Sinhgad Institutes

Sinhgad Technical Education Society's

SINHGAD ACADEMY OF ENGINEERING,

PUNE-411048

DEPARTMENT OF COMPUTER ENGINEERING

Laboratory manual (Instructor's Manual)

Course **310257**

Web Technology

Prepared by:

- Ms. P. R. Dongare
- Mrs. K. T. Mohite
- Mrs. R. S. Mesharam

Vision

उत्तमपुरुषान् उत्तमाभियंत्रृन् निर्मातुं कटीबध्दः वयम् !

"We are committed to produce not only good engineersbut good human beings, also."

Mission

"Holistic development of students and teachers is what we believe in and work for. We strive to achieve this by imbibing a unique value system, transparent work culture, excellent academic and physical environment conducive to learning, creativity and technology transfer. Our mandate is to generate, preserve and share knowledge for developing a vibrant society."

Department of Computer Engineering

Vision

"To build the Department as a Centre of Excellence for students in Computer Engineering."

Mission

"We believe in developing value based system for student and staff by providing healthy and transparent work culture to cultivate new ideas in the field of engineering and technology which will contribute to build a vibrant Society."

Programme Outcomes (POs)

PO1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2. Problem analysis: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professionalengineering practice.

PO7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11. Project management and finance: Demonstrate knowledge and understanding of the engineeringand management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Programme Specific Outcomes (PSOs)

Computer Engineering graduate will be able to,

PSO1: **Project Development:** Successfully complete hardware and/or software related system or applicationprojects, using the phases of project development life cycle to meet the requirements of service and product industries; government projects; and automate other engineering stream projects.

PSO2: Domain Expertise: Demonstrate effective application of knowledge gained from different computer domains like, data structures, data bases, operating systems, computer networks, security, parallel programming, in project development, research and higher education.

PSO3: Career Development: Achieve successful Career and Entrepreneurship- The ability to employ modern computer languages, environments, and platforms in creating innovative career paths to be an entrepreneur, and a zest for higher studies.

Savitribai Phule Pune University

Third Year of Computer Engineering (2019 Course)

Home

310257: Web Technology Laboratory

Teaching Scheme Practical: 02 Hours/Week	Credit Scheme 01	Examination Scheme and Marks Term Work: 25 Marks Oral: 25 Marks

Companion Course : Web Technology (310252)

Course Objectives:

- To learn the web based development environment
- To use client side and server side web technologies
- To design and develop web applications using front end technologies and backend databases

Course Outcomes:

On completion of the course, learners will be able to

CO1: Understand the importance of website planning and website design issues

CO2: Apply the client side and server side technologies for web application development

CO3: Analyze the web technology languages, frameworks and services

CO4: Create three tier web based applications

Guidelines for Instructor's Manual

The instructor‘s manual is to be developed as a reference and hands-on resource. It should include prologue (about University/program/ institute/ department/foreword/ preface), curriculum of the course, conduction and Assessment guidelines, topics under consideration, concept, objectives, outcomes, set of typical applications/assignments/ guidelines, and references.

Guidelines for Student's Laboratory Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of Certificate, table of contents, and handwritten write-up of each assignment (Title, Date of Completion, Objectives, Problem Statement, Software and Hardware requirements, Assessment grade/marks and assessor's sign, Theory- Concept in brief, algorithm, flowchart, test cases, Test Data Set(if applicable), mathematical model (if applicable), conclusion/analysis. Program codes with sample output of all performed assignments are to be submitted as softcopy. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and programlisting to journal must be avoided. Use of DVD containing students programs maintained by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints in the Laboratory.

Guidelines for Laboratory /Term Work Assessment

Continuous assessment of laboratory work should be based on overall performance of Laboratory assignments by a student. Each Laboratory assignment assessment will assign grade/marks based on parameters, such as timely completion, performance, innovation, efficient codes, and punctuality.

Guidelines for Oral Examination

Oral examination should be jointly conducted by the internal examiner and external examiner. Relevant questions may be asked at the time of evaluation to test the student‘s understanding of the fundamentals, effective and efficient implementations in term work. This will encourage, transparent evaluation and fair approach, and hence will not create any uncertainty or doubt in the minds of the students. So, adhering to these principles will consummate our team efforts to the promising start of student's academics.

Guidelines for Laboratory Conduction

The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The assignment framing policy need to address theaverage students and inclusive of an element to attract and promote the intelligent students. Use of open source software is encouraged. Based on the concepts learned. Mini project should be implemented by the students in a group of 2-3 students.

Suggested List of Laboratory Experiments/Assignments

(All assignments are compulsory)

Sr. No.	Assignment Title
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, eithergood website or bad website in following format: From the evaluation, students should learn and conclude different website design issues, which should be considered while developing a 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, 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.

@The CO-PO Mapping Matrix

PO/CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	-	1	3	1	-	1	1	-	-	1	-	-
CO2	2	2	-	2	1	-	-	-	1	-	-	-
CO3	2	-	3	-	-	1	-	-	-	1	1	-
CO4	1	2	2	1	2	1	1	-	-	-	-	1

Sr. No.	Title of Assignment	Page No.
	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 in following format: From the evaluation, students should learn and conclude different website design issues, which should be considered while developing a website.	
	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	
	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.	
	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.	
	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.	
	Implement the program demonstrating the use of JSP. e.g., Create a database table students info (studied, stud name, class, division, city) using database like Oracle/MySQL etc. and display (use SQL select query) the table content using JSP.	
	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	
	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	
	Design an application using Angular JS. e.g., Design registration (first name, last name, username, password) and login page using Angular JS.	
	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.	
	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.	

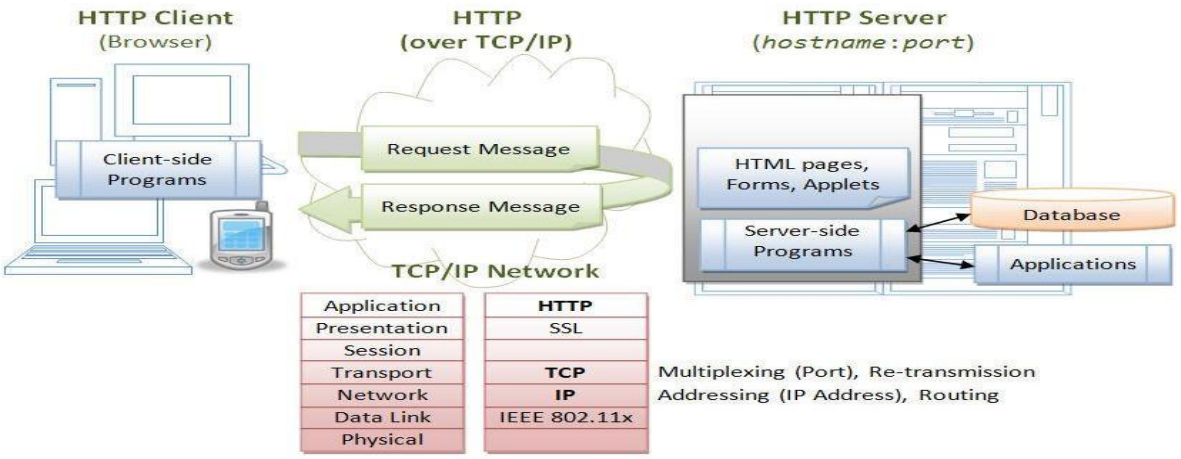
Experiment No: 1

Title	Case Study
Problem statement	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 in following format: From the evaluation, students should learn and conclude different website design issues, which should be considered while developing a website.
Prerequisite	Web server knowledge
CO mapped	1
Hardware required	OS

TITLE: TOMCAT SERVER installation

THEORY-CONCEPT:

Web Application:
A web application runs over the Internet. Ex. eBay, Amazon ,Google, facebooketcA webapp contains five components:
1.HTTP Server: Examples are- Google Web Server , Apache HTTP Server, ApacheTomcat 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.



1. To start a webapp, A user, issues a URL request via a web browser (HTTP client), toHTTP server.
2. The HTTP server returns an HTML form (client-side program), which is loaded into theclient'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 onthese 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 Apachemother site:http://www.apache.org.

How to Run Tomcat

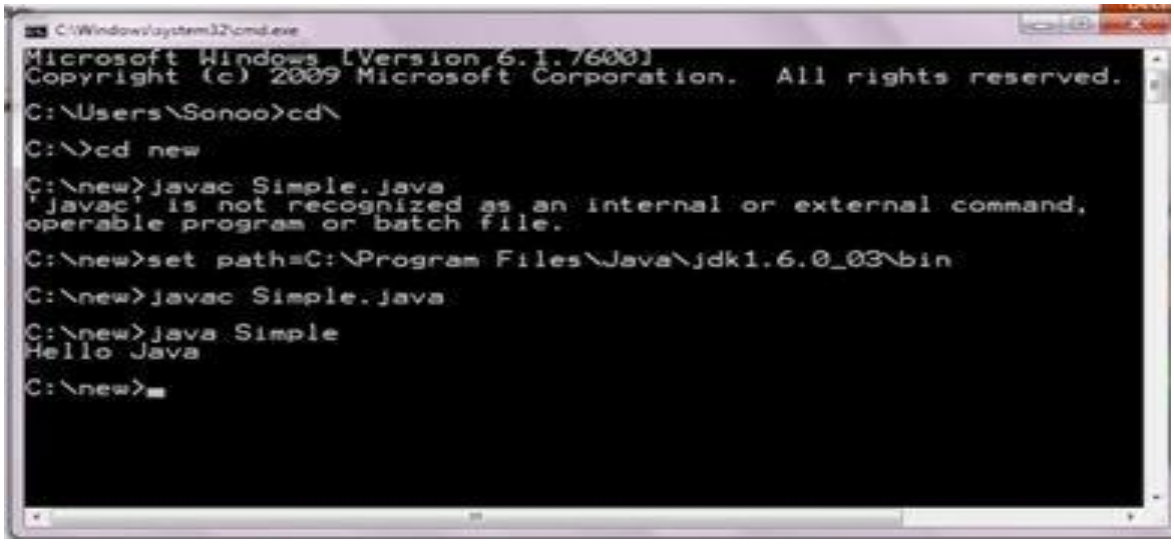
Following are the steps to check whether Tomcat has been installed successfully or not-
1. Find start program in the Programs Menu. Look under Apache Tomcat and select "StartTomcat" or “startup.bat”.
2. Open any web browser and type in the given URL:
o http://localhost:8080/

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

CONCLUSION / ANALYSIS:

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

OUTPUT:



Experiment No: 2

Title	HTML, CSS
Problem statement	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
Prerequisite	HTML, CSS
CO mapped	1
Hardware required	OS

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

CSS:

CSS stands for Cascading Style Sheet. It is nothing, but design language intended to simplifythe 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 understandand it provides powerful control on presentation of anHTML 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

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 aunique 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 todefine a style for a single HTML page. An internal CSS is defined in the <head> sectionof 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.

CODE:

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>
```

samp.css

```
.con
{
width:400px;
height:400px;
background-color:white;
color:blue;
padding:0px 10px 0px 10px;
margin:auto;
}
.main
{
width:400px;
height:80px;
float:left;
padding:60px 10px 10px10px;
margin: auto;
}
.con1
{
width:200px;
height:30px;
float:left;
margin:auto;
padding:0px 0px 0px0px;
}
.con2
{
width:150px;
height:30px;
float:left;
margin:10px 10px 10px10px;
padding:0px 0px 0px0px;
}

#terms
{

color:blue;
padding-left:400px;
}
```

two.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">
<imgsrc="/home/sanyuja/Desktop/b.jpg" width=380px height=100px alt="img">
</div>
<div class="main">
<table align="center">
<tr>
<td><h1>Welcome to Registration Page</h1></td></tr>
</table>

<div class="con1">
<table align="center">
<tr>
<td>Full Name:</td>
<td><input type="text" value=""></input></td>
</tr>


<tr>
<td>Select Gender</td>
<td><input type="radio" name="gender"/>Female</input>
<input type="radio" name="gender"/>Male</input></td>
</tr>
<br>
<tr>
<td>Class</td>
<td><input type="text" value=""></input></td>
</tr>
<br>
<tr>
<td>Skills</td>
<td><input type="checkbox" value="">C </input>
<input type="checkbox" value="">C++</input>
<input type="checkbox" value="">JAVA</input>
<input type="checkbox" value="">PHP</input>
<input type="checkbox" value="">AJAX</input></td>
```

```
</tr>
  <br>
  <tr>
  <td>Department</td>
  <td>
  <select>
    <option value="comp">COMPUTER</option>
    <option value="it">Information Technology</option>
  <option value="it">Electronics & Telecomunications</option>
    <option value="mech">Mechnical</option>
    <option value="civil">Civil</option>
  </select>
  </td>

  </tr>

  <tr>
  <td>email:</td>
  <td><input type="text" value=""></input></td>
  </tr>
  <tr>
  <td>phone:</td>
  <td><input type="text" value=""></input></td>
  </tr>

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

  <table align="center">
  <tr>
  <td><input type="submit" value="SUBMIT"></td>
  <td><input type="reset" value="CANCEL"></td>

  </td>
  </tr>
  </table>

  </div>
  </div>
  </form>

  </div>
  <br><br><br><br>

  <div id="terms">
  <span>By clicking Submit you agree to our Terms & Conditions</span>
  </div>
  </body>

  </html>
```

OUTPUT:


Welcome to Registration Page
Full Name:
Select Gender
Female
Male
Class
Skills
C
C++
JAVA
PHP
AJAX
Department
COMPUTER
email:
phone:
SUBMIT
CANCEL
By clicking Submit you agree to our Terms & Conditions

Experiment No: 3

Title	XML and CSS
Problem statement	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.
Prerequisite	XML and CSS
CO mapped	3
Hardware required	OS

THEORY:

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 asshown 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 −
XMLTutorial

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

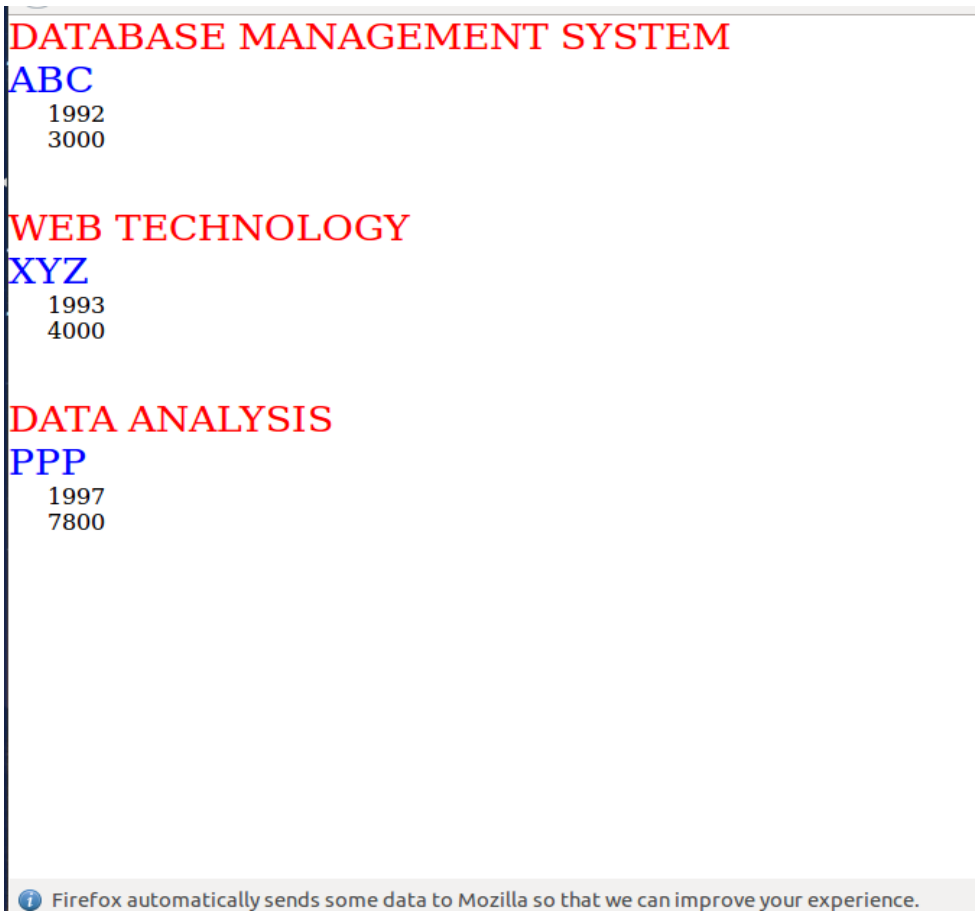
DESIGN/EXECUTION STEPS:

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 browser.

CODE:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="book.css"?>
<CATLOG>
<BOOK>
<TITLE> DATABASE MANAGEMENT SYSTEM</TITLE>
<AUTHOR>ABC</AUTHOR>
<YEAR>1992</YEAR>
<PRICE>3000</PRICE>
</BOOK>
<BOOK>
<TITLE> WEB TECHNOLOGY</TITLE>
<AUTHOR>XYZ</AUTHOR>
<YEAR>1993</YEAR>
<PRICE>4000</PRICE>
</BOOK>
<BOOK>
<TITLE> DATA   ANALYSIS</TITLE>
<AUTHOR>PPP</AUTHOR>
<YEAR>1997</YEAR>
<PRICE>7800</PRICE>
</BOOK>
</CATLOG>
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:#00000; Display:
block; Margin-left: 20pt;
Color:#00000; Display:
block;
```

OUTPUT:


DATABASE MANAGEMENT SYSTEM
ABC
1992
3000
WEB TECHNOLOGY
XYZ
1993
4000
DATA ANALYSIS
PPP
1997
7800
Firefox automatically sends some data to Mozilla so that we can improve your experience.

Experiment No: 4

Title	HTML, Java Script
Problem statement	Write a program to design registration form for students by using HTML, CSS& Java Scriptand perform following validations: all fields mandatory, phone number and email address validation.
Prerequisite	HTML, Java Script
CO mapped	CO1, CO2
Hardware required	OS

THEORY:

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. Thisis 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 sendingit 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 ofdata.

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.
1. Write the function for validation of email id and phone no and enclosed this function inscript.
2. Call this function on ‘onClick’ event of submit button.
3. Open HTML page in the browser.

CONCLUSION/ANALYSIS:

Hence, we applied validate the data using JavaScript.

CODE:

CSS

```
.one
{
width:4
00px;
```

```
height:4
00px;
backgro
und-
color:bl
ue;

padding:0px,
10px,0px,10p
x;
margin:auto;
bor
der:
soli
d;
bord
er-
colo
r:bla
ck;

}
.main{}
```

JS

```
<html>
<head>
<title>Javascript</title>
</head>
<body>
<script language="JavaScript">

document.write("Student Registeration Form");
document.write("<input type="text" value="
"></input>");

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

HTML

```
<html>
<head>
<titel>  </title>
<link href="ass4.css" rel="stylesheet"></link>
<script language="JavaScript">
```

```
function validateForm()
{
var
x=document.forms["
myForm"]
["fname"].value;
if(x=="")
{
alert("Enter the all details.....");
return false;
}

var
x=document.forms["
myForm"]
["mname"].value;
if(x=="")
{
alert("Enter the all details.....");
return false;
}

var
x=document.forms["
myForm"]
["lname"].value;
if(x=="")
{
alert("Enter the all details.....");
return false;
}

var
y=document.forms["
myForm"]
["add"].value;
if(y=="")
{
alert("Enter the all details.....");
return false;
}

var
z=document.forms["
myForm"]
["email"].value;
if(z=="")
{
alert("Enter the all details.....");
```

```
return false;
}

var
a=document.forms["
myForm"]
["ph"].value;
if(a=="")
{
alert("Enter the all details.....");
return false;
}

document.forms["myForm"].reset()
}
</script>
</head>
<body >
<center>
<div class="main">
<h1>Form Validation Using jQuery</h1>
<div id="form">
<p id="head"></p> <!-- This Segment Displays The Validation Rule -->



<form name="myForm" action="message.html" align="center"
onsubmit="returnvalidateForm()" method="post" >


First Name: <input type="text"
name="fname"><br><br> Middle Name:
<input type="text" name="mname"><br><br>
Last Name: <input type="text"
name="lname"><br><br> Email: <input
type="email" name="email"><br><br>
Phone:<input type="number"
name="ph"><br><br> Country:<select id='s'>
<option>  </option>
<option>India</option>
<option>America</option>
<option>Australia</option>
<option>Sweden</option>
<option>Africa</option>

</select>

<br><br><br>
Address:<textarea rows="5" cols="25" value="g" required></textarea><br><br><br>
 <br><input type ="submit" value="Submit" >
```

```
</form>


</div>
</center>
</body>
</html>
```

OUTPUT

:

Experiment No: 5

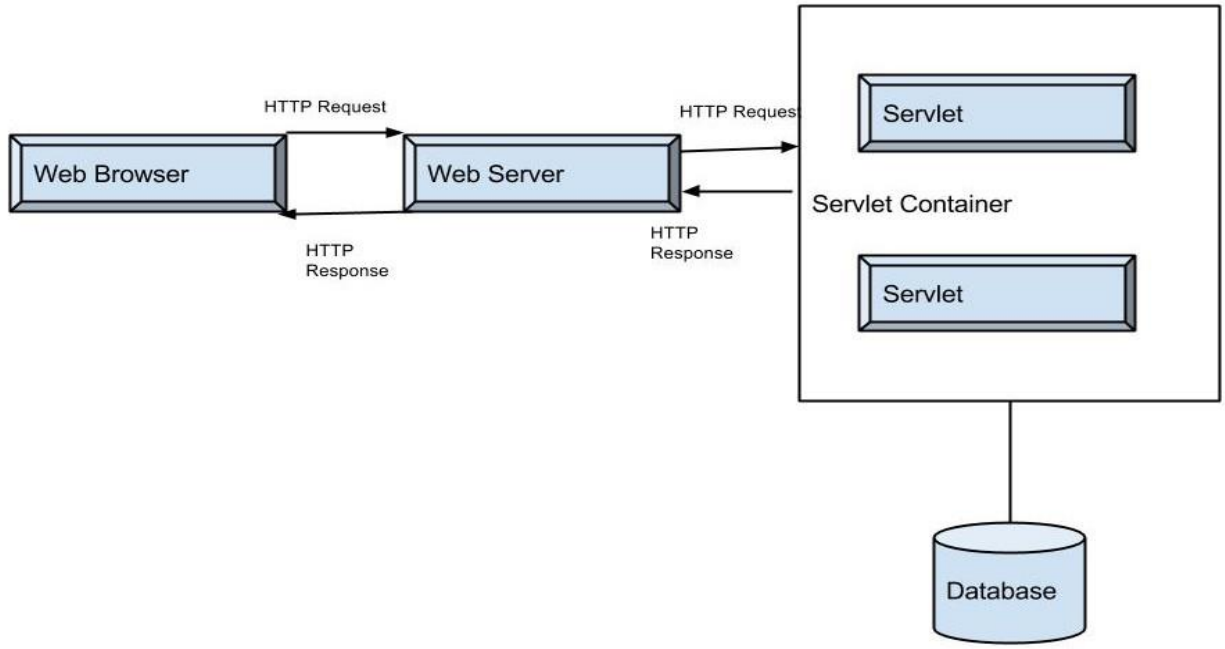
Title	Implement the sample program demonstrating the use of Servlet.
Problem statement	Write a program to design registration form for students by using HTML, CSS& Java Scriptand perform following validations: all fields mandatory, phone number and email address validation.
Prerequisite	HTML, Java Script
CO mapped	CO1, CO2
Hardware required	OS

THEORY:

What is Servlet ?

- Servlets are an important component of a J2EE application. Servlets along with JavaServer Pages (JSP) and EJB modules can be termed as server-side J2EE component types.
- Servlet is a Java Programming Language.
- Servlets are used to create web applications.
- Servlets are used to extend the applications hosted by web servers. Servlet runs in a J2EE application server

Common Gateway Interface (CGI) technology was used for dynamic content prior to introduction of Servlets. Servlet Architecture Following diagram shows the process of Servlets in a Web Application



The process can be summarized as follows:

- A client sends a request to a Web Server, through a Web Browser.
- The Web Server searches for the required Servlet and initiates it.
- The Servlet then processes the client request and sends the response back to the server, which is then forwarded to the client.

Advantages of a Servlet

- Servlets provide component based and platform-independent methods for building Web based applications.

- Each Request is run in a separate thread ,so servlet request processing is faster than CGI.
- Servlets overcomes the limitations of CGI program.
- Servlets run on Java Virtual Machine and in any platform and it is simple to write.
- Servlet are more powerful and the performance is better.
- Servlets are platform-independent.
- Servlet technology, in addition to improved performance, offers security, robustness, object orientation, and platform independence.
- As mentioned in the definition of servlets are fully integrated with the Java language and its standard APIs. Hence JDBC for Java database connectivity is also integrated in it.
- A servlet handles concurrent requests
- Handling HTTP requests and send text and data back to the client is made easy by servlet request and response objects.

Disadvantages of a Servlet

- Servlets often contain both business logic and presentation logic so it makes application difficult to understand.
- You would need JRE to be installed to run a servlet program.

Servlet API

The **Servlet API** is supported by all Servlet containers, such as Tomcat and Weblogic, etc. The Application Programming Interface (API) contains interface and classes to write a servlet program. The servlet API contains two packages as listed below:

- **javax.servlet**
- **javax.servlet.http**

Package javax.servlet

javax.servlet contains a number of classes and interface that allows the servlet to access the basic services provided by the servlet container. Following table lists the classes and interface of **javax.servlet** package:

NAME	DESCRIPTION	TYPE
HttpServletRequest	The web container provides implementation to this interface and encapsulates all HTTP based request information.	Interface

HttpServletResponse	Provide HTTP- specific functionality while sending a response.	Interface
HttpSession	It is a mechanism for storing client data across multiple HTTP requests.	Interface
HttpSessionBindingListener	Notifies when the objects of its implementation class is added or removed from the session.	Interface
Cookie	It is a file containing the information that is sent by web server to a client.	Class
HttpSessionBindingEvent	This method is used to indicate whether the object is added into the HttpSession object or removed from the HttpSession object.	Class
HttpServlet	Provides convenient methods for implementing for handling HTTP request.	Class

Servlet LifeCycle

Servlets are small programs that run at server side and creates dynamic web pages. Servlets respond to any type of requests sent by user.

In MVC architecture servlet act as controller.

The controller is the logic that processes and responds to the user requests. Life Cycle of Servlets contain following steps:

- Load servlet class.
- Create servlet instance.
- Call the init method.
- Call the service method.
- Call the destroy method.

Servlet Class Loading:- The first step in creation of a servlet component is to load the servlet class file into web container's JVM (Java Virtual Machine). This step is invoked when either first time servlet is invoked or configured in the web.xml with load-on-startup element.

Creating Servlet Instance

After the servlet class has been loaded into the web container's JVM, the next step is to create an instance of that class. Servlet specification declares one and only one servlet instance will be created for a single definition in the deployment descriptor.

The init () method

After servlet instance is created, the web container initializes the parameters that were specified in the deployment descriptor. This method is invoked only when servlet is first loaded into memory.

The syntax of init () method look like this:

public void init () throws Servlet Exception { //code }

The service () method

After the servlet component has been initialized, the web container can begin sending requests to that component using the service method .This method is used to process the request. For each request the web container will issue unique request and response to the service method.

The syntax of service () method as follows:

public void service (Servletrequest request, Servletresponse response) throws ServletException, IOException { }

When service () method is called by web container it invokes doGet (), doPost (), doPut (), doDelete (), doTrace (), doOptions (), getLastModified () methods.

The doGet () and doPost () are the two methods which are frequently used with each service request. We must override doGet () and doPost () methods depending on type of request. The doGet () method: By using doGet () method we can send specific amount of data. If we use doGet () method data is shown in address bar. We must override doGet () method depending on type of request.

It can be defined as follows:

public void doGet (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{ //code }

The doPost () method: We can send large amount of data by using doPost () method. By using this method data is not viewable in address bar. When we want to send secure data like passwords and other things doPost () method is used. We must override doPost () method depending on type of request. It can be defined as follows:

public void doPost (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{ //code }

doDelete (): It is used to delete files, web pages or documents from the server. If requests are formatted incorrectly then it will return HTTP "Bad Request" error. It can be defined as follows: protected void doDelete (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{ //code }

doPut() : This method is used to put files, web pages or documents in the server means for uploading files on the server. If requests are formatted incorrectly then it will return HTTP "Bad Request" error. It can be defined as follows: protected void doPut (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{ //code }

doTrace () : This method is used for logging and debugging purpose. It can be used for testing the requested message. There is no need to override this method. It can be defined as follows: protected void doTrace(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{ //code }

doOptions (): This method handles OPTIONS request. There is no need to override this method. It determines which HTTP method supported by server and returns correct header. If requests are formatted incorrectly then it will return HTTP "Bad Request" error. It can be defined as follows: protected void doOptions(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{ //code }

getLastModified (): It returns the time when request was last modified. This method should override GET request to return modification time of object. It can be defined as follows: protected long getLastModified (HttpServletRequest request) throws ServletException, IOException{ //code }

The destroy () method When a web application is being shut down web container will call destroy method. It is used to clean up any resources that servlet might have initialized.

The syntax of destroy () method as follows: public void destroy(){ //code }

Sample Code Following is the sample source code structure of a servlet example to show Hello World −

```
// Import required java libraries
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
// Extend HttpServlet class
public class HelloWorld extends HttpServlet {
private String message;
public void init() throws ServletException {
// Do required initialization
message = "Hello World";
}
public void doGet(HttpServletRequest request, HttpServletResponse
response)
throws ServletException, IOException {
// Set response content type
response.setContentType("text/html");
// Actual logic goes here.
PrintWriter out = response.getWriter();
out.println("<h1>" + message + "</h1>");
}
public void destroy() {
// do nothing.
}
}
```

MySQL

MySQL is a Relational DataBase Management System (RDBMS).

RDBMS means R--DB--MS.

- DB stands for Database, a repository for the information store.

The data in a database is organized into tables, and each table is organized into rows and columns.

Each row in a table is called a record. A record may contains several pieces (called fields) of information, and each column in a table is known as a field.

-MS stands for Management System, the software that allows you to insert, retrieve, modify, or delete records.

-R stands for Relational, indicates a particular kind of DBMS that is good at relating information stored in one table to information stored in another table by looking for elements common to each of them. Relational DBMS has the advantage of efficient storage, and retrieval mechanisms for data, and uses normalization process during design of RDBMS. Database normalization process is beyond the scope of this article, and several references are available.

MySQL operates using client/server architecture in which the server runs on the machine containing the databases and clients connect to the server over a network. The server operating systems is usually a Linux (like Redhat 9.0 etc.) or Windows 2000 operating system. Typically mySQL is supported on Windows XP, Windows Server 2003, Red Hat Fedora Linux, and Debian Linux, and others. As with any other client/server application, MySQL is a multi-user database system, meaning several users can access the database simultaneously. Here:

- The server (MySQL server) listens for client requests coming in over the network and accesses database contents according to those requests and provides that to the clients.
- Clients are programs that connect to the database server and issue queries in a pre-specified format. MySQL is compatible with the standards based SQL (SQL stands for Structured Query Language) language. The client program may contact the server programmatically (meaning a

program call the server during execution) or manually. For example, when you are issuing commands over a telnet session to a MySQL server, you are issuing the requests to the server by typing commands at your command prompt manually. On the other hand, if you have input some data (say your credit card information on the Internet towards purchase of some goods) in a form, and the form is processed by using a server side program, then the MySQL server is contacted programmatically. This is often the case in credit card approvals, member subscriptions etc.

Features of MySQL

1. Speed: Ofcourse, the speed at which a server side program runs depends primarily on the server hardware. Given that the server hardware is optimal, MySQL runs very fast. It supports clustered servers for demanding applications.

2. Ease of use: MySQL is a high-performance, relatively simple database system. From the beginning, MySQL has typically been configured, monitored, and managed from the command line. However, several MySQL graphical interfaces are available as described below:

- **MySQL Administrator:** This tool makes it possible for administrators to set up, evaluate, and tune their MySQL database server. This is intended as a replacement formysqladmin.
- **MySQL Query Browser:** Provides database developers and operators with a graphical database operation interface. It is especially useful for seeing multiple query plans and result sets in a single user interface.
- **Configuration Wizard:** Administrators can choose from a predefined list of optimal settings, or create their own.
- **MySQL System Tray:** Provides Windows-based administrators a single view of their MySQL instance, including the ability to start and stop their database servers.

3. Cost: MySQL is available free of cost. MySQL is a "Open Source" database. MySQL is part of LAMP (Linux, Apache, MySQL, PHP / Perl / Python) environemtn, a fast growing open source enterprise software stack. More and more companies are using LAMP as an alternative to expensive proprietary software stacks because of its lower cost, reliability, and documentation.

4.Query Language Support: MySQL understands standards based SQL (Structured Query Language).

5.Capability :Many clients can connect to the server at the same time. Clients can use multiple database simultaneously. You can access MySQL using several interfaces such as command-line clients, Web browsers.

6.Connectivity and security: MySQL is fully networked, and database can be accessed from anywhere on the Internet, so you can share your data with anyone, anywhere. The connectivity could be achieved with Windows programs by using ODBC drivers. By using the ODBC connector to MySQL, any ODBC-aware client application (for example, Microsoft Office, report writers, Visual Basic) can connect to MySQL.

7.Portability: MySQL runs on many varieties of UNIX, as well as on other non-UNIX systems, such as Windows and OS/2. MySQL runs on hardware from home PCs to high-end server. MySQL can be installed on Windows XP, Windows Server 2003, Red Hat Fedora Linux, Debian Linux, and others.

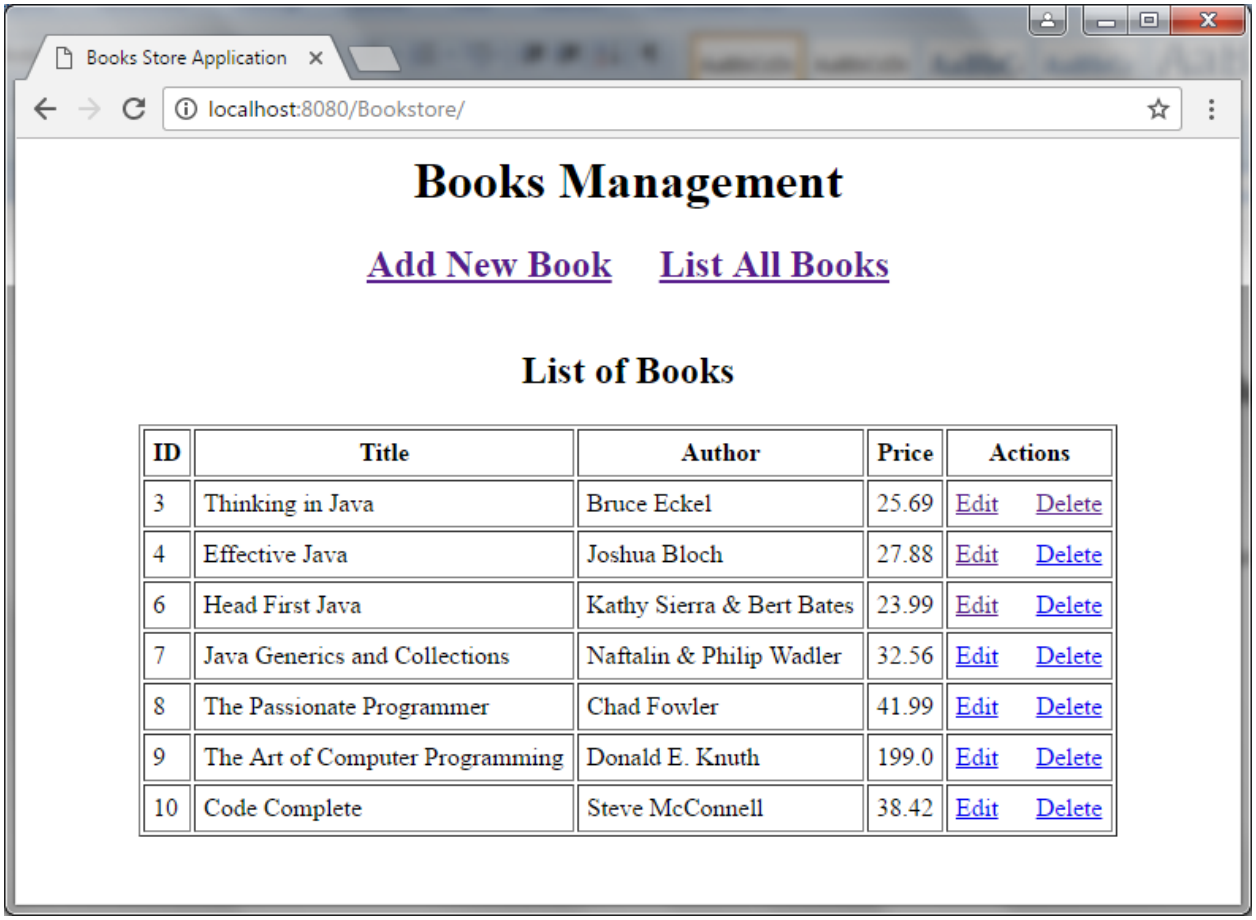
MySQL Tools

1. A SQL server: This is an engine which provides access to your databases.
2. Client programs for accessing the server: A program allows you to enter queries directly and view results.
3. A client library for writing your own programs: You can write your own programs into the client library using C.

DESIGN/EXECUTION STEPS: https://www.codejava.net/coding/jsp-servlet-jdbc-mysql-create-read-

update-delete-crud-example refer this for code and execution purpose.

CONCLUSION/ANALYSIS: Hence, we applied validate the data using Java Servlet.



****

How to build this application using the following technologies:

- Java Servlets and Java Server Pages (JSP)
- JSP Standard Tag Library (JSTL)
- Java Database Connectivity (JDBC)
- MySQL database
- Apache Tomcat Server

1. Creating MySQL Database

For simplicity, we have only one table. Execute the following MySQL script to create a database named **Bookstore** and a table named **Book**:

```
CREATE DATABASE 'Bookstore';
USE Bookstore;

CREATE TABLE `book` (
  `book_id` int(11) NOT NULL AUTO_INCREMENT,
  `title` varchar(128) NOT NULL,
  `author` varchar(45) NOT NULL, `price` float NOT NULL,
  PRIMARY KEY (`book_id`),
  UNIQUE KEY `book_id_UNIQUE` (`book_id`),
```

```
  UNIQUE KEY `title_UNIQUE` (`title`)
) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=latin1
```

The table **book** has structure like this:

```
mysql> desc book;
+---------+--------------+------+-----+---------+----------------+
| Field   | Type         | Null | Key | Default | Extra          |
+---------+--------------+------+-----+---------+----------------+
| book_id | int(11)      | NO   | PRI | NULL    | auto_increment |
| title   | varchar(128) | NO   | UNI | NULL    |                |
| author  | varchar(45)  | NO   |     | NULL    |                |
| price   | float        | NO   |     | NULL    |                |
+---------+--------------+------+-----+---------+----------------+
4 rows in set (0.03 sec)
```

Creating Eclipse Project with Maven

In Eclipse IDE, click **File > New > Dynamic Web Project** to create a new Java dynamic web project. Name the project as Bookstore:

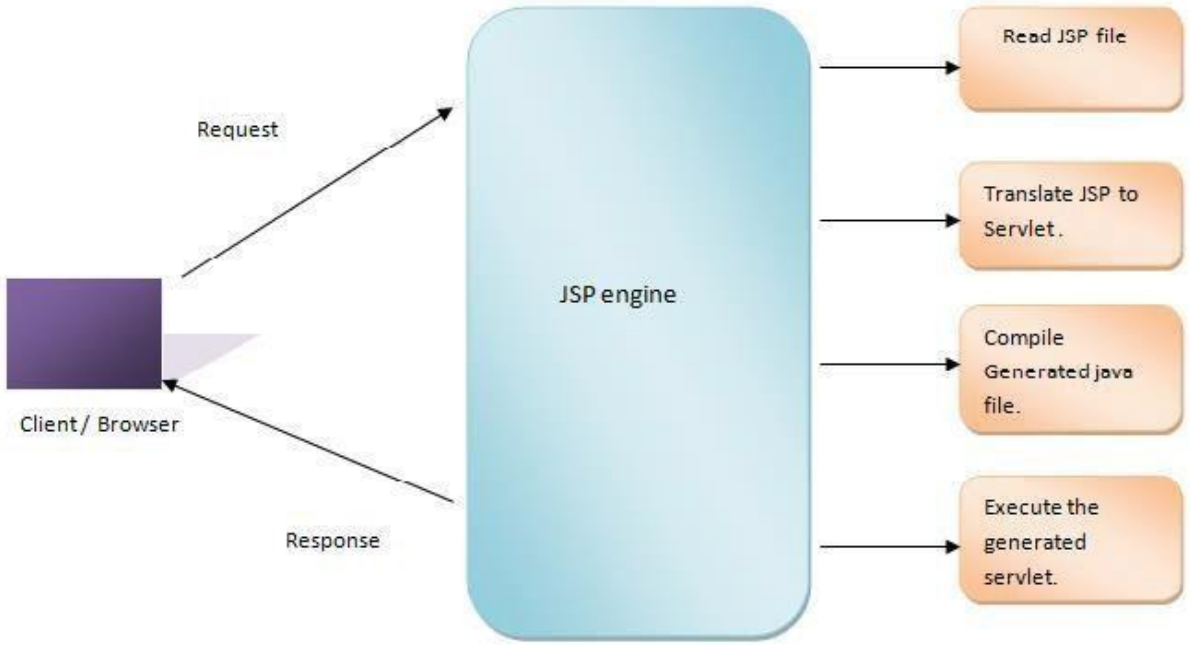
Experiment No: 6

Title	JSP, Servlet and MySQL(Backend)
Problem statement	1. Design and Build Student Login Page using JSP, Servlet and MySQL. 2. Design and Build Employee Login Page using JSP, Servlet and MySQL. .
Prerequisite	JSP, Servlet and MySQL(Backend)
CO mapped	CO3
Hardware required	OS

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 theJDBC API to access the databases.
It is a technology that helps software developers to create dynamic web pages based onHTML,
XML and other document types. It was released in 1999 by Sun Microsystems. It is just like aPHP 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 adatabase 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.

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 controllerand the JSP is the view.



Why we need JSP?

The JSP Architecture

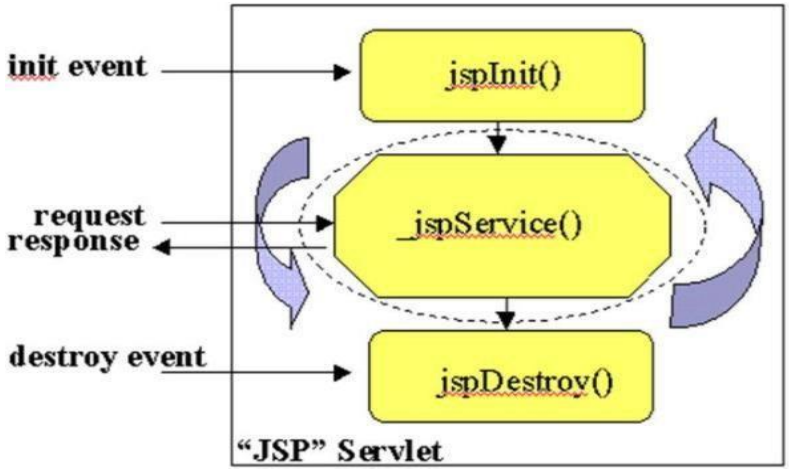
Syntax of JSP:

JSP declarations is used to declare variables and methods as shown below,

<%
text
%>.

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.



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

CONCLUSION / ANALYSIS:

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

CODE:

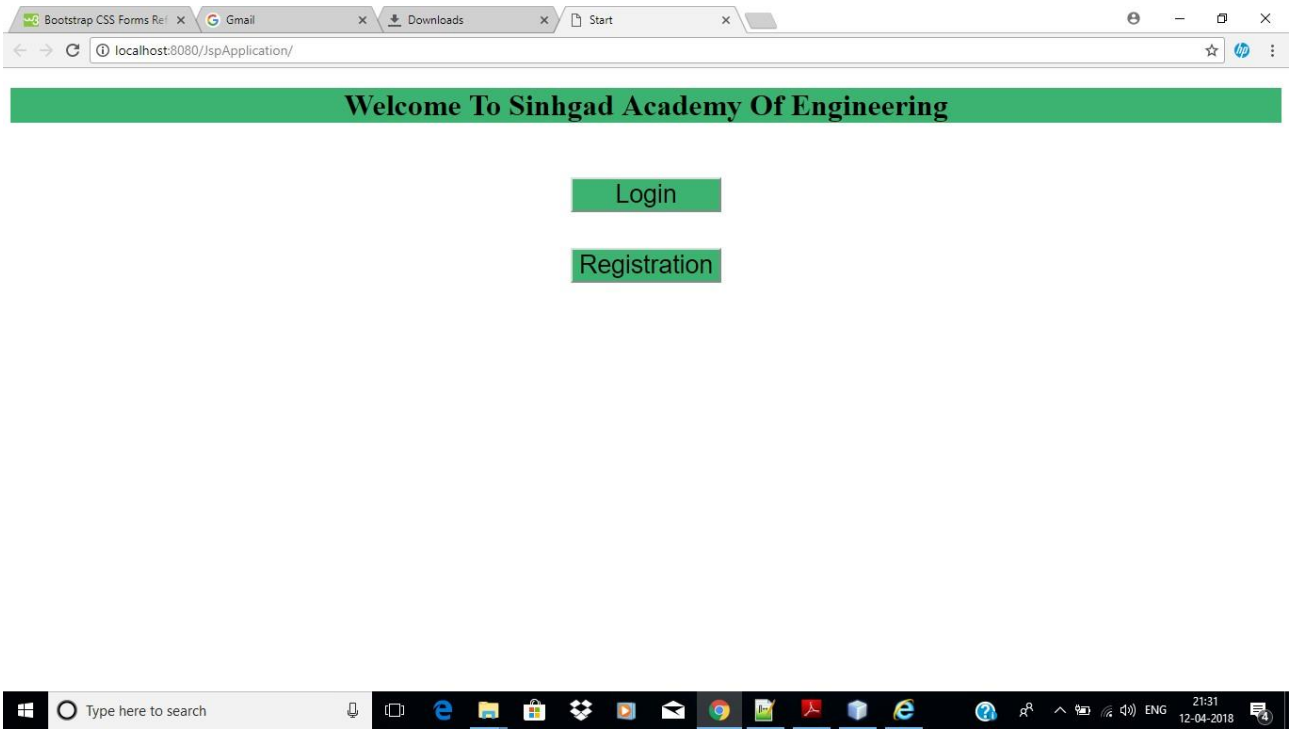
Problem Statement: Design and Build Student Login Page using JSP, Servlet andMySQL.

Program:

Index.html Page:

```
<html>
   <head>
      <title>Start</title>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
   </head>
   <body>
      <center><h1 style="background-color:MediumSeaGreen;">Welcome To
Sinhgad Academy OfEngineering</h1></center><br><br>

<a href="Signin.html"><center><input type="button" value="Login"
style="width:160px;font-size:20pt;color:black;
background-color:MediumSeaGreen;"></center></a>
```

```
<br>

 <a href="registration.html"><center><h2><input type="button"
 value="Registration"style="width:160px;font-size:20pt;color:black;

 background-color:MediumSeaGreen;"></h2></center></a>


 <br><br>

 <
 /
 b
 o
 d
 y
 >

 </html>
```



Sinhagad.jsp

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

 <!DOCTYPE html>

 <html>

   <head>

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

```
        <title>Sign In JSP</title>
        <%@ page import="java.sql.*" %>
    <%@ page import="javax.sql.*" %>
    <%@ page import="java.io.*" %>
      </head>
      <body>
        <%
    String
    user=request.getParameter("userna
    me"); String
    pword=request.getParameter("pass
    word");
    Try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/stud", "root",
    "");
            String sql="select * from reg where username=?
            and password=?";PreparedStatement
            pst=conn.prepareStatement(sql); pst.setString(1,
            user);
            pst.setString(2,pword);

            ResultSet rs=pst.executeQuery();
            if(rs.next())
            {
                    %>
                    <jsp:forward page="home.jsp"/>
                    <%
            }
            else
            {
    %>
```

```
                    <jsp:forward page="Signin.html"/>

                    <%

            }


}

catch(Exception e){

System.out.pri
        ntln(e);

}

%>

   </body>

</html>
```

Signin.html

```
<html>

   <head>

  <title>Login
   Page</title>

          <meta
charset="UTF
          -8">

      <meta name="viewport" content="width=device-width, initial-scale=1.0">

   </head>

   <body>


      <form  action="Signin.jsp" id="std"  method="POST" >

<center>Username : <input type="text" name="username" id="user"
style="width:200px;font-size:15pt;" ><br><br></center>

<center>Password : <input type="password" name="password" id="pass"
```

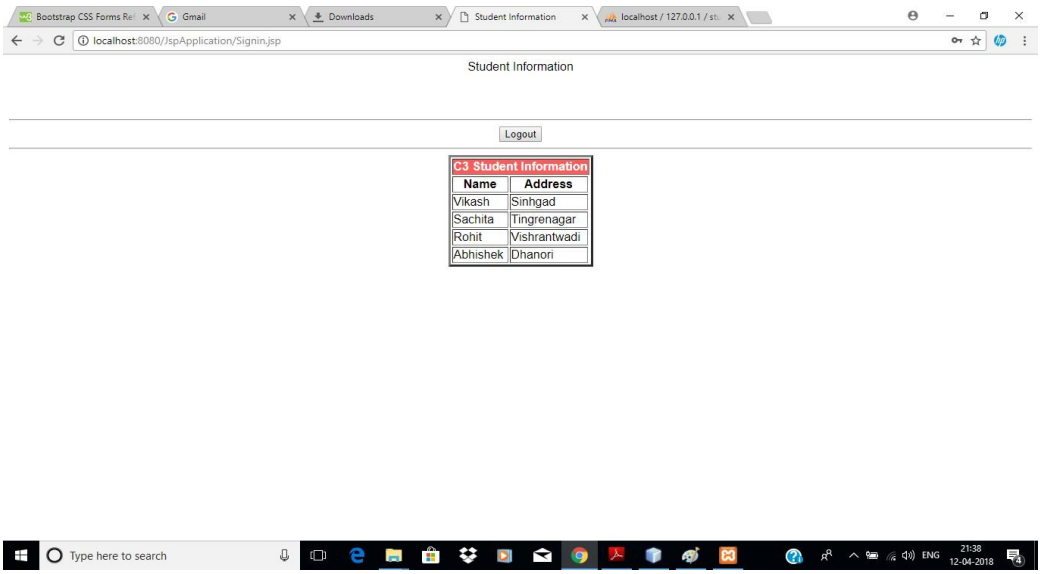
```
style="width:200px;font-size:15pt;" ></center><br>

<center><input type="submit" name="login" value="Login" id="sub"
style="width:160px;font-size:20pt;color:black;

background-color:grey;">    


<input type="reset" value="Reset" style="width:160px;height:38px;font-
size:20pt;color:black;background-color:grey;"></center>

</form>

</body>

</html>
```

Omg.jsp

```
<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Student Information</title>
```

```
<%@ page import="java.sql.*" %>
<%@ page import="javax.sql.*" %>
<%@ page import="java.io.*" %>
</style>
</head>
<body style="font-family:Veranda,sans-serif;">
<center>
<div class="header">Student Information</div>
<br><br><br>
<%
try{
        Class.forName("com.mysql.jdbc.Driver");
        Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/stud", "root",
"");

        String sql="select * from info";
        PreparedStatement
        pst=conn.prepareStatement(sql);ResultSet
        rs=pst.executeQuery();

        out.print("<table border=3px solid red>");
    //width=\"100%\">
        out.print("<tr style=\"background-
        color:#ff5b5b;color:white;\">");out.print("<th
        colspan=\"5\">C3 Student Information</th>");
        out.print("</tr>");
        out.print("<tr
        >");
        out.print("<t
```

```
        h>Name<\t>

        ");

        out.print("<th>Address</th>");
        out.print("</tr>");

        while(rs.next())

        {
                out.print("<tr>");

         out.print("<td>");

                out.print(rs.get

                String("name");

                out.print("</t");

                out.print("<td>");

                out.print(rs.getStr

                ing("address"));

                out.print("</td>";

                out.print("</tr>");

        }
}

catch(Exception e)

{
        System.out.println(e);

}

%>

<hr/>

<% %>

<a href="index.html" style="hover:green;"><button  class="button logout">Logout</button></a>

<hr/>

</body>

</html>
```



registrat ion.jsp

```
<%@ page language="java" contentType="text/html;

   charset=ISO-8859-1"pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML
4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Sign Up JSP</title>

</head>

<body>

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

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

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

<%

String name=request.getParameter("name");
```

```
String
gender=request.getParameter("g
ender");String
email=request.getParameter("e
mail"); String
college=request.getParameter("c
ollege");String
branch=request.getParameter("b
ranch");String
mobile=request.getParameter("
mobile");
String
username=request.getParameter("username
");String
pass=request.getParameter("password");


   System.ou
t.println(na
me);try{
        Class.forName("com.mysql.jdbc.Driver");
        Connection conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/stud", "root",
"");


        String sql="insert into
reg(name,gender,email,college,branch,mobile,username,password)
values(?,?,?,?,?,?,?,?)";
        PreparedStatement
        pst=conn.prepareStatement(sql);
        pst.setString(1, name);
        pst.setString(2
```

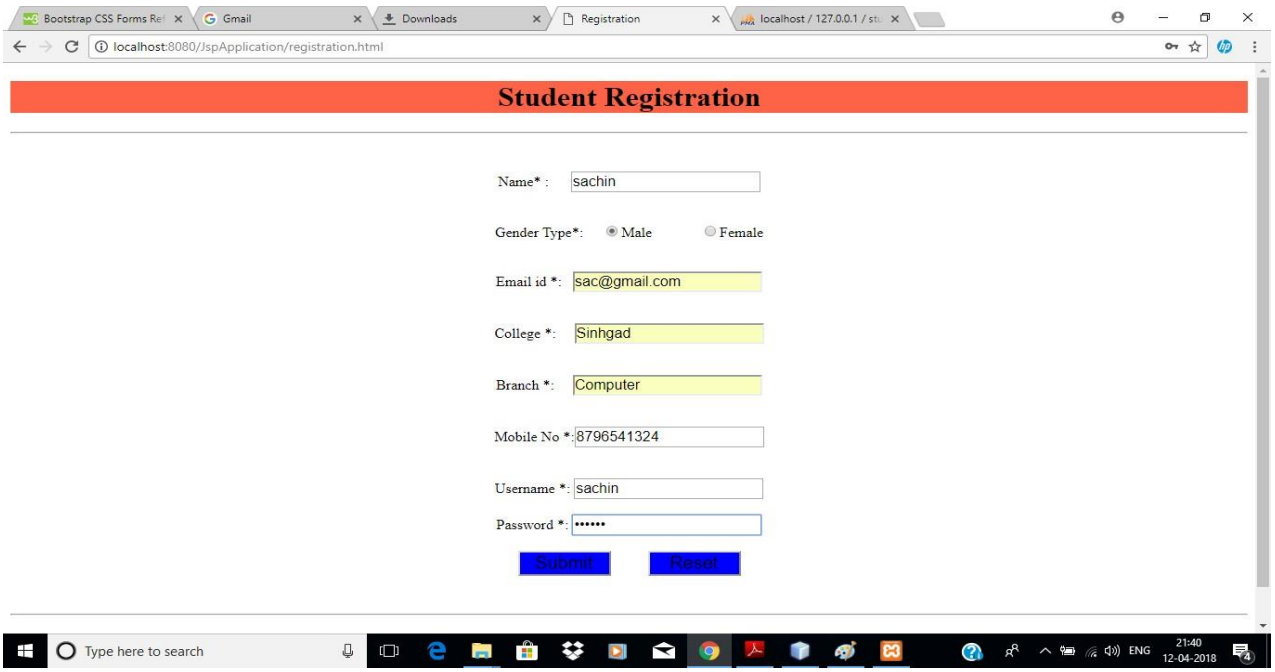
```
            , gender);
            pst.setString(3
            , email);
            pst.setString(4
            , college);
            pst.setString(5
            , branch);
            pst.setString(6
            , mobile);
            pst.setString(7
            , username);
            pst.setString(8
            , pass)
            pst.executeUpdate();
            %>
            alert('Registered Successfully!');
            <jsp:forward page="Signin.html"/>
            <%
}
catch(Exception e)
{
System.out.pri
        ntln(e);
}

%>

</body>
</html>
```

registration.html

```
<!DOCTYPE html>

<html>

   <head>

<title>Registrat
      ion</title>

   </head>

   <body>

      <center><h1 style="background-color:Tomato;">Student Registration</h1></center>

<hr />

<br>

<form  id="fr" action="registration.jsp" method="POST">

<br>

<center> Name* :    <input type="text" name="name" id="name"
placeholder="Entername" style="width:200px;font-size:12pt;color:black; " ></center>

<br><br>

<center>Gender Type*:   <input type="radio" name="gender"  value="male"
>Male      

<input type="radio" name="gender"  value="female" >Female </center><br><br>


<center>Email id *:  <input type="text" name="email" id="email"
placeholder="Enter email"style="width:200px;font-size:12pt;color:black;"
></center>

<br><br>

<center>College *:   <input type="text" name="college"
id="college" placeholder="Enter College" style="width:200px;font-
size:12pt;color:black;" ></center>

<br><br>
```

```
<center>Branch *:   <input type="text" name="branch" id="branch"
placeholder="EnterBranch" style="width:200px;font-size:12pt;color:black;"
></center>

<br><br>

<center>Mobile No *:<input type="text" name="mobile" id="mobile"
placeholder="Enter Mobileno" style="width:200px;font-size:12pt;color:black;"
></center>

<br><br>

<center>Username *: <input type="text" name="username" id="username"
placeholder="Enterusername" style="width:200px;font-size:12pt;"
><br><br></center>

<center>Password *: <input type="password" name="password" id="password"
placeholder="EnterPassword" style="width:200px;font-size:12pt;" ></center><br>


<center><input type="submit" value="Submit" style="width:100px;font-
size:15pt;color:black;background-color:blue;" > 

   

<input type="reset" value="Reset" style="width:100px;font-
size:15pt;color:black; background-color:blue;"></center>

<br><br>

<hr />

<center><p style="font-size:15pt;color:black;">(*) Field indicates field is mandatory!!</p></center>

</form>

 </body>

</html>
```

OUTPUT:


Bootstrap CSS Forms Ref
Gmail
Downloads
Registration
localhost / 127.0.0.1 / stu
localhost:8080/JspApplication/registration.html
Student Registration
Name* : sachin
Gender Type*: Male Female
Email id *: sac@gmail.com
College *: Sinhgad
Branch *: Computer
Mobile No *: 8796541324
Username *: sachin
Password *: ••••••
Submit
Reset
Type here to search
ENG
21:40
12-04-2018

Experiment No: 7

Title	Add dynamic web application essence using PHP, HTML and MySQL.
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.
Prerequisite	PHP, MySQL
CO mapped	CO5, CO^
Hardware required	OS

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 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.

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

- Web Server
- PHP Parser
- Database

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 get 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.

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 tou**tilize 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 includingPHP, PERL, C, C++, JAVA, and so forth.
- MySQL works rapidly and functions admirably even with extensive informationalindexes.
- 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.

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.

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.

Problem Statement:

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

Code:

- **db_connect.php**

```
<?php
  $conn=mysqli_connect('localhost','root','','assignment6');
if(mysqli_connect_errno()){
echo 'Connection error';
  }
?>
```

- **index.php**

2.

```
<?php
session_start();
if(!isset($_SESSION['email']))
  {
    $_SESSION['email']='';
  }
 ?>
<html>
```

```
<head><title>Home Page</title>
<style>
.button {
background-color:white;
color:black; padding:5px;
text-align:center;
padding:15px 30px;
transition-duration:0.4s;
border-radius:4px;
font-size:25px;
}

.top {
background-color:#ff5b5b;
color:white;
padding:5px;
text-align:center;
padding:15px 30px;
font-size:25px;
float:left;
border:none;
width:50%;
}
</style>
</head>
<body style="font-family:Veranda,sans-serif;">
<div class="top_bar">
<a href="index.php"><button class="top">Home</button></a>
<a href="abt.php"><button class="top">About Us</button></a>
</div>
```

<button style="margin-top:10%;" class="button s_in">Sign-in</button>

<button class="button s_up">Sign-up</button>

</body>

</html>

- **s_in.php**

<html>

<head><title>Sign-in to your account</title>

<style>

.header {

margin-top:5px;

margin-bottom:10%;

border-left:2px solid #4CAF50;

border-top:2px solid #4CAF50;

border-right:2px solid #008CBA;

border-bottom:2px solid #008CBA;

padding:8px;

font-size:30px;

text-align:center;

}

.top {

background-color:#ff5b5b;

color:white;

```
text-align:center;
padding:15px 30px;
font-size:25px;
float:left;
border:none;
width:50%;
margin-bottom:10px;
}

.button {
background-color:white;
color:black;
text-align:center;
padding:10px 15px;
transition-duration:0.4s;
border-radius:4px;
font-size:20px;
}
</style>

</head>
<body style="font-family:Veranda,sans-serif;">
<div class="top_bar">
<a href="index.php"><button class="top">Home</button></a>
<a href="abt.php"><button class="top">About Us</button></a>
</div>
<center>
<div style="clear:left;" class="header">Sign-in to your Account</div>
```

<form action="login_validate.php" method="POST">

<table>

<tr><td>Email:</td><td style="text-align:center;"><input type="text" id="email" name="email" placeholder="Enter your email id" class="login"></td></tr>

<tr><td>Password:</td><td style="text-align:center;"><input type="password" id="pword" name="pword" placeholder="Enter your password" class="login"></td></tr>

</table>

<button type="submit" name="submit" class="button s_in">Login</button>

<button type="reset" name="reset" class="button reset">Reset</button>

</form>

</body>

</html>

- **login_validate.php**

```
<?php

require 'db_connect.php';

session_start();

  $email=$_POST['email'];

  $pword=$_POST['pword'];

  $query="select * from registration_details where email='$email' AND pword='$pword' ";

$result=mysqli_query($conn,$query);

if(mysqli_num_rows($result)>0)

  {

    $_SESSION['email']=$email;

    ?>

<script>location.href="s_info.php";</script>
```

```
<?php
  }
else {
    ?>
<script>alert('Please check email id or password');
location.href="s_in.php";
</script>
<?php
  }
?>
```

- **s_up.php**

```
<html>
<head><title>Sign-up for a new account</title>
<style>
.top {
background-color:#ff5b5b;
color:white;
text-align:center;
padding:15px 30px;
font-size:25px;
float:left;
border:none;
width:50%;
margin-bottom:10px;
}

.header {
margin-top:5px;
```

```
margin-bottom:5%;
border-left:2px solid #4CAF50;
border-top:2px solid #4CAF50;
border-right:2px solid #008CBA;
border-bottom:2px solid #008CBA;
padding:8px;
font-size:30px;
text-align:center;
}

.button {
background-color:white;
color:black;
padding:10px 15px;
text-align:center;
border-radius:4px;
transition-duration:0.4s;
font-size:20px;
}
</style>

</head>
<body style="font-family:Veranda,sans-serif;">
<div class="top_bar">
<a href="index.php"><button class="top">Home</button></a>
<a href="abt.php"><button class="top">About Us</button></a>
</div>
<center>
```

<div style="clear:left;"class="header">Sign-up for a new account

<form id="sign_up" action="reg_user.php" method="POST">

<table>

<tr><td>First Name:</td><td style="text-align:center;"><input type="text" id="fname" name="fname" placeholder="Saad" class="info"></td></tr>

<tr><td>Last Name:</td><td style="text-align:center;"><input type="text" id="lname" name="lname" placeholder="Shaikh" class="info"></td></tr>

<tr><td>Gender:</td><td style="text-align:center;"><input type="radio" name="gender" id="mradio" value="Male" checked>Male   <input type="radio" name="gender" id="fradio" value="Female">Female</td></tr>

<tr><td>Age:</td><td style="text-align:center;"><input type="text" id="age" name="age" placeholder="20" class="info"></td></tr>

<tr><td>Mobile No.:</td><td style="text-align:center;"><input type="text" id="mno" name="mno" placeholder="123456789" class="info"></td></tr>

<tr><td>Address:</td><td style="text-align:center;"><input type="text" id="add" name="add" placeholder="Ganga Savera" class="info"></td></tr>

<tr><td>Email ID:</td><td style="text-align:center;"><input type="email" id="email" name="email" placeholder="saad@gmail.com" class="info"></td></tr>

<tr><td>Password:</td><td style="text-align:center;"><input type="text" id="pword" name="pword" placeholder="********" class="info"></td></tr>

</table>

<button type="submit" class="button register">Register</button>

<button type="reset" class="button reset">Reset</button>

```
</center>
</form>
</div>
</body>
</html>
```

- **reg_user.php**

```
<?php
require 'db_connect.php';

  $fname=$_POST['fname'];
  $lname=$_POST['lname'];
  $gender=$_POST['gender'];
  $age=$_POST['age'];
  $mno=$_POST['mno'];
  $add=$_POST['add'];
  $email=$_POST['email'];
  $pword=$_POST['pword'];

  $query="insert into
registration_detailsvalues('$fname','$lname','$gender','$age','$mno','$add','$email','$pword')";

if(mysqli_query($conn,$query))
  {
    ?>
<script>alert('Registered Successfully!');
location.href="s_in.php";
</script>
<?php
```

```
  }
else {
echo 'Error: '.mysqli_error($conn);
  }
?>
```

- **s_info.php**

```
<?php
require 'db_connect.php';
session_start();
if($_SESSION['email']=='')
  {
    ?>
<script>alert('Please login to the system first!');
location.href="s_in.php";</script>
<?php
  }
else{
    $query="select * from s_info";
    $result=mysqli_query($conn,$query);

    $info=mysqli_fetch_all($result,MYSQLI_ASSOC);
mysqli_free_result($result);
mysqli_close($conn);
  }
?>

<html>
<head><title>Student Information</title>
```

```
<style>
.header {
margin-top:5px;
margin-bottom:5%;
border-left:2px solid #4CAF50;
border-top:2px solid #4CAF50;
border-right:2px solid #008CBA;
border-bottom:2px solid #008CBA;
padding:8px;
font-size:30px;
text-align:center;
}

.button {
background-color:white;
color:black;
text-align:center;
padding:10px 15px;
transition-duration:0.4s;
border-radius:4px;
font-size:20px;
}

tr:nth-child(even) {background-color:#efefef;}

.tablebtn:link, .tablebtn:visited {
background-color: white;
color: black;
```

```
padding: 5px 5px;
text-align: center;
text-decoration: none;
display: inline-block;
transition-duration:0.4s;
border-radius:4px;
border:1px solid #ff5b5b;
}
</style>
</head>
<body style="font-family:Veranda,sans-serif;">
<center>
<div class="header">Student Information</div>
<br><br><br>
<table width="100%">
<tr>
<thcolspan="6">C3 Student Information</th>
</tr>
<tr style="background-color:#ff5b5b;color:white;">
<th>Roll No.</th>
<th>First Name</th>
<th>Last Name</th>
<th>Gender</th>
<th>Mobile No.</th>
<th>Action</th>
</tr>
<?phpforeach($info as $row){ ?>
<tr>
```

```
<td><?php echo $row['rollno']; ?></td>

<td><?php echo $row['first_name']; ?></td>

<td><?php echo $row['last_name']; ?></td>

<td><?php echo $row['gender']; ?></td>

<td><?php echo $row['mobile_number']; ?></td>

<td>

<a href="update.php?rollno=<?php echo $row['rollno']; ?>" class="tablebtn">Update</a>

    

<a href="delete.php?rollno=<?php echo $row['rollno']; ?>" class="tablebtn">Delete</a>

</td>

</tr>

<?php }?>

</table>

<a href="insert.php"><button type="button" name="insert" class="button insert" style="margin-top:5%;">Insert</button></a>

<a href="logout.php"><button style="position:absolute; top:12%; right:3%;" class="button logout">Logout</button></a>

</center>

</body>

</html>
```

- insert.php

```
<?php

session_start();

if($_SESSION['email']=='')

  {

    ?>

<script>alert('Please login to the sytsem first!');

location.href="s_in.php";</script>
```

```
<?php
  }
 ?>
<html>
<head><title>Insert Student Information</title>
<style>
.header {
margin-top:5px;
margin-bottom:5%;
border-left:2px solid #4CAF50;
border-top:2px solid #4CAF50;
border-right:2px solid #008CBA;
border-bottom:2px solid #008CBA;
padding:8px;
font-size:30px;
text-align:center;
}

.button {
background-color:white;
color:black;
text-align:center;
padding:10px 15px;
transition-duration:0.4s;
border-radius:4px;
font-size:20px;
```

```
}

</style>

</head>

<body style="font-family:Veranda,sans-serif;">

<center>


<div class="header">Insert Records to Table</div>

<br><br><br>


<form action="insert_rec.php" method="POST">


<table>

<tr><td>First Name:</td><td style="text-align:center;"><input type="text" id="fname" name="fname" placeholder="Enter first name" class="addrec"></td></tr>


<tr><td>Last Name:</td><td style="text-align:center;"><input type="text" id="lname" name="lname" placeholder="Enter last name" class="addrec"></td></tr>


<tr><td>Gender:</td><td style="text-align:center;"><input type="text" id="gender" name="gender" placeholder="Enter the gender" class="addrec"></td></tr>


<tr><td>Mobile No.:</td><td style="text-align:center;"><input type="text" id="mno" name="mno" placeholder="Enter the mobile no" class="addrec"></td></tr>

</table>

<br><br>


<button type="submit" name="submit" class="button insert">Insert</button>

    
```

```
<button type="reset" name="reset" class="button reset">Reset</button>

</form>

</center>

<a href="s_info.php"><button style="position:absolute; top:12%; left:3%;" class="button back">Back</button></a>

</body>

</html>
```

- insert_rec.php

```
<?php

require 'db_connect.php';

session_start();


if($_SESSION['email']=='')

  {

    ?>

<script>alert('Please login to the system first!');

location.href="s_in.php";</script>

<?php

  }

else{

    $fname=$_POST['fname'];

    $lname=$_POST['lname'];

    $gender=$_POST['gender'];

    $mno=$_POST['mno'];


if($fname!='' && $lname!='' && $gender!='' && $mno!='')
```

```
    {
      $query="insert into s_info(first_name,last_name,gender,mobile_number)
values('$fname','$lname','$gender','$mno')";


if(mysqli_query($conn,$query))
      {
        ?>
<script>alert('Inserted Successfully!');
location.href="s_info.php";
</script>
<?php
      }
else {
echo 'Error: '.mysqli_error($conn);
      }
    }
else {
      ?>
<script>alert('Please fill in all fields!');
location.href="insert.php";</script>
<?php
    }
  }
 ?>
```

- update.php

```
<?php
```

```
require 'db_connect.php';
session_start();

if($_SESSION['email']=='')
  {
    ?>
<script>alert('Please login to the system first!');
location.href="s_in.php";</script>
<?php
  }
else{
    $rollno=$_GET['rollno'];
    $query="select * from s_info where rollno='$rollno'";
    $result=mysqli_query($conn,$query);
    $info=mysqli_fetch_assoc($result);
  }
 ?>
<html>
<head><title>Update Student Information</title>
<style>
.header {
margin-top:5px;
margin-bottom:5%;
border-left:2px solid #4CAF50;
border-top:2px solid #4CAF50;
border-right:2px solid #008CBA;
```

```
border-bottom:2px solid #008CBA;

padding:8px;

font-size:30px;

text-align:center;

}

.button {

background-color:white;

color:black;

text-align:center;

padding:10px 15px;

transition-duration:0.4s;

border-radius:4px;

font-size:20px;

}

</style>

</head>

<body style="font-family:Veranda,sans-serif;">

<center>

<div class="header">Update Existing Record</div>

<br><br><br>

<form action="update_rec.php?rollno=<?php echo $rollno; ?>" method="POST">

<table>

<tr><td>First Name:</td><td style="text-align:center;"><input type="text" id="fname"
name="fname" value="<?php echo $info['first_name']; ?>" class="updtrec"></td></tr>


<tr><td>Last Name:</td><td style="text-align:center;"><input type="text" id="lname"
name="lname" value="<?php echo $info['last_name']; ?>" class="updtrec"></td></tr>
```

<tr><td>Gender:</td><td style="text-align:center;"><input type="text" id="gender" name="gender" value="<?php echo $info['gender']; ?>" class="updtrec"></td></tr>

<tr><td>Mobile No.:</td><td style="text-align:center;"><input type="text" id="mno" name="mno" value="<?php echo $info['mobile_number']; ?>" class="updtrec"></td></tr>

</table>

<button type="submit" name="submit" class="button update">Update</button>

<button type="reset" name="reset" class="button reset">Reset</button>

</form>

<button style="position:absolute; top:12%; left:3%;" class="button back">Back</button>

</body>

</html>

- update_rec.php

<?php

require 'db_connect.php';

session_start();

if($_SESSION['email']=='')

{

?>

<script>alert('Please login to the system first!');

```
location.href="s_in.php";</script>
<?php
  }
else {
    $rollno=$_GET['rollno'];
    $fname=$_POST['fname'];
    $lname=$_POST['lname'];
    $gender=$_POST['gender'];
    $mno=$_POST['mno'];

if($fname!='' && $lname!='' && $gender!='' && $mno!='')
    {
      $query="update s_info set
first_name='$fname',last_name='$lname',gender='$gender',mobile_number='$mno' where
rollno='$rollno'";
if(mysqli_query($conn,$query))
      {
        ?>
<script>alert('Updated Successfully!');
location.href="s_info.php";
</script>
<?php
      }
else {
echo 'Error: '.mysqli_error($conn);
      }
    }
else {
```

```
    ?>
<script>alert('Please fill in all fields!');
location.href="update.php?rollno=<?php echo $rollno; ?>";</script>
<?php
   }
  }
 ?>
```

- delete.php

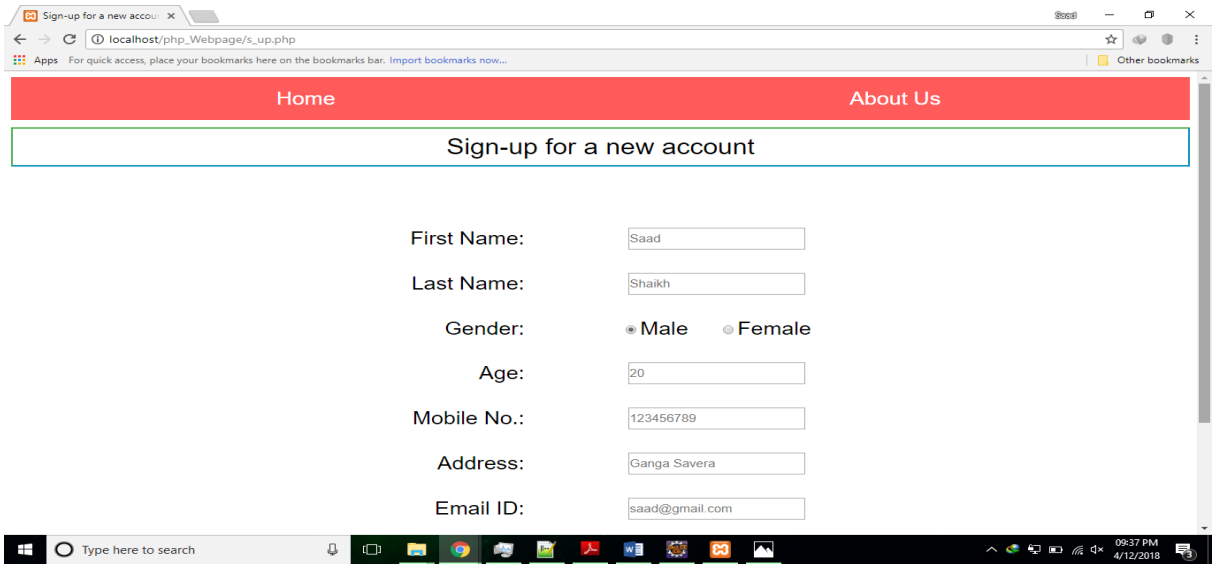
```
<?php
require 'db_connect.php';
session_start();

if($_SESSION['email']=='')
  {
   ?>
<script>alert('Please login to the system first!');
location.href="s_in.php";</script>
<?php
  }
else {
   $rollno=$_GET['rollno'];
   $query="delete from s_info where rollno='$rollno'";

if(mysqli_query($conn,$query))
   {
    ?>
```

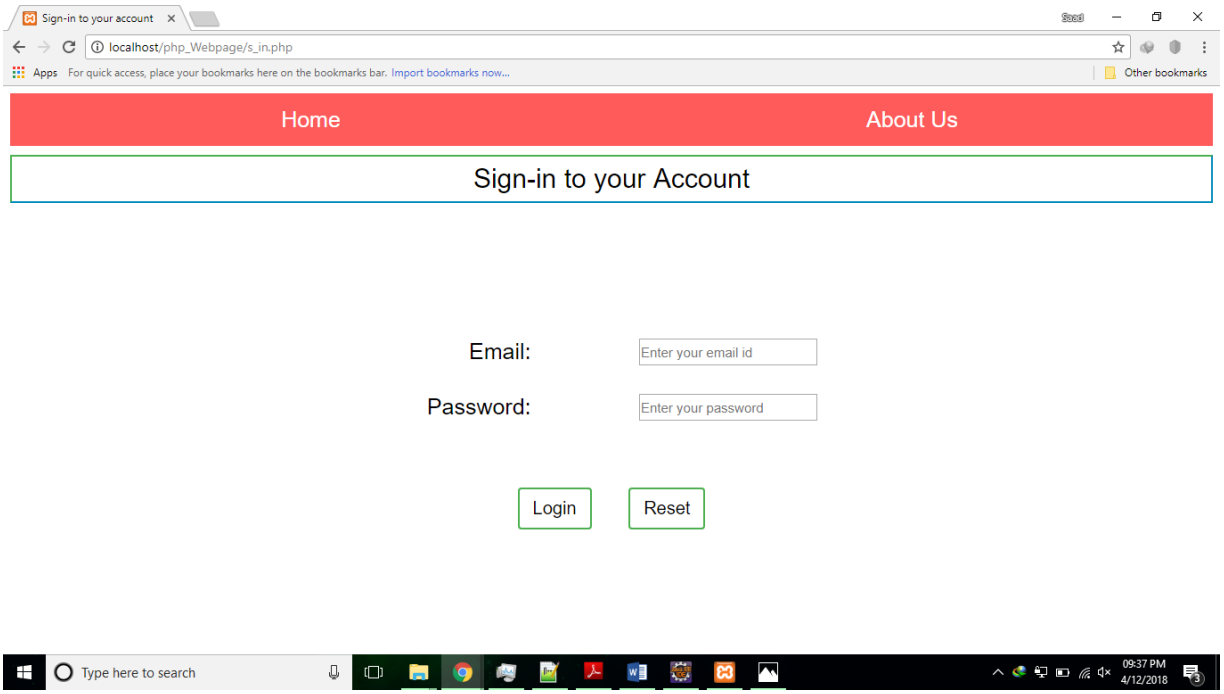
```
<script>alert('Deleted

Successfully!');

location.href="s_info.p

hp";

</script>
<? Php
else

 {

echo 'Error: '.mysqli_error($conn);

}

  }

 ?>
```

- logout.php

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

<script>location.href="index.php";</script>
```

Output:


Sign-up for a new accou
localhost/php_Webpage/s_up.php
Apps
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
Other bookmarks
Home
About Us
Sign-up for a new account
First Name:
Saad
Last Name:
Shaikh
Gender:
Male
Female
Age:
20
Mobile No.:
123456789
Address:
Ganga Savera
Email ID:
saad@gmail.com
Type here to search
09:37 PM
4/12/2018


Sign-in to your account
localhost/php_Webpage/s_in.php
Apps
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
Other bookmarks
Home
About Us
Sign-in to your Account
Email:
Enter your email id
Password:
Enter your password
Login
Reset
Type here to search
09:37 PM
4/12/2018

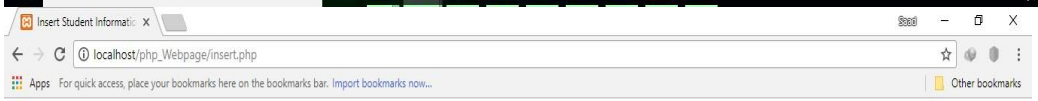
Student Information

Logout

C3 Student Information

Roll No.	First Name	Last Name	Gender	Mobile No.	Action	
1	Saad	Shaikh	Male	123456	Update	Delete
2	Prashant	Wakchaure	Male	456789	Update	Delete
3	Owais	Shaikh	Male	123789	Update	Delete
4	Ankit	Hiremath	Male	456012	Update	Delete

Insert



Insert Records to Table

Back

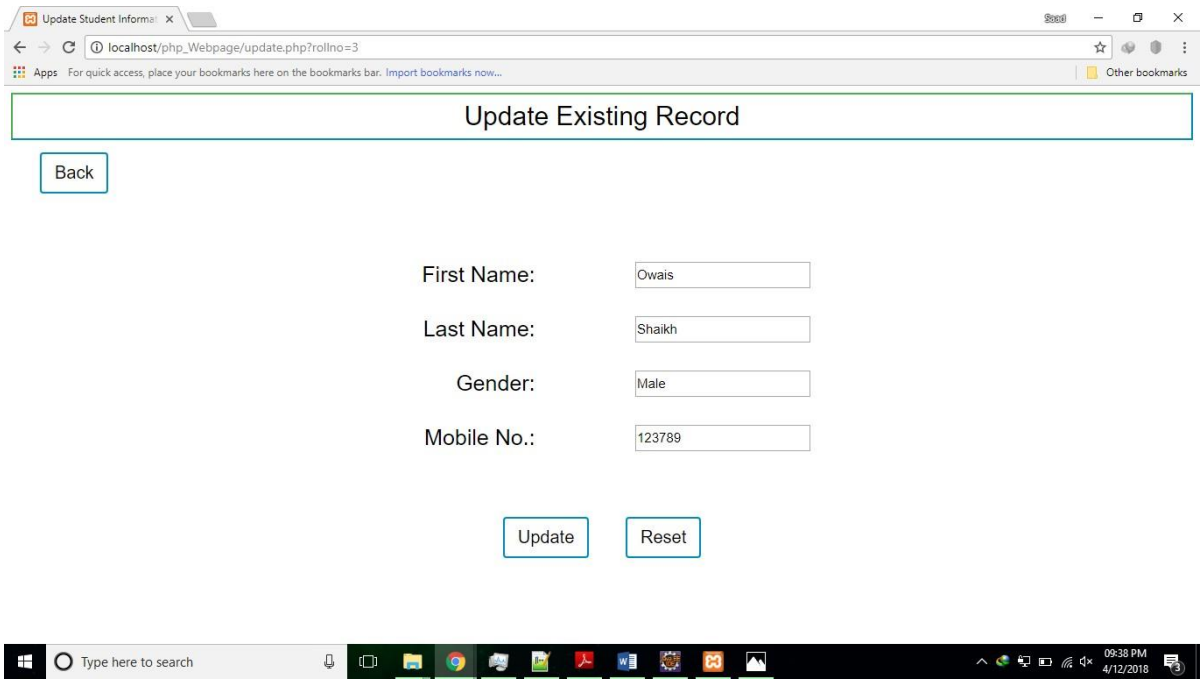
First Name: Enter first name

Last Name: Enter last name

Gender: Enter the gender

Mobile No.: Enter the mobile no

Insert Reset


Update Student Informa
localhost/php_Webpage/update.php?rollno=3
Apps
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
Other bookmarks
Update Existing Record
Back
First Name:
Owais
Last Name:
Shaikh
Gender:
Male
Mobile No.:
123789
Update
Reset
Type here to search
09:38 PM
4/12/2018

Experiment No: 8

Title	Add dynamic web application essence using PHP, AJAX and MySQL.
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
Prerequisite	PHP, MySQL, AJAX
CO mapped	CO5, CO^
Hardware required	OS

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 synchronoussolicitations.

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 andexperts.

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 itis quick.

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

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

TECHNOLOGY/TOOL:

AJAX, PHP and MySQL

CONCLUSION/ANALYSIS:

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

CODE:

Problem Statement: Design and develop dynamic web application using PHP,AJAX and MySQL as a backend for employee data with insert and view operations.

Code:

- **getvalues.php**

```
<!DOCTYPE html>
```

```
<html>

<head>

<title></title>

</head>

<body>

<?php

$role=$_GET['p']; if($role==1)
{

$sel_role="Developer";

} else if($role==2)

{

$sel_role="Tester";

} else if($role==3)

{

$sel_role="Technician";

} else if($role==4)

{

$sel_role="Salesperson";

}

$conn=mysqli_connect("localhost","root","","ajax");


$result=mysqli_query($conn,"select * from employee_data where role='$sel_role'");


echo "<table width='50%'>

<tr style='background-color:#ff5b5b;color:white;'>

<th>id</th>
```

```
<th>name</th>

<th>salary</th>

<th>email</th>

</tr>";

while ($row=mysqli_fetch_array($result))

{

echo "<tr>";

echo "<td>".$row['id']."</td>";

echo "<td>".$row['name']."</td>";

echo "<td>".$row['salary']."</td>";

echo "<td>".$row['email']."</td>"; echo "</tr>";
} echo "</table>";

?>

</body>

</html>
```

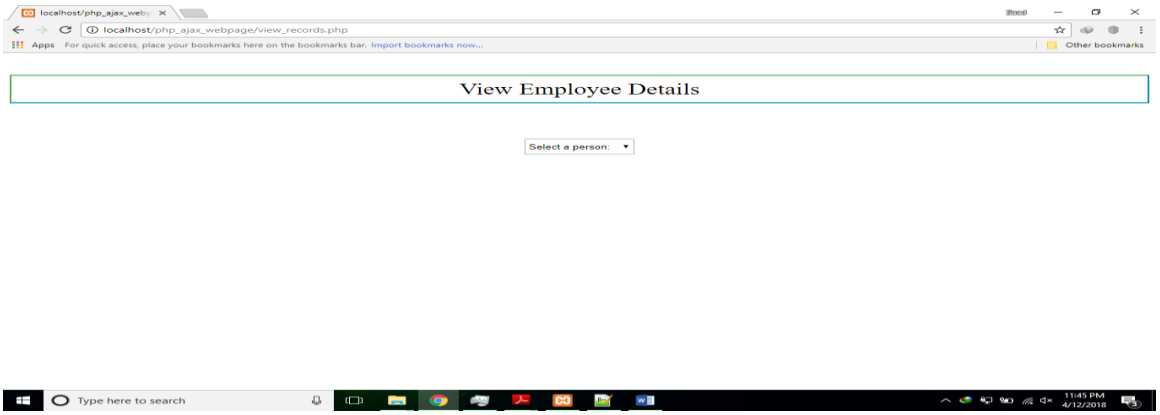
- view_records.php

```
<!DOCTYPE html>

<html>

<head>

<title></title>

<script type="text/javascript"> functionshowUser(role_val)
{

varxmlhttp = new XMLHttpRequest(); xmlhttp.onreadystatechange=function(){
if(this.readyState==4 &&this.status==200)

{

console.log('Result:'+this.responseText);
document.getElementById("table1").innerHTML=this.responseText;
```

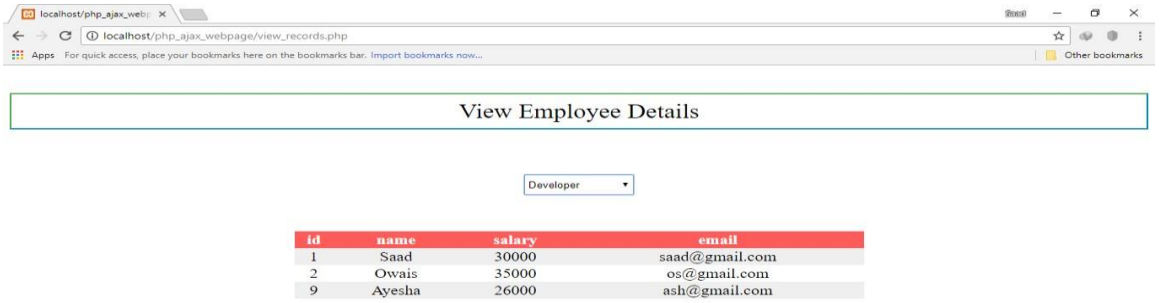
```
}

};

xmlhttp.open("GET","getvalues.php?p="+role_val,true); xmlhttp.send();
}

</script>


<style type="text/css">

.header { margin-top:3%;
margin-bottom:5%;

border-left:2px solid #4CAF50; border-top:2px solid #4CAF50; border-right:2px solid
#008CBA; border-bottom:2px solid #008CBA; padding:8px;
font-size:30px; text-align:center;

}


table {

text-align:center;

border-collapse:collapse; font-size:20px;
}


tr:nth-child(even) {background-color:#efefef;}


select {

width:130px; height:30px;
}

</style>

</head>

<body>

<center>

<div class="header">View Employee Details</div>

<form>
```

```
<div>
<select name="users" onchange="showUser(this.value)">
<option value="select">Select a person:</option>
<option value="1">Developer</option>
<option value="2">Tester</option>
<option value="3">Technician</option>
<option value="4">Salesperson</option>
</select>
</div>
</form>
<br><br><br>
<div id="table1">
</div>
</center>
</body>
</html>
```

Output:


localhost/php_ajax_webpage/view_records.php
Apps
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
Other bookmarks
View Employee Details
Select a person:


Type here to search
11:45 PM
4/12/2018


localhost/php_ajax_webpage/view_records.php
Apps
For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...
Other bookmarks
View Employee Details
Developer
id name salary email
1 Saad 30000 saad@gmail.com
2 Owais 35000 os@gmail.com
9 Ayesha 26000 ash@gmail.com


Type here to search
11:46 PM
4/12/2018

View Employee Details

Tester

id	name	salary	email
4	Prashant	28000	prash@gmail.com
5	Vikash	27000	vikash@gmail.com

View Employee Details

Technician

id	name	salary	email
3	Sachita	25000	sachita@gmail.com
8	Komal	24000	komal@gmail.com

Experiment No:9

Title	Design and develop any web application using Struts framework.
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
Prerequisite	Struct version 2,
CO mapped	CO3, CO4
Hardware required	OS

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.

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.

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 the
name	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 tagthe
scope	scope to search for the bean named in the name attribute

TECHNOLOGY/TOOL:

1) Eclipse IDE
2) Apache Tomcat 7.0 or higher

CONCLUSION/ANALYSIS:

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

CODE:

Problem Statement: Create a login module for the web application using strutsframework.

Program:

login.jsp

```
<%@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>Login</h1>
    <form action="loginform.do"
      method="post"> username <input
      type="text" name="uname"/><br/>
      password <input type="password" name="upass"/><br/>
      <input type="submit" value="Login"/>
    </form>
  </body>
</html>
```

success.jsp

```
<%@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>Success Page</h1>
    </body>
</html>
```

failure.jsp

```
<%@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>Failure page</h1>
    </body>
</html>
```

loginbean.java

```
package com.myapp.struts;

import javax.servlet.http.HttpServletRequest;

import org.apache.struts.action.ActionErrors;
 import org.apache.struts.action.ActionMapping;
 import org.apache.struts.action.ActionMessage;
/**

*

* @author hp

*/

public class loginbean extends org.apache.struts.action.ActionForm {

private String uname;
```

```
private String upass;

public String getUname() { return uname;
}
public void setUname(String uname) { this.uname = uname;
}
public String getUpass() { return upass;
}
public void setUpass(String upass) { this.upass = upass;
}
public loginbean() { super();
// TODO Auto-generated constructor stub

}
public ActionErrors validate(ActionMapping mapping, HttpServletRequest request) { ActionErrors
errors = new ActionErrors();

return errors;

}

}
```

loginform.java

```
package com.myapp.struts;

import javax.servlet.http.HttpServletRequest;
 import javax.servlet.http.HttpServletResponse;
import org.apache.struts.action.ActionForm;
 import org.apache.struts.action.ActionForward;
 import org.apache.struts.action.ActionMapping;

public class loginform extends org.apache.struts.action.Action {

/* forward name="success" path="" */ private static final String SUCCESS = "success"; private static
final String FAILURE = "failure";

@Override

public ActionForward execute(ActionMapping mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response)
throws Exception {

loginbean lb = (loginbean)form;

if(lb.getUname().equals("abc") && lb.getUpass().equals("123")) return
mapping.findForward(SUCCESS);
else

return mapping.findForward(FAILURE);

}

}
```

structs-config.xml

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

<!DOCTYPE struts-config PUBLIC

"-//Apache Software Foundation//DTD Struts Configuration 1.3//EN"
"http://jakarta.apache.org/struts/dtds/struts-config_1_3.dtd">

<struts-config>

<form-beans>

<form-bean name="loginbean" type="com.myapp.struts.loginbean"/>

</form-beans>

<global-exceptions>

</global-exceptions>

<global-forwards>

<forward name="failure" path="/failure.jsp"/>

<forward name="success" path="/success.jsp"/>

<forward name="welcome" path="/Welcome.do"/>

</global-forwards>

<action-mappings>

<action name="loginbean" path="/loginform" scope="request" type="com.myapp.struts.loginform"
validate="false"/>

<action path="/Welcome" forward="/welcomeStruts.jsp"/>

</action-mappings>
<controller processorClass="org.apache.struts.tiles.TilesRequestProcessor"/>
<message-resources parameter="com/myapp/struts/ApplicationResource"/>
<plug-in className="org.apache.struts.tiles.TilesPlugin" >
<set-property property="definitions-config" value="/WEB-INF/tiles-defs.xml" />
<set-property property="moduleAware" value="true" />
</plug-in>

<!-- ========================== Validator plugin
================================= -->

<plug-in className="org.apache.struts.validator.ValidatorPlugIn">

<set-property property="pathnames"
value="/WEB-INF/validator-rules.xml,/WEB-INF/validation.xml"/>

</plug-in>
```

</struts-config>

web.xml

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

<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd">

<servlet>

<servlet-name>action</servlet-name>

<servlet-class>org.apache.struts.action.ActionServlet</servlet-class>

<init-param>

<param-name>config</param-name>

<param-value>/WEB-INF/struts-config.xml</param-value>

</init-param>

<init-param>

<param-name>debug</param-name>

<param-value>2</param-value>

</init-param>

<init-param>

<param-name>detail</param-name>

<param-value>2</param-value>

</init-param>

<load-on-startup>2</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>action</servlet-name>

<url-pattern>*.do</url-pattern>

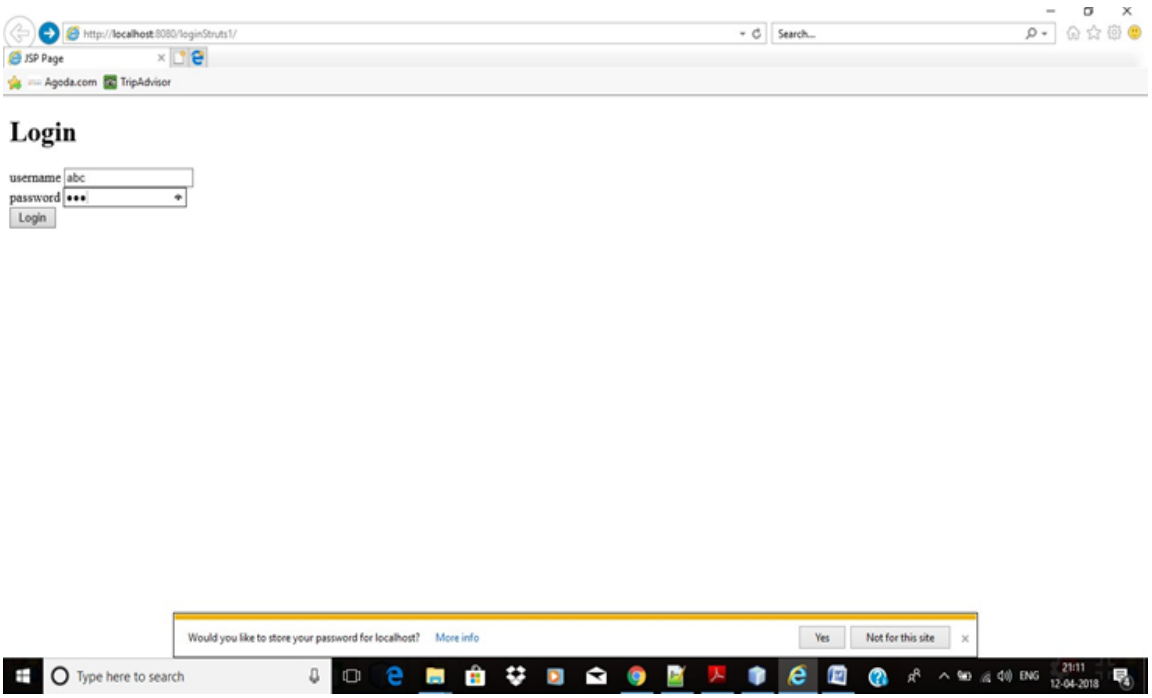
</servlet-mapping>

<session-config>

```
<session-timeout> 30
</session-timeout>

</session-config>

<welcome-file-list>

<welcome-file>login.jsp</welcome-file>

</welcome-file-list>

<jsp-config>

<taglib>

<taglib-uri>/WEB-INF/struts-bean.tld</taglib-uri>

<taglib-location>/WEB-INF/struts-bean.tld</taglib-location>

</taglib>

<taglib>

<taglib-uri>/WEB-INF/struts-html.tld</taglib-uri>

<taglib-location>/WEB-INF/struts-html.tld</taglib-location>

</taglib>

<taglib>

<taglib-uri>/WEB-INF/struts-logic.tld</taglib-uri>

<taglib-location>/WEB-INF/struts-logic.tld</taglib-location>

</taglib>

<taglib>

<taglib-uri>/WEB-INF/struts-nested.tld</taglib-uri>

<taglib-location>/WEB-INF/struts-nested.tld</taglib-location>

</taglib>

<taglib>

<taglib-uri>/WEB-INF/struts-tiles.tld</taglib-uri>

<taglib-location>/WEB-INF/struts-tiles.tld</taglib-location>

</taglib>

</jsp-config>
```

</web-app>

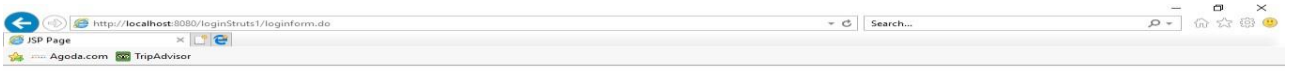
Output:



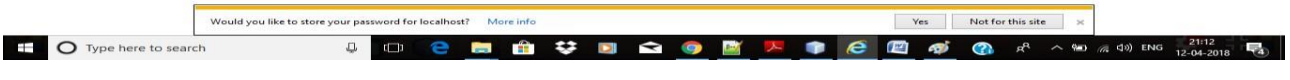
Success Page

Login

username abc
password
Login



Failure page



Experiment No: 10

Title	Design and develop any web application using AngularJS.
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.
Prerequisite	Angular JS
CO mapped	
Hardware required	OS

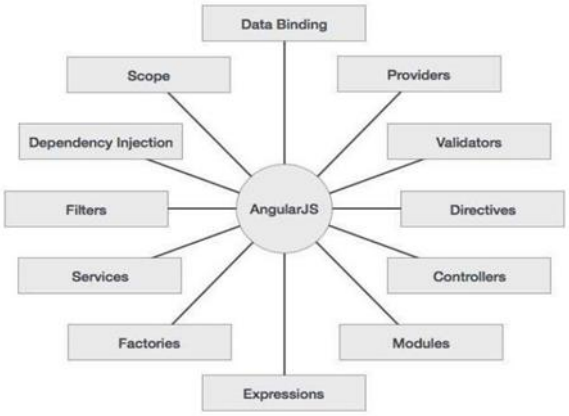
THEORY-CONCEPT:

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 tokeepup web applications.

Core Features:



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 richand 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 dothe business handling.

CONCLUSION/ANALYSIS:

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

CODE:

Problem Statement: Create an application for Bill Payment Record usingAngularJS.

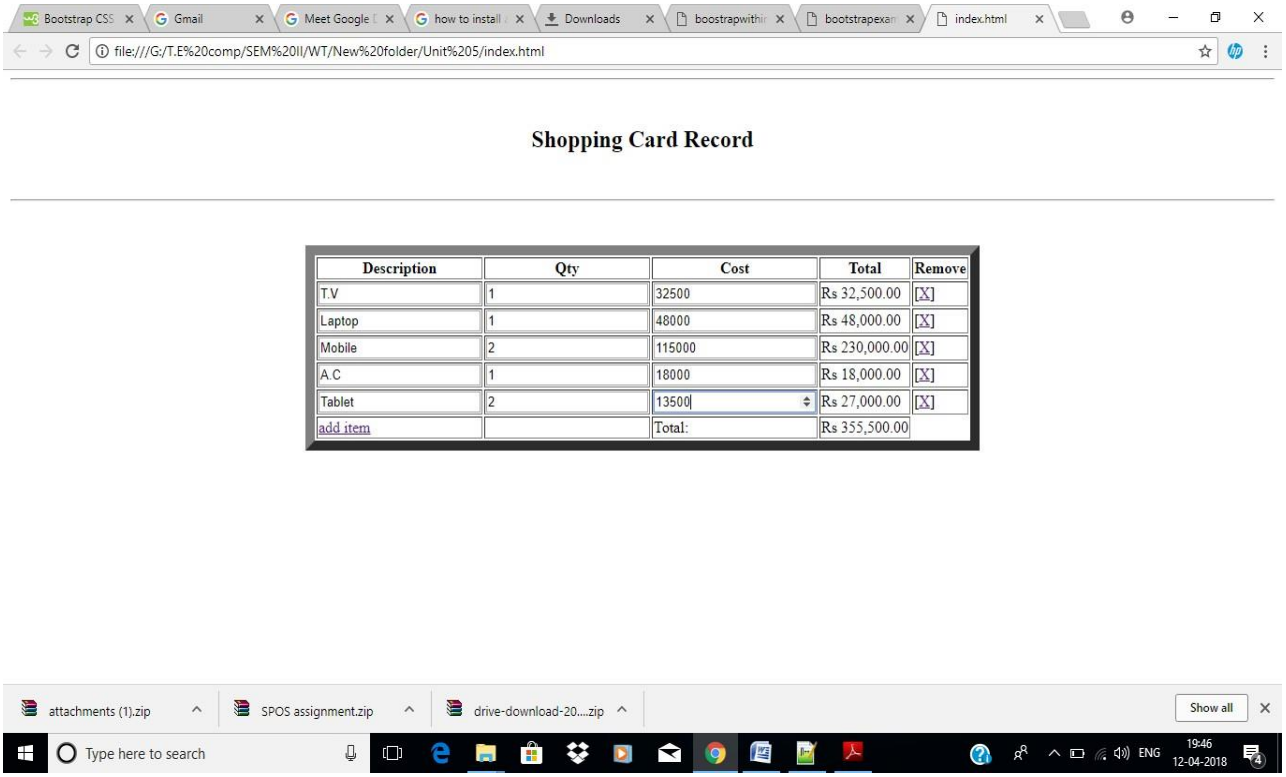
Program:

```
<html>

<head>

<script src="angular.js"></script>

</head>

<body>

<hr><br>

<center><h2>Shopping Card Record</h2></center>
<div data-ng-app="myapp" data-ng-controller="CartForm">
<center><br>

<hr>

<br><br>

<table class="table" >

<table border="10" >

<tr>
```

```
<th>Description</th>
<th>Qty</th>
<th>Cost</th>
<th>Total</th>
<th>Remove</th>
</tr>
<tr ng:repeat="item in cart.items">
<td><input type="text" ng:model="item.description"></td>
<td><input type="number" ng:model="item.qty" ng:required></td>
<td><input type="number" ng:model="item.cost" ng:required></td>
<td>{{item.qty * item.cost | currency:'Rs '}}</td>
<td>[<a href ng:click="removeItem($index)">X</a>]</td>
</tr>
<tr>
<td><a href ng:click="addItem()">add item</a></td>
<td></td>
<td>Total:</td>
<td>{{total() | currency:'Rs '}}</td>
</tr>
</table>
</center>
</div>

<script>
var
app=angular.module('myapp',[]);
app.controller('CartForm',
function($scope)
{
$scope.cart =
```

```
{
items: [{ qty: '', description: '', cost: ''}]
    }
$scope.addItem = function()
{
$scope.cart.items.push({ qty: '', description: '', cost: ''});
}
$scope.removeItem = function(index)
{
$scope.cart.items.splice(index, 1);
}
$scope.total = function()
{
var total = 0;
angular.forEach($scope.cart.items, function(item)
{
total += item.qty * item.cost;
})
return total;
}
}
</script>
 </body>
 </html>
```

Output:


Bootstrap CSS
Gmail
Meet Google
how to install
Downloads
boostrapwithin
bootstrapexam
index.html
file:///G:/T.E%20comp/SEM%20II/WT/New%20folder/Unit%205/index.html
Shopping Card Record
Description
Qty
Cost
Total
Remove
T.V
1
32500
Rs 32,500.00
[X]
Laptop
1
48000
Rs 48,000.00
[X]
Mobile
2
115000
Rs 230,000.00
[X]
A.C
1
18000
Rs 18,000.00
[X]
Tablet
2
13500
Rs 27,000.00
[X]
add item
Total:
Rs 355,500.00
attachments (1).zip
SPOS assignment.zip
drive-download-20....zip
Show all
Type here to search
ENG
19:46
12-04-2018

Experiment No: 11

Title	Mini Project
Problem statement	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.
Prerequisite	All Basic concepts
CO mapped	
Hardware required	OS