

Query OK, 1 row affected (0.01 sec)

Output:-

book_id	book_title	book_author	book_price	book_quantity
102	Vahana Masterclass	Alfredo Covlli	1000	10
101	A Passage to India	E.M. Foster	100	20
103	A Revenue Stamp	Amrita Pritam	200	30
104	Pinjar	Amrita Pritam	250	15
105	A Suitable Boy	Vikram Seth	350	15

CONCLUSION/ANALYSIS

Hence, we have Implemented the sample program demonstrating the use of Servlet.

Assignment 6

Title: Servlet

PROBLEM STATEMENT

Implement the program demonstrating the use of JSP.

e.g., Create a database table students_info (stud_id, stud_name, class, division, city) using database like Oracle/MySQL etc. and display (use SQL select query) the table content using JSP.

OUTCOMES

Students will be able to,

1. Design webpage using JSP.

SOFTWARE & HARDWARE REQUIREMENTS

Software: Eclipse, Mysql , Any Browser

THEORY-CONCEPT

Introduction

- It stands for Java Server Pages.
- It is a server side technology.
- It is used for creating web application.
- It is used to create dynamic web content.
- In this JSP tags are used to insert JAVA code into HTML pages.
- It is an advanced version of Servlet Technology.
- It is a Web based technology helps us to create dynamic and platform-independent web pages.
- In this, Java code can be inserted in HTML/ XML pages or both.
- JSP is first converted into servlet by JSP container before processing the client's request.

JSP pages are more advantageous than Servlet:

- They are easy to maintain.
- No recompilation or redeployment is required.
- JSP has access to entire API of JAVA .
- JSP are extended version of Servlet.

Features of JSP

- Coding in JSP is easy :- As it is just adding JAVA code to HTML/XML.
- Reduction in the length of Code :- In JSP we use action tags, custom tags etc.
- Connection to Database is easier :-It is easier to connect website to database and allows to read or write data easily to the database.
- Make Interactive websites :- In this we can create dynamic web pages which helps user to interact in real time environment.
- Portable, Powerful, flexible and easy to maintain :- as these are browser and server independent.
- No Redeployment and No Re-Compilation :- It is dynamic, secure and platform independent so no need to re-compilation.
- Extension to Servlet :- as it has all features of servlets, implicit objects and custom tags
    1. Declaration Tag :-It is used to declare variables.
    2. Java Scriplets :- It allows us to add any number of JAVA code, variables and expressions.
    3. JSP Expression :- It evaluates and convert the expression to a string.
    4. It does not require advanced knowledge of JAVA
    5. It is capable of handling exceptions
    6. Easy to use and learn
    7. It contains tags which are easy to use and understand
    8. Implicit objects are there which reduces the length of code
    9. It is suitable for both JAVA and non JAVA programmer
    10. Difficult to debug for errors.
    11. First time access leads to wastage of time
    12. It's output is HTML which lacks features.

TEST CASES

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

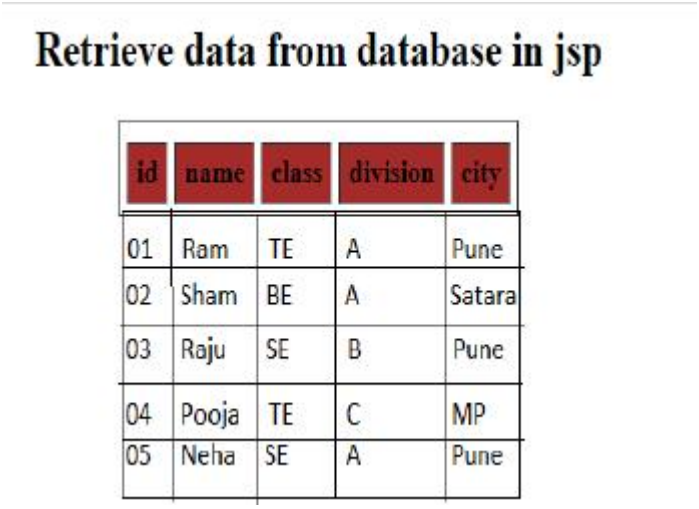
PROGRAM CODE & OUTPUT

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

```
<%@page import="java.sql.Statement"%>
<%@page import="java.sql.Connection"%>
<%
String id = request.getParameter("id");
String driverName = "com.mysql.jdbc.Driver";
String connectionUrl = "jdbc:mysql://localhost:3306/";
String dbName = "student";
String userId = "root";
String password = "aarati";
try {
Class.forName(driverName);
} catch (ClassNotFoundException e)
{ e.printStackTrace();
}
Connection connection = null;
Statement statement = null;
ResultSet resultSet = null;
%>
<h2 align="center"><font><strong>Retrieve data from database in jsp</strong></font></h2>
<table align="center" cellpadding="5" cellspacing="5" border="1">
<tr>
</tr>
<tr bgcolor="#A52A2A">
<td><b>id</b></td>
<td><b>name</b></td>
<td><b>class</b></td>
<td><b>division</b></td>
<td><b>city</b></td>
</tr>
<%
try{
connection = DriverManager.getConnection(connectionUrl+dbName, userId, password);
statement=connection.createStatement();
String sql ="SELECT * FROM studinfo";
resultSet = statement.executeQuery(sql);
while(resultSet.next()){
```

```
%>
<tr bgcolor="#DEB887">
<td><%=resultSet.getString("id") %></td>
<td><%=resultSet.getString("name") %></td>
<td><%=resultSet.getString("class")  %></td>
<td><%=resultSet.getString("division")  %></td>
<td><%=resultSet.getString("city")  %></td>
</tr>
<%
}
} catch (Exception e)
{ e.printStackTrace();
}
%>
</table>
```

Output:-



CONCLUSION/ANALYSIS

Hence, we have Implemented the the program demonstrating the use of JSP.

Assignment 7

Title: PHP and MySQL.

PROBLEM STATEMENT

Build a dynamic web application using PHP and MySQL.
a. Create database tables in MySQL and create connection with PHP.
b. Create the add, update, delete and retrieve functions in the PHP web app interacting with MySQL database

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

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 realize that anything was even transmitted to the server.

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

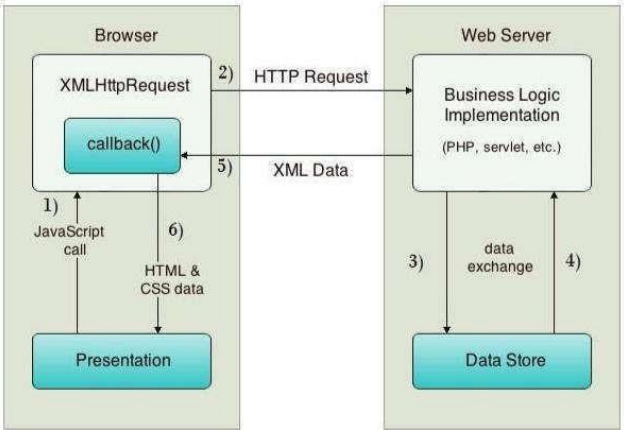
AJAX is an acronym for Asynchronous JavaScript and XML. It is a gathering of between related innovations like JavaScript, DOM, XML, HTML, CSS and so forth.

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:



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

PROGRAM CODE & OUTPUT

Connection.php :-

?php

// Enter your Host, username, password, database below.

```
// I left password empty because i do not set password on localhost.
$con=mysqli_connect('localhost:4306','root','','myprograming') or die("connection failed :
".mysqli_connect_error());
if ($con) {
// echo "Connection Successfully";
}
else{
echo "Sorry Some Mistakes is";
}
// Check connection
if (mysqli_connect_errno())
{
echo "Failed to connect to MySQL: " . mysqli_connect_error();
}
?>
```

Index.php :-

```
<!DOCTYPE html>
<html>
<head>
<title>insert</title>
</head>
<body>
<div class="row text-center">
<div class="container">
<h1>Insert DATA</h1>
<form action="index.php" method="post">
<input type="text" name="firstname" placeholder="firstname"><br><br>
<input type="text" name="lastname" placeholder="lastname"><br><br>
<input type="gmail" name="gmail" placeholder="gmail"><br><br>
<input type="text" name="number" placeholder="number"><br><br>
<input type="text" name="address" placeholder="address"><br><br>
<input type="submit" name="submit" value="insert"><br><br>
</form>
<button><a href="show.php">show data</a></button>
</div>
</div>
```

```
</body>
</html>
<?php  error_reporting(0);
include 'connection.php';
if (isset($_POST['submit'])) {
$firstname = $_POST['firstname'];
$lastname = $_POST['lastname'];
$gmail = $_POST['gmail'];
$number = $_POST['number'];
$address = $_POST['address'];
$sql = "INSERT INTO `reg` VALUES ('$id','$firstname','$lastname','$gmail','$number','$address')";
$data=mysqli_query($con,$sql);
if ($data) {
echo "insert";
}else
{
echo "sorry";
}
}
?>
```

Show.php :-

```
<!DOCTYPE html>
<html>
<head>
<title>show table</title>
</head>
<body>
<?php
include ('connection.php');
$sql ="select * from reg";
$data =mysqli_query($con,$sql);
$total=mysqli_num_rows($data);
if ($total=mysqli_num_rows($data)) {
```

```
?>
<table border="2">
<tr>
<th>id</th>
<th>firstname</th>
<th>lastname</th>
<th>gmail</th>
<th>number</th>
<th>address</th>
<th>delete</th>
<th>update</th>
</tr>
<?php
while ($result = mysqli_fetch_array($data))
{ echo "
<tr>
<td>".$result['id']."</td>
<td>".$result['firstname']."</td>
<td>".$result['lastname']."</td>
<td>".$result['gmail']."</td>
<td>".$result['number']."</td>
<td>".$result['address']."</td>
<td><a href='update.php?id=$result[id] & firstname=$result[firstname] &
lastname=$result[lastname] & gmail=$result[gmail] & number=$result[number]
&address=$result[address]'> update </a></td>
<td><a href='delete.php?id=$result[id] '>delete </a></td>
</tr>
";
}
}
else
{
echo "no record found";
}
?>
</table>
```

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

Update.php:-

```
<!DOCTYPE html>
<html>
<head>
<title>update</title>
</head>
<body>
<form action="" method="get">
<input type="text" name="id" placeholder="id" value="<?php echo $_GET['id']; ?>"><br><br>
<input type="text" name="firstname" placeholder="firstname" value="<?php echo
$_GET['firstname']; ?>"><br><br>
<input type="text" name="lastname" placeholder="lastname" value="<?php echo
$_GET['lastname']; ?>" ><br><br>
<input type="gmail" name="gmail" placeholder="gmail" value="<?php echo $_GET['gmail'];
?>"><br><br>
<input type="text" name="number" placeholder="number" value="<?php echo $_GET['number'];
?>"><br><br>
<input type="text" name="address" placeholder="address" value="<?php echo $_GET['address'];
?>"><br><br>
<input type="submit" name="submit" value="update">
</form>
<?php
error_reporting(0);
include ('connection.php');
if ($_GET['submit'])
{
$id = $_GET['id'];
$firstname = $_GET['firstname'];
$lastname = $_GET['lastname'];
$gmail = $_GET['gmail'];
$number = $_GET['number'];
$address = $_GET['address'];
$sql="UPDATE reg SET firstname='$firstname' , lastname='$lastname', gmail='$gmail' ,
number='$number', address='$address' WHERE id='$id'";
```

```
$data=mysqli_query($con, $sql);
if ($data) {
//echo "record update";
header('location:show.php');
}
else{
echo "not update";
}
}
else
{
echo "click on button to save the change";
}
?>
</body>
</html>
```

Delete.php :-

```
<?php
include ('connection.php');
$id = $_GET['id'];
$sql ="DELETE FROM `reg` WHERE id='$id'";
$data = mysqli_query($con,$sql);
if ($data) {
echo "deleted";
header('location:show.php');
}else
{
echo "error";
}
?>
```

Output:-

CONCLUSION/ANALYSIS

Hence, we have implemented a dynamic web application using PHP and MySQL.

Assignment 8

Title: Struts.

PROBLEM STATEMENT

Design a login page with entries for name, mobile number email id and login button. Use struts and perform following validations
a. Validation for correct names
b. Validation for mobile numbers
c. Validation for email id
d. Validation if no entered any value
e. Re-display for wrongly entered values with message
f. Congratulations and welcome page upon successful entries

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 (Minimum Requirement):

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

THEORY-CONCEPT

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

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.

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

PROGRAM CODE & OUTPUT

index.jsp

```
<%@ taglib prefix="s" uri="/struts-tags" %>
<html>
<body>
<s:form action="verify">
<s:textfield name="uname" label="Enter Username" /><br>
<s:password name="password" label="Enter Password" /><br>
<s:submit value="Click" align="center" />
</s:form>
</body>
</html>
```

success.jsp

```
<%@ taglib prefix="s" uri="/struts-tags" %>
Hello <s:property value="uname" />, you have been successfully logged in
```

error.jsp

```
<%@ taglib prefix="s" uri="/struts-tags" %>
```

Login failed...!

LogingEx.java

```
package example;
import com.opensymphony.xwork2.ActionSupport;
public class LogingEx extends
ActionSupport{ private static final long
serialVersionUID = 1L; private String
uname,password;
public String getUname()
{ return uname;
}
public void setUname(String uname)
{ this.uname = uname;
}
public String getPassword()
{ return password;
}
public void setPassword(String password)
{ this.password = password;
}
public String execute()
{
if(uname.equals("example") && password.equals("pass"))
{
return SUCCESS;
}else
return ERROR;
}
}
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://java.sun.com/xml/ns/javaee/web-
app_2_5.xsd" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" id="WebApp_ID" version="2.5">
```

```
<filter>
<filter-name>struts2</filter-name>
<filter-class>org.apache.struts2.dispatcher.FilterDispatcher</filter-class>
</filter>
<filter-mapping>
<filter-name>struts2</filter-name>
<url-pattern>/*</url-pattern>
</filter-mapping>
<welcome-file-list>
<welcome-file>index.jsp</welcome-file>
</welcome-file-list>
</web-app>
```

struts.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
"-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
"http://struts.apache.org/dtds/struts-2.0.dtd">
<struts>
<include file="struts-default.xml"/>
<package name="a" extends="struts-default">
<action name="verify" class="examples.LogingEx">
<result name="success">/success.jsp</result>
<result name="error">/error.jsp</result>
</action>
</package>
</struts>
```

Output:-

CONCLUSION/ANALYSIS

Hence, we have designed a login page with entries for name, mobile number email id and login button using struts and performed validations

Assignment 9

Title: Angular JS.

PROBLEM STATEMENT

Design an application using Angular JS.
e.g., Design registration (first name, last name, username, password) and login page using Angular JS.

OUTCOMES

Students should be able to,

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

SOFTWARE & HARDWARE REQUIREMENTS

Software (Minimum Requirement):

Eclipse IDE/ Notepad/ Notepad++, Visual studio, 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 "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 innovations."

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.

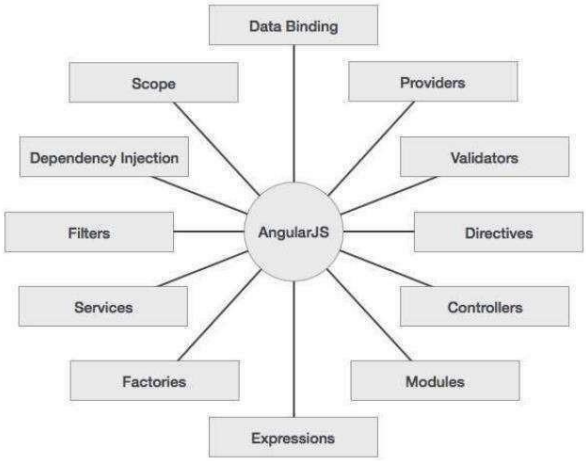


Figure.7: Architecture of AngularJS

Core Features:

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.

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.

PROGRAM CODE & OUTPUT

App.component.html :-

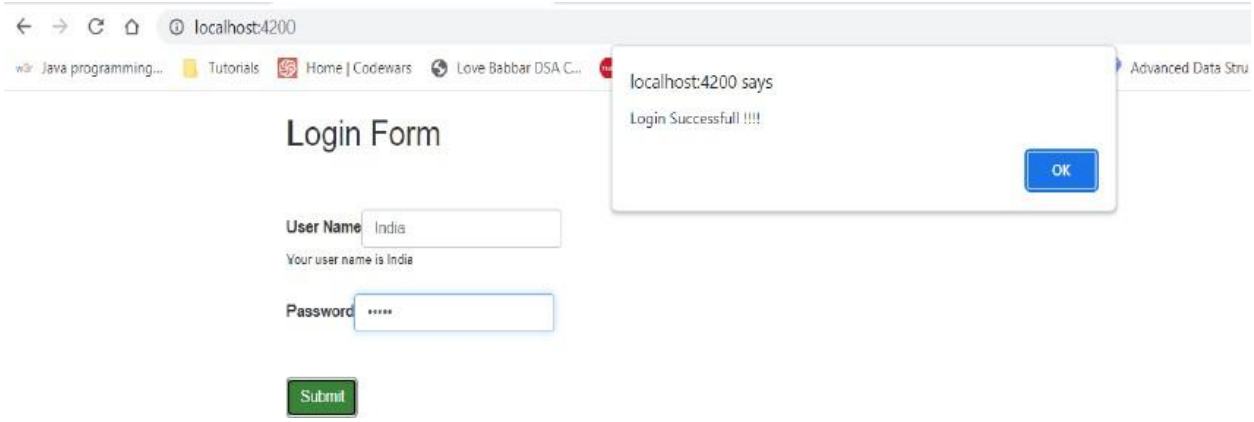
```
<!DOCTYPE html>
<html lang="en">
<head>
```

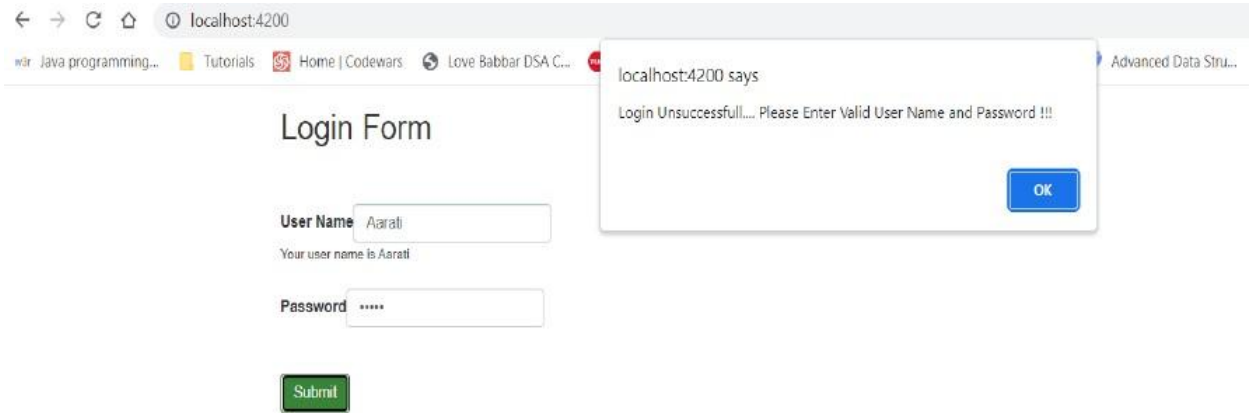
```
<title>Bootstrap Example</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
</head>
<body>
<div class="container">
<h2>Login Form</h2>
<br>
<br>
<form class="form-inline" (ngSubmit)="onSubmit(myForm)" #myForm="ngForm">
<div class="form-group">
<label for="username">User Name </label>
<input type="text" class="form-control" id="username" placeholder="Enter user name"
name="username" [(ngModel)]="UserName" required>
</div>
<br>
<small *ngIf="UserName">Your user name is {{UserName}}</small> <br>
<br>
<div class="form-group">
<label for="password">Password</label>
<input type="password" class="form-control" id="password" placeholder="Enter Password"
name="password" [(ngModel)]="password" required>
</div>
<br>
<br>
<br>
<button type="submit" class="btn btn-success" (click)="clickme()">Submit</button>
</form>
</div>
</body>
</html>
```

App.cpmponent.ts :-

```
import { Component, ViewChild } from '@angular/core';
import { NgForm } from '@angular/forms';
@Component({
selector: 'app-root',
templateUrl: './app.component.html',
styleUrls: ['./app.component.css']
})
export class AppComponent
{ title = 'Form1';
defaultCourse="Angular";
UserName='';
password='';
@ViewChild('myForm') myForm=NgForm
onSubmit(form:NgForm){ console.log(this.
myForm)
}
clickme() {
var uname=this.UserName;
var pw=this.password;
console.log(uname);
console.log(pw);
var un="India";
var ps="India";
if(uname===un &&
pw===ps){ alert("Login
Successfull !!!!");
}
else{
alert("Login Unsuccessfull..... Please Enter Valid User Name and Password !!!");
}
}
}
```

Output:-





CONCLUSION/ANALYSIS

Hence, we have designed an application using Angular JS.

Assignment 10

TITLE: Enterprise Java Beans (EJB)

PROBLEM STATEMENT

Design and implement a business interface with necessary business logic for any web application using EJB.
e.g., Design and implement the web application logic for deposit and withdraw amount transactions 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 Processusing EJB.
3. Create a simple EJB 3 stateless session bean and a local Java application client which willcall/invoke the bean to develop for addition of two numbers.

SOFTWARE & HARDWARE REQUIREMENTS

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

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 stateleful 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

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.

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

PROGRAM CODE & OUTPUT

Source Code :

Index.jsp:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<title>Implement Bank Application using EJB</title>
```

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
<form method="post" action="transact">
<h1> Bank Application using EJB </h1>
Enter Amount : <input type="text" name="t1" /> <br/> <br/>
<h3>Select Options </h3>
<input type="radio" name="transaction" value="deposit" /> Deposit <br/>
<input type="radio" name="transaction" value="withdraw" /> Withdraw <br/>
<input type="submit" value="Submit" />
</form>
</body>
</html>
```

transact.java

```
import Bankexamp.BankTransact;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class transact extends HttpServlet {
BankTransact bankTransact = lookupBankTransactBean();
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
try(PrintWriter out = response.getWriter()) {
String selectedType = request.getParameter("transaction");
int amount = Integer.parseInt(request.getParameter("t1"));
if(selectedType.equals("deposit")){
```

```
bankTransact.deposit(amount);
}
}
if(selectedType.equals("withdraw")){
int amt = bankTransact.withdraw(amount);
out.println(amount +"succesfylly withdrawn. Your Balace is : Rs."+amt);
}
}
}
// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to
edit the code.">
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
processRequest(request, response);
}
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
processRequest(request, response);
}
@Override
public String getServletInfo()
{ return "Short description";
}// </editor-fold>
private BankTransact lookupBankTransactBean()
{ try {
Context c = new InitialContext();
return (BankTransact) c.lookup("java:global/Bank/Bank-
ejb/BankTransact!Bankexamp.BankTransact");
} catch (NamingException ne)
{ Logger.getLogger(getClass().getName()).log(Level.SEVERE, "exception caught", ne);
throw new RuntimeException(ne);
}
}
```

BankTransact

```
package Bankexamp;
import javax.ejb.Stateful;
import javax.ejb.LocalBean;
@Stateful
@LocalBean
public class BankTransact implements BankTransactLocal
{ int balance = 10000;
public void deposit(int amount)
{ balance = balance + amount;
}
public void withdraw(int amount)
{ balance = balance - amount;
return balance;
}
}
```

Output:-

CONCLUSION/ANALYSIS

Hence, we have designed and implemented a business interface with necessary business logic for any web application using EJB.

.

WT Oral Questions

- What is the difference between HTML and HTML5?
- What is the difference between html elements and tags?
- What is marquee?
- What is the use of span tag? Give an example?
- What is the use of „required "attribute in HTML5?
- What is External stylesheet? What are the advantages and disadvantages?
- What is CSS selector?
- What are the components of CSS style?
- What are browser safe color?

- What is XML DOM?
- Explain difference between CDATA and PCDATA?
- What is mean by simple element and complex element?
- What is XPATH?
- Explain XSL and XSLT?
- Explain difference between HTML and XML

- Name some Java Script features.
- How to define anonymous function?
- What is callback?
- What is the difference between undefined and not-defined in JavaScript?
- What is „closure" in JavaScript?
- What are JavaScript data types?
- What are all the types of Pop up boxes available in JavaScript?

- What is Tomcat?
- What is the tomcat default port?
- What is the Servlet container life cycle?
- What services are provided by Tomcat?
- Explain directory structure of tomcat.

- What is JSP?
- What is Servlet?
- What is the purpose of MySQL?
- What is database?
- What is the syntax of JSP?
- How do we connect JSP file to database?

- What is the use of "echo" in PHP?
- How to include a file to a PHP page?
- Differences between GET and POST methods ?
- What is the use of 'print' in PHP?
- What is the difference between Session and Cookie?

- What are the different errors in PHP?

- How to print current date and time?

- What is the difference between SQL and MySQL?

- Why do we use GROUP BY and ORDER BY function in mysql?

- What is JOIN in MySQL? What are the different types of join?

- What is AJAX?
- What is jQuery?
- How many TRIGGERS allows per table in mysql?
- What is difference between COMMIT and ROLLBACK?
- What is Ajax?
- Whether jQuery HTML work for both HTML and XML documents?
- What is the use of jQuery.ajax method ()?
- What are Ajax applications?
- How to control the duration of an Ajax request?
- What are the advantages and disadvantages of Ajax?
- Which are the two methods used for cross domain Ajax calls?
- What are all the technologies used by Ajax?
- What is JSON in Ajax?
- What are the difference between AJAX and Javascript?
- How Ajax objects can be created?

- What are the components of Struts Framework?
- What‟s the role of a handler in MVC based applications?
- What‟s the flow of requests in Struts based applications?
- Which file is used by controller to get mapping information for request routing?
- What‟s the role of Action Class in Struts?
- How an actionForm bean is created? And Its uses.
- How validation is performed in struts application?

- What‟s the purpose of Execute method of action class?
- How can we display all validation errors to user on JSP page?
- What are the benefits of Struts framework?

- What is AngularJS and what are some of its advantages?
- What is the Model View Controller (MVC)?
- What is data binding in AngularJS? How does it relate to the MVC architecture?
- Explain the concept of scope. How does scope inheritance work in AngularJS?
- Explain the difference between a factory and a service in AngularJS.
- Explain why there are two **"**destroy**"** events associated with the termination of a scope in AngularJS.
- What is dependency injection and how does it work?
- What are directives? Can you explain the functions of the following directives?
- Explain the role of $routeProvider in AngularJS.

- What is EJB?
- What is JSP?
- What is the purpose of JBOSS?
- What is the syntax of JSP?
- How to deploy java beans to server?