

EXCEL ENGINEERING COLLEGE

KOMARAPALAYAM –637 303



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CS8661 - INTERNET PROGRAMMING LABORATORY

VI SEMESTER - R 2017

REFERENCE MANUAL

PREPARED BY

D.VADUGANATHAN, AP/CSE

T.ARUNKUMAR, AP/CSE

EXCEL ENGINEERING COLLEGE

VISION

To create competitive human resources in the fields of engineering for the benefit of society to meet global challenges.

MISSION

- To provide a conducive ambience for better learning and to bring creativity in the students.
- To develop sustainable environment for innovative learning to serve the needy.
- To meet global demands for excellence in technical education.
- To train young minds with values, culture, integrity, innovation and leadership.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION

To create better quality technical engineers in computer science and engineering with ethically strong values which cater local and global needs of the society.

MISSION

- To instill quality in engineering education that demands excellence.
- To initiate desires among the students to work in close cooperation and collaboration with industry and professional bodies.
- To train the students for developing software and novel software systems.
- To create ambience for taking initiatives towards entrepreneurship and lifelong learning.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

1. To provide fundamental knowledge to formulate, solve and analyze engineering problems and pursue higher studies.
2. To develop the ability of the students in comprehending, analyzing and synthesizing data in order to design software and create novel software systems.
3. To inculcate effective communication skills, team skills, professional and ethical attitude in the students for enabling them to relate engineering issues with social issues in a broader context.
4. To provide students managerial and leadership skills so as to make them successfully employed and to demonstrate a pursuit of lifelong learning in multidisciplinary environment.

PROGRAMME OUTCOMES (POs)

1. **Engineering Knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.
2. **Problem Analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **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.
4. **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.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling of complex engineering activities with an understanding of the limitations.
6. **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 professional engineering practice.
7. **Environment and Sustainability**: Understand the impact of the professional engineering solutions to societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **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.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering 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.
12. **Lifelong learning**: Recognize the need for and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

OBJECTIVES:

- To be familiar with Web page design using HTML/XML and style sheets
- To be exposed to creation of user interfaces using Java frames and applets.
- To learn to create dynamic web pages using server side scripting.
- To learn to write Client Server applications.
- To be familiar with the PHP programming.
- To be exposed to creating applications with AJAX

LIST OF EXPERIMENTS

1. Create a web page with the following using HTML
 - a. To embed a map in a web page
 - b. To fix the hot spots in that map
 - c. Show all the related information when the hot spots are clicked.
2. Create a web page with the following.
 - a. Cascading style sheets.
 - b. Embedded style sheets.
 - c. Inline style sheets. Use our college information for the web pages.
3. Validate the Registration, user login, user profile and payment by credit card pages using JavaScript.
4. Write programs in Java using Servlets:
 - i. To invoke servlets from HTML forms
 - ii. Session tracking using hidden form fields and Session tracking for a hit count
5. Write programs in Java to create three-tier applications using servlets for conducting on- line examination for displaying student mark list. Assume that student information is available in a database which has been stored in a database server.
6. Install TOMCAT web server. Convert the static web pages of programs into dynamic web pages using servlets (or JSP) and cookies. Hint: Users information (user id, password, credit card number) would be stored in web.xml. Each user should have a separate Shopping Cart.
7. Redo the previous task using JSP by converting the static web pages into dynamic web pages. Create a database with user information and books information. The books catalogue should be dynamically loaded from the database.
8. Create and save an XML document at the server, which contains 10 users Information. Write a Program, which takes user Id as an input and returns the User details by taking the user information from the XML document
9. Write programs in PHP for
 - i. Validate the form using PHP regular expression.
 - ii. PHP stores a form data into database.
10. Write a web service for finding what people think by asking 500 people's opinion for any consumer product.

TOTAL: 60PERIODS**OUTCOMES:****Upon Completion of the course, the students will be able to:**

- Construct Web pages using HTML/XML and style sheets.
- Build dynamic web pages with validation using Java Script objects and by applying different event handling mechanisms.
- Develop dynamic web pages using server side scripting.
- Use PHP programming to develop web applications.
- Construct web applications using AJAX and web services.

CS8661 INTERNET PROGRAMMING LABORATORY

CONTENTS

S.No	Name of the Experiment	Page No.
1	IMAGE MAPPING USING HTML	
2	WEB PAGE USING CASCADING STYLE SHEETS (CSS)	
3	VALIDATE THE REGISTRATION FORM USING JAVASCRIPT	
4.1	INVOKING SERVLET FROM HTML FORMS	
4.2	SESSIONS TRACKING USING HIDDEN FORM FIELDS	
4.3	SESSION TRACKING FOR A HIT COUNT	
5	THREE TIER ARCHITECTURE USING SERVLETS	
6	SHOPPING CART USING TOMCAT WEB SERVER	
7	CONVERSION OF STATIC WEB PAGES INTO DYNAMIC WEB PAGES	
8	USER INFORMATION AND INFORMATION RETRIEVAL FROM XML	
9.1	VALIDATE THE FORM USING PHP REGULAR EXPRESSION	
9.2	PHP STORES A FORM DATA INTO DATABASE	
10	OPINIONS FOR CONSUMER PRODUCT	
11	AJAX.	
12	IMPLEMENTING AN APPLICATION USING THE WEB SERVICES	

EX.NO:1

IMAGE MAPPING USING HTML

DATE:

AIM:

Create a web page with the following using HTML

- To embed an image map in a web page
- To fix the hot spots
- Show all the related information when the hot spots are clicked.

PROCEDURE:

1. Insert an image in the web page by using element.
2. Define an image map by using <map> element and use the name attribute to identify the image map.
3. Define the hotspots (certain areas of an image) by using <area> element with the following attributes:
 - href – specifies the target of the link.
 - shape – specifies the shape of the hotspots.
 - coords – specifies the coordinate values of the hotspots
4. Use an image map with an element by using the usemap attribute.
5. Display the web page and click the hotspots to display the related information.

PROGRAM:

india.html

```
<html>
<head>
<title>image map</title>
</head>
<body>
<map id="picture">
<area href="Tamilnadu.html" shape="circle" coords="170,490,30"alt="Tamilnadu"/>
<area href="Andhrapradesh.html" shape="poly"
coords="165,355,200,355,220,380,170,425,165,355"alt="Andhrapradesh"/>
<area href="Kerala.html" shape="poly"
coords="115,455,160,470,140,485,150,505,150,530,135,500,115,455"alt="Kerala"/>
</map>

</body>
</html>
```

Tamilnadu.html:

```
<html>
<head>
<title>About Tamilnadu</title>
</head>
<body>
<center><h1>Tamilnadu</h1></center>
<hr>
<ul>
<li>Area:1,30,058 Sq.kms.</li>
<li>Capital:chennai</li>
<li>Language:Tamil</li>
<li>Population:6,21,10,839</li>
</ul>
<a href="india.html">Home</a>
</body>
</html>
```

Andhrapradesh.Html:

```
<html>
<head>
<title>About Andhrapradesh</title>
</head>
<body>
<center><h1>Andhrapradesh</h1></center>
<hr>
<ul>
<li>Area:2,75,068 Sq.Kms.</li>
<li>Capital:Hyderabad</li>
<li>Language:Telugu</li>
<li>Population:7,57,27,541</li>
</ul>
<a href="india.html">Home</a>
</body>
</html>
```

Kerala.Html:

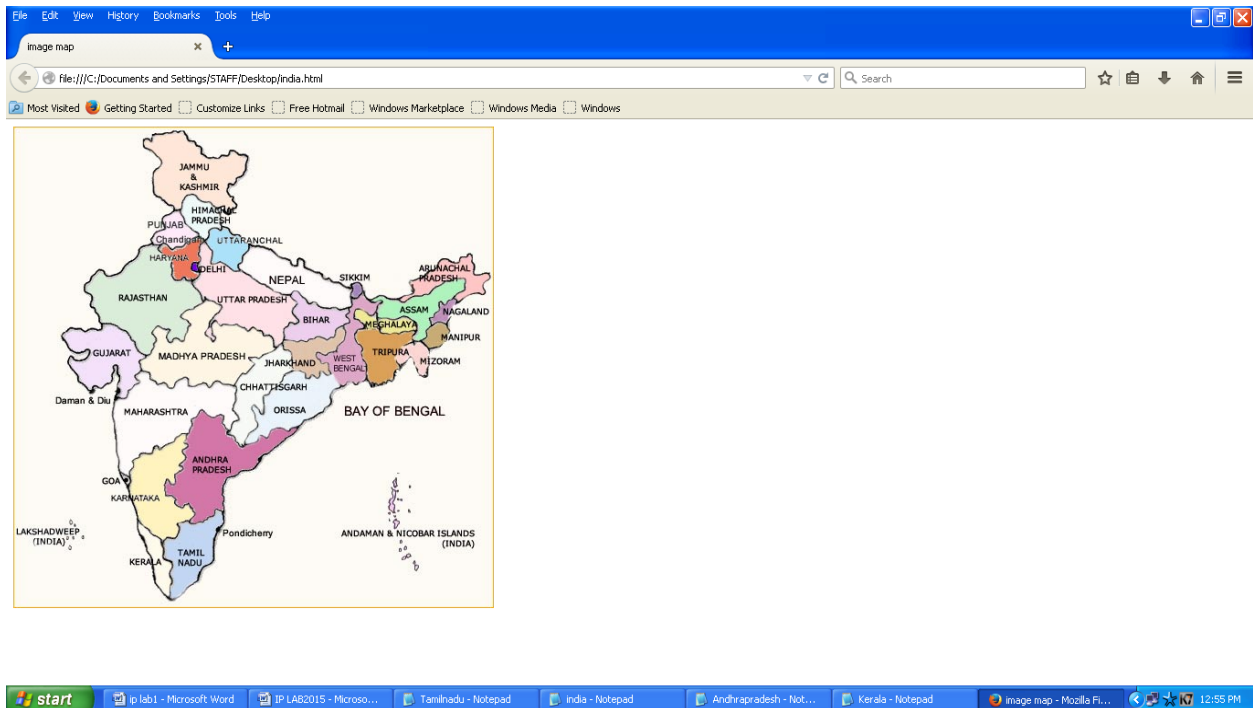
```
<html>
<head>
<title>About Kerla</title>
</head>
<body>
<center><h1>Kerla</h1></center>
```


COURSE: CS8661 INTERNET PROGRAMMING LABORATORY

```
<hr>
<ul>
<li>Area:38,863 Sq.Kms.</li>
<li>Capital:Thiruvananthapuram</li>
<li>Language:Malayalam</li>
<li>Population:3,18,38,619</li>
</ul>
<a href="india.html">Home</a>
</body></html>
```

Output:

India

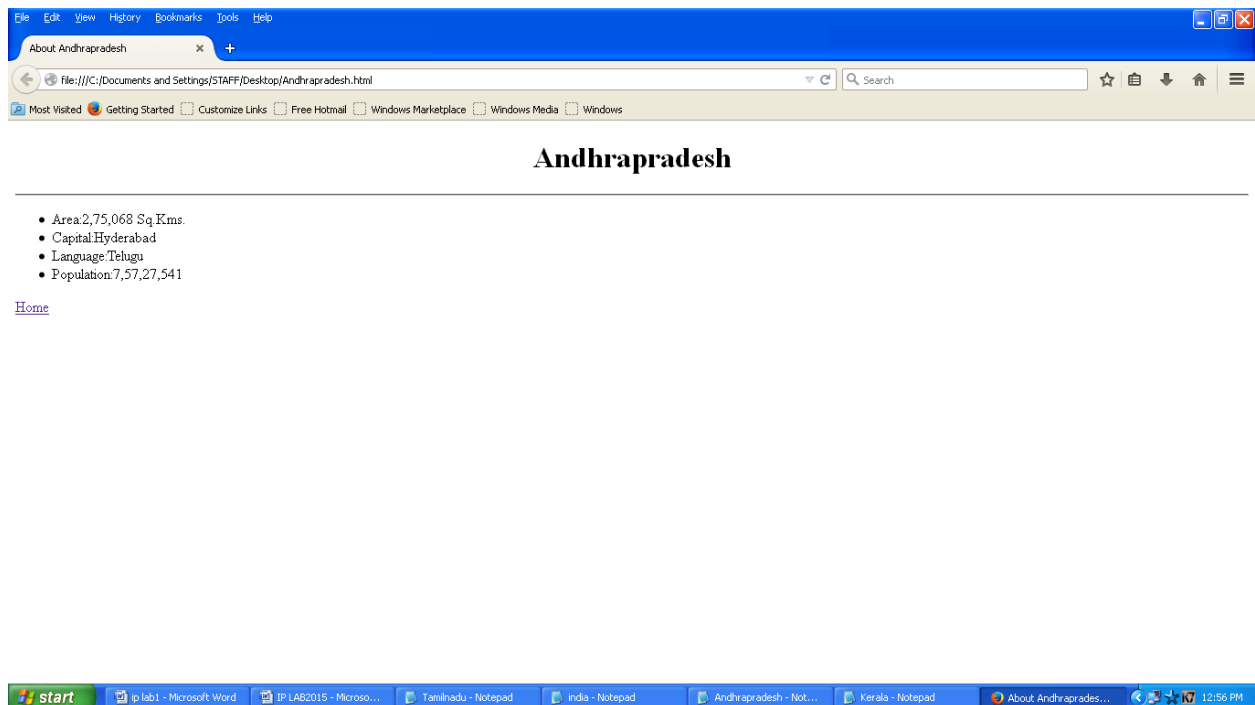


COURSE: CS8661 INTERNET PROGRAMMING LABORATORY

Tamilnadu:

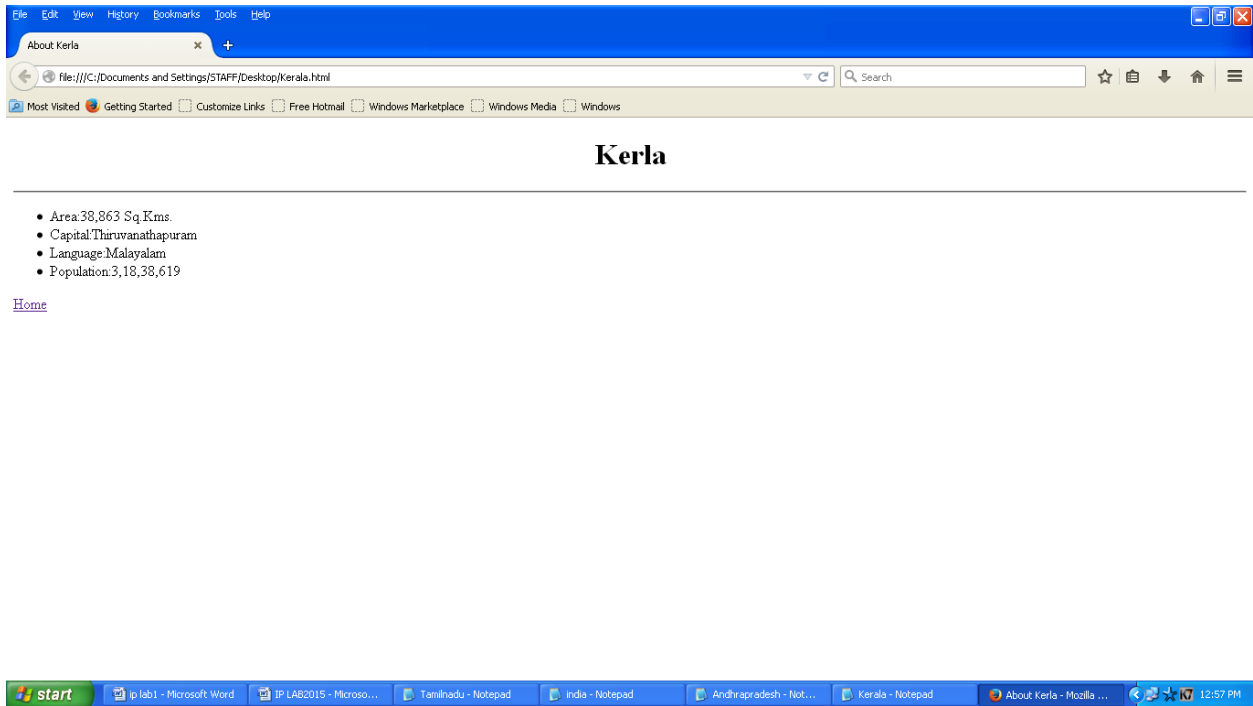


Andhrapradesh:



COURSE: CS8661 INTERNET PROGRAMMING LABORATORY

Kerala:



RESULT:

Thus the HTML program for image hotspots was executed.

Course Teacher: Mr.D.Vaduganathan., Mr.T.Arunkumar., Assistant Professor, Department of CSE,
Excel Engineering College, Komarapalayam-637303

EX.NO: 2 WEB PAGE USING CASCADING STYLE SHEETS (CSS)

DATE:

AIM:

To create a web page with the following Cascading style sheets (CSS)

- Inline style
- Embedded style
- External style

PROCEDURE:

1. Declare an individual element's style by using 'style' attribute (Inline style).
2. Embed the entire style rules in an HTML document's head section (Embedded style) by using <style> element as follows.

```
<style type="text/css">
```

3. Create the separate document that contain only CSS rules (External style) and save it as 'extStyle.css'.
4. Create the link between the external style rules (extStyle.css) and HTML document by using <link> element as follows.

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

PROGRAM:

CSSDemo.html

```
<html>
< head>
<title>Sachin Tendulkar</title>
<style type="text/css">
    #tid1 { background-color:gray;width:800;height:200}
    #tid2 { background-color:gray;width:800;height:300}
    img { width:794;height:200}
</style>
<link rel="stylesheet" type="text/css" href="extStyle.css"/>
</head>
```

```

<body>
<h1 style="text-align:center;font-family:Courier New;color:blue">Sachin Ramesh
Tendulkar</h1>
<hr/>
<table id="tid1" align="Center">
<tr>
<td></td>
</tr>
</table>
<table id="tid2" align="Center">
<tr>
<td class="cls1" valign="top">
<h3 style="font-family:Lucida Sans Unicode;color:blue">Profile</h3>
<p>
Sachin Ramesh Tendulkar was born on 24 April 1973 in Mumbai. Sachin Tendulkar is an Indian
cricketer. He has been the most complete batsman of his time. He holds several batting records,
including the most Test centuries and the most one-day international centuries, and was rated in
2002 by Wisden as the second greatest Test batsman ever, after Sir Don Bradman. He received
the Rajiv Gandhi Khel Ratna, India's highest sporting honour, for 1997-1998, and the civilian
award Padma Shri in 1999. Tendulkar was a Wisden Cricketer of the Year in 1997.
</p>
</td>
<td class="cls1" valign="top">
<h3 style="font-family:Lucida Sans Unicode;color:blue">Highlights of Tendulkar's
Career</h3>
<div style="font-size:10pt;color:gold;font-family:Lucida Sans Unicode;text-align:justify">
<ul type="square">
<li>Highest number of Test centuries (51)</li>
<li>Highest number of ODI centuries (49)</li>
<li>First cricketer to cross 10,000 runs in both ODIs and Test</li>
<li>Highest batting average among batsmen with over 10,000 ODI runs</li>
<li>In 1998 he hit 9 ODI centuries, the highest by any player in an year</li>
<li>673 runs in 2003 World Cup, highest by any one in a single Cricket World Cup</li>
<li>First cricketer to reach double century in ODI</li>
<li>Played in the highest number of Cricket Grounds</li>
<li>Rated as the second best batsman of all time (next to Don Bradman) by Wisden.</li>
</ul>
</div>
</td>
</tr>
</table>
</body>

```

COURSE: CS8661 INTERNET PROGRAMMING LABORATORY

</html>

extStyle.css

```
body{background-color:black}
```

```
.cls1{width:325;height:300}
```

```
p{font-family:Lucida Sans Unicode;color:silver;font-size:12pt;text-align:justify}
```

Output:



RESULT:

Thus the program to create webpage using CSS was executed.

EX.NO: 3 VALIDATE THE REGISTRATION FORM USING JAVASCRIPT

DATE:

AIM:

To write a java script program Validate the Registration, user login, user profile and payment by credit card pages.

PROCEDURE:

1. Create the html <form> with user's fields
2. User fields has to create by following syntax

`<input type="text" name="" id="">`

3. Validate the fields using java scripts
4. Include the Validate fields using <scripts> </scripts> tag

PROGRAM:

Home page:

Main.html:

```
<html>
<frameset rows="25%,*">
<frame src="top.html" name="top" scrolling="no" frameborder="0">
<frameset cols="25%,75%">
<frame src="left.html" name="left" scrolling="no" frameborder="0">
<frame src="right.html" name="right" scrolling="auto" frameborder="0">
</frameset>
</frameset>
</html>
```

Top.html:

```
<html>
<body bgcolor="pink">
<br><br>
<marquee><h1 align="center"><b><u>ONLINE BOOK
STORAGE</u></b></h1></marquee>
</body>
</html>
```

Right.html:

```
<html>
```

```
<body>
<br><br><br><br><br>
<h2 align="center">
<b><p> welcome to online book storage. Press login if you are
having id otherwise press registration.
</p></b></h2>
</body> </html>
```

Left.html:

```
<html>
<body bgcolor="pink">
<h3>
<ul>
<li><a href="login.html" target="right"><font color="black">
LOGIN</font></a></li><br><br>
<li><a href="profile.html" target="right"><font color="black">
USER PROFILE</font></a></li><br><br>
<li><a href="catalog.html" target="right"><font color="black">
BOOKS CATALOG</font></a></li><br><br>
<li><a href="scart.html" target="right"><font color="black">
SHOPPINGCART</font></a></li><br><br>
<li><a href="payment.html" target="right"><font color="black">
PAYMENT</font></a></li><br><br>
<br><br>
</ul>
</body>
</html>
```

Registration and user Login

Login.html:

```
<html>
<body bgcolor="pink"><br><br><br>
<script language="javascript">
function validate()
{
var flag=1;
if(document.myform.id.value=="")||
document.myform.pwd.value=="")
{
alert("LoginId and Password must be filled")
flag=0;
}
if(flag==1)
{
```


[illegible]

User profile page

Profile.html:

```
<html>
<body bgcolor="pink"><br><br>
<script type="text/javascript">
function validate()
{
var flag=1;
if(document.myform.name.value=="")
document.myform.addr.value=="||
document.myform.phno.value=="||
document.myform.id.value=="||
document.myform.pwd.value=="")
{
alert("Enter all the details");
flag=0;
}
var str=document.myform.phno.value;
var x=new RegExp("\\d","g");
if(!(str.match(x)))
{
```

COURSE: CS8661 INTERNET PROGRAMMING LABORATORY

[illegible]

</form></body></html>

Books catalog:

Scart.html:

```
<html>
<body bgcolor="pink"><br><br><br>
<script language="javascript">
function validate()
{
var flag=1;
if(document.myform.title.value=="")
{
flag=0;
}
str=document.myform.title.value;
if(str=="c"||str=="C")
{
document.myform.t1.value="C";
document.myform.t2.value=444;
}
else if(str=="jsp"||str=="JSP")
{
document.myform.t1.value="JSP";
document.myform.t2.value=555;
}
else
{
flag=0;
}
if(flag==1)
{
alert("VALID INPUT");
}
else
{
alert("INVALID INPUT");
document.myform.focus();
}
}
</script>
<form name="myform" action="payment.html" target="right">
<div align="center"><pre>
BOOK TITLE :<input type="text" name="title"><br>
```

[illegible]

Catalog.html:

Course Teacher: Mr.D.Vaduganathan., Mr.T.Arunkumar., Assistant Professor, Department of CSE,
Excel Engineering College, Komarapalayam-637303

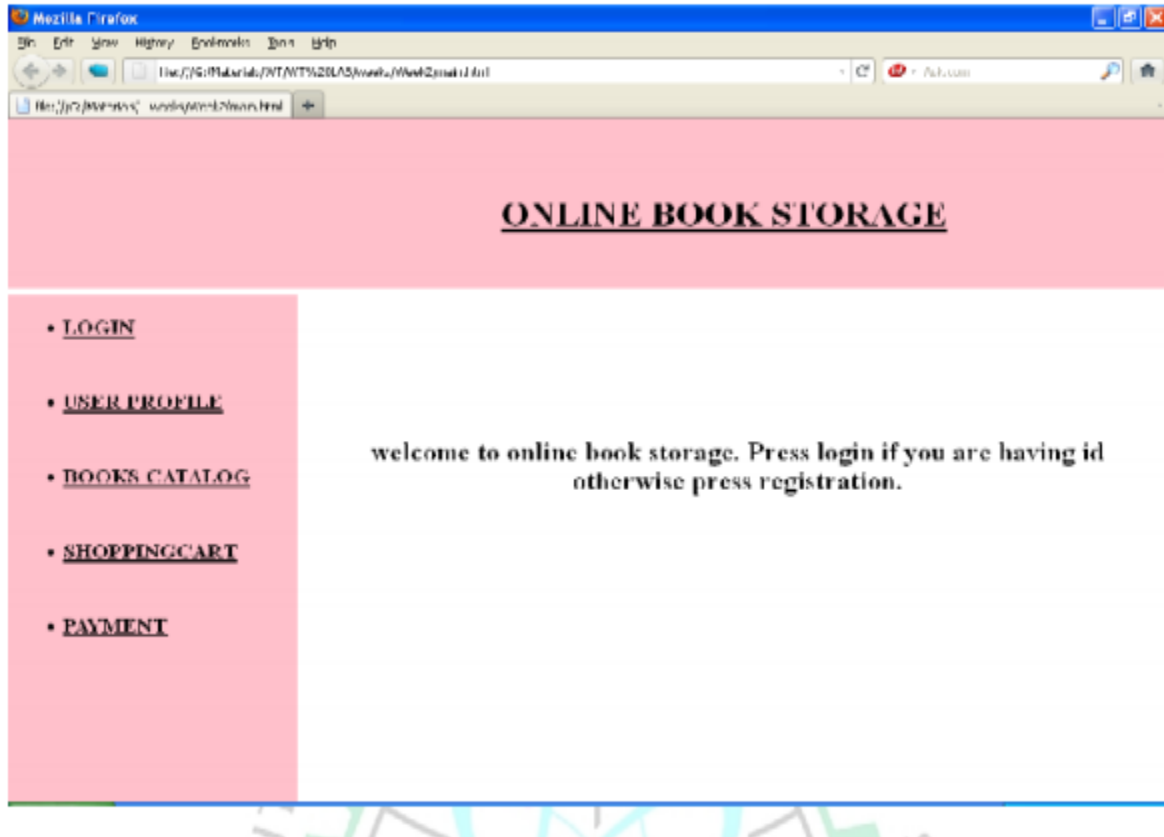
Payment.html:

Course Teacher: Mr.D.Vaduganathan., Mr.T.Arunkumar., Assistant Professor, Department of CSE,
Excel Engineering College, Komarapalayam-637303

Order.html:

OUTPUT:

Main.html



Login.html:

ONLINE BOOK STORAGE

- [LOGIN](#)
- [USER PROFILE](#)
- [BOOKS CATALOG](#)
- [SHOPPINGCART](#)
- [PAYMENT](#)

LOGIN ID: phani
PASSWORD: *****

ok cancel

Catalog.html:

ONLINE BOOK STORAGE

- [LOGIN](#)
- [USER PROFILE](#)
- [BOOKS CATALOG](#)
- [SHOPPINGCART](#)
- [PAYMENT](#)

LOGIN ID:
TITLE:
NO. OF BOOKS:
COPY OF BOOK:

ok clear

Scart.html:

ONLINE BOOK STORAGE

- LOGIN
- USER PROFILE
- BOOKS CATALOG
- SHOPPINGCART
- PAYMENT

BOOK TITLE :

Book Title: Book Cost:

Payment.html:

ONLINE BOOK STORAGE

- LOGIN
- USER PROFILE
- BOOKS CATALOG
- SHOPPINGCART
- PAYMENT

LOGIN ID:

PASSWORD:

AMOUNT:

CREDITCARDNUMBER:

Order.html



RESULT:

Thus the java script program for Validating the Registration, user login, user profile and payment by credit card pages was executed and verified successfully.

EX.NO:4.1

INVOKING SERVLET FROM HTML FORMS

DATE:

AIM:

To write a html program for invoking servlet using html.

PROCEDURE:

- Step 1: In html program, define the html, head and title tag.
- Step 2: Then the title is Student Information Form and closes the title and head tag.
- Step 3: Define the body tag inside the body tag create form and table simultaneously.
- Step 4: The table consists of following information Roll no, Student name, Address, Phone no and total marks.
- Step 5: In the servlet program, import the summary package and create a own servlet class extends with generic servlet.
- Step 6: In the service method defined to request and response.
- Step 7: Create the object and for print writer and get writer() value.
- Step 8: The enumeration object get the servlet request parameter.
- Step 9: Create objects for string method and it is displayed another object value received get parameter of name received and displayed the value received value.

EXECUTION STEPS:

- Step 1: Install the Tomcat Server
- Step 2: Setting up Java Development Kit
 - C:\> set PATH=C:\jdk1.7.0\bin;%PATH%
 - C:\> set JAVA_HOME=C:\jdk1.7.0
- Step 3: Setting up Web-Server %CATALINA_HOME%\bin\startup.bat or
 - C:\apache-tomcat-7\bin\startup.bat
- Step 4: After startup, the default web applications included with Tomcat will be available by Visiting <http://localhost:8080/>
- Step 5: Setting up class path
 - C:\> set CATALINA=C:\apache-tomcat-7
 - C:\> set CLASSPATH=%CATALINA%\common\lib\servlet-api.jar;%CLASSPATH%.
- Step 6: By default, a servlet application is located at the path<Tomcat-installation directory>/webapps/ROOT and the class file would reside in <Tomcat-installation-directory>/webapps/ROOT /WEB-INF/classes.
- Step 7: web.xml file located in <Tomcat-installationdirectory>/webapps/ROOT /WEB-INF/.
- Step 8: Open any browser and type <http://localhost:8080/myservletdemo>

PROGRAM:

Myservletdemo.java

```
import java.io.*;
import java.util.*;
import javax.servlet.*;
public class Myservletdemo extends GenericServlet
{
    public void service(ServletRequest req,ServletResponse res)
    throws ServletException,IOException
    {
        PrintWriter out=res.getWriter();
        Enumeration en=req.getParameterNames();
        while(en.hasMoreElements())
        {
            String name_received=(String)en.nextElement();
            out.print(name_received+"=");
            String value_received=req.getParameter(name_received);
            out.println(value_received);
            out.println(" ");
        }
        out.close();
    }
}
```

Myservletdemo.html

```
<html>
<head>
<title>Student Information Form</title>
</head>
<body>
<form name="form1" action="http://localhost:8080/servlet/myservletdemo">
<h1>STUDENT INFORMATION FORM</H1>
<h3>Enter student information in following fields-</h3>
<table>
<tr>
<td>Roll Number</td>
<td><input type="text" name="Roll Number" size="25" value=" "></td>
</tr> <tr>
<td>Student Name</td>
<td><input type="text" name="Student Name" size="25" value=" "></td>
</tr>
<tr>
<td>Student Address</td>
```

COURSE: CS8661 INTERNET PROGRAMMING LABORATORY

```
<td><input type="text" name="Student Address" size="50" value=" "></td>
</tr>
<tr>
<td>Phone</td>
<td><input type="text" name="Phone" size="10" value=" "></td>
</tr>
<tr>
<td>Total Marks</td>
<td><input type="text" name="Total Marks" size="25" value=" "></td>
</tr>
</table>
<input type="submit" value="submit">
</form>
</center>
</body>
</html>
```

WEB.XML

```
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
version="3.0"
metadata-complete="true">
<display-name>Welcome to Tomcat</display-name>
<description>
Welcome to Tomcat
</description>
<servlet>
<servlet-name>myservletdemo</servlet-name>
<servlet-class>myservletdemo</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>myservletdemo</servlet-name>
<url-pattern>/myservletdemo</url-pattern>
</servlet-mapping>
</web-app>
```

COURSE: CS8661 INTERNET PROGRAMMING LABORATORY

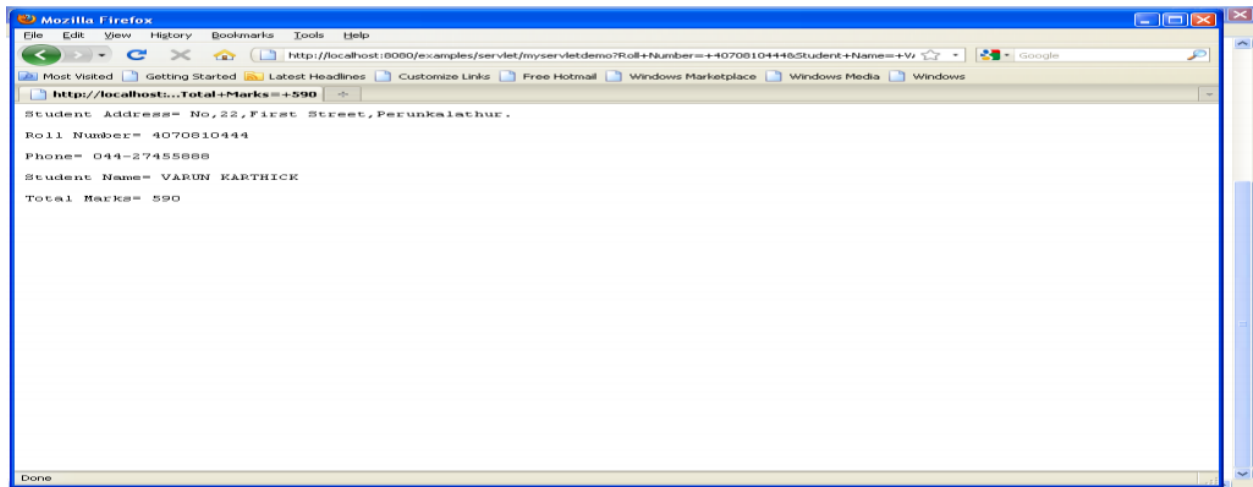
SAMPLE OUTPUT:



STUDENT INFORMATION FORM

Enter student information in following fields-

Roll Number	4070810444
Student Name	VARUN KARTHICK
Student Address	No.22,First Street,Perunkalathur.
Phone	044-27455888
Total Marks	590



http://localhost:8080/examples/servlet/myServletdemo?Roll+Number=+4070810444&Student+Name=+Vr

http://localhost:8080/examples/servlet/myServletdemo?Roll+Number=+4070810444&Student+Name=+Vr

Student Address= No,22,First Street,Perunkalathur.

Roll Number= 4070810444

Phone= 044-27455888

Student Name= VARUN KARTHICK

Total Marks= 590

RESULT:

Thus the program for invoking servlet using html was created successfully and output is verified.

Course Teacher: Mr.D.Vaduganathan., Mr.T.Arunkumar., Assistant Professor, Department of CSE,
Excel Engineering College, Komarapalayam-637303

EX.NO: 4.2

SESSIONS TRACKING USING HIDDEN FORM FIELDS

DATE:

AIM:

To write a java servlet program for session tracking using hidden form fields.

PROCEDURE:

Step 1: Import all the necessary packages.

Step 2: Declare a class FirstServlet1 that extends HttpServlet.

Step 3: In hidden form field a hidden (invisible) textfield is used for maintaining the state of an user. <input type='hidden' name='uname' value="" ">)

Step 4: We are storing the name of the user in a hidden textfield and getting that value from another servlet.

Step 5: After entering the name from the HTML page, it goes to the first servlet. First Servlet gets the name and sends as an invisible text field to the servlet2.

Step 6: Second servlet accepts the hidden text field and constructs the invisible text.

PROGRAM:

FirstServlet1.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class FirstServlet1 extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response){
        try{
            response.setContentType("text/html");
            PrintWriter out = response.getWriter();
            String n=request.getParameter("userName");
            out.print("Welcome "+n);
            out.print("<form action='http://localhost:8080/secserv/SecondServlet1' method ='get'>");
            out.print("<input type='hidden' name='uname' value='"+n+"'>");
            out.print("<input type='submit' value='go'>");
            out.print("</form>");
            out.close();
        }catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```

```
}
```

Web.xml:

```
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
version="3.0"
metadata-complete="true">
<display-name>Welcome to Tomcat</display-name>
<description>
Welcome to Tomcat
</description>
<servlet>
<servlet-name>FirstServlet1</servlet-name>
<servlet-class>FirstServlet1</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>FirstServlet1</servlet-name>
<url-pattern>/FirstServlet1</url-pattern>
</servlet-mapping>
</web-app>
```

SecondServlet1.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class SecondServlet1 extends HttpServlet {
public void doGet(HttpServletRequest request, HttpServletResponse response)
{
try{
response.setContentType("text/html");
PrintWriter out = response.getWriter();
String n=request.getParameter("uname");
out.print("Hello "+n);
out.close();
}catch(Exception e){System.out.println(e);}
}
}
```

Web.xml:

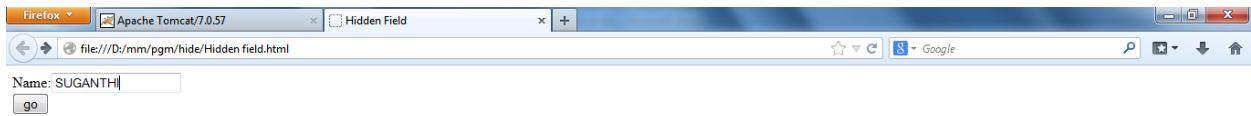
```
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
```


COURSE: CS8661 INTERNET PROGRAMMING LABORATORY

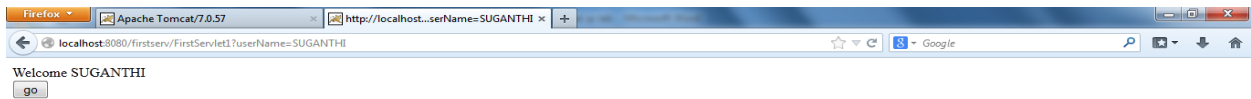
```
version="3.0"
metadata-complete="true">
<display-name>Welcome to Tomcat</display-name>
<description>
Welcome to Tomcat
</description>
<servlet>
<servlet-name>SecondServlet1</servlet-name>
<servlet-class>SecondServlet1</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>SecondServlet1</servlet-name>
<url-pattern>/SecondServlet1</url-pattern>
</servlet-mapping>
</web-app>
Hidden Field.html
<html>
<head><title>Hidden Field</title></head>
<body>
<form action="http://localhost:8080/firstserv/FirstServlet1" method="get">
Name:<input type="text" name="userName"/><br/>
<input type="submit" value="go"/>
</form>
</body>
</html>
```

SAMPLE OUTPUT:

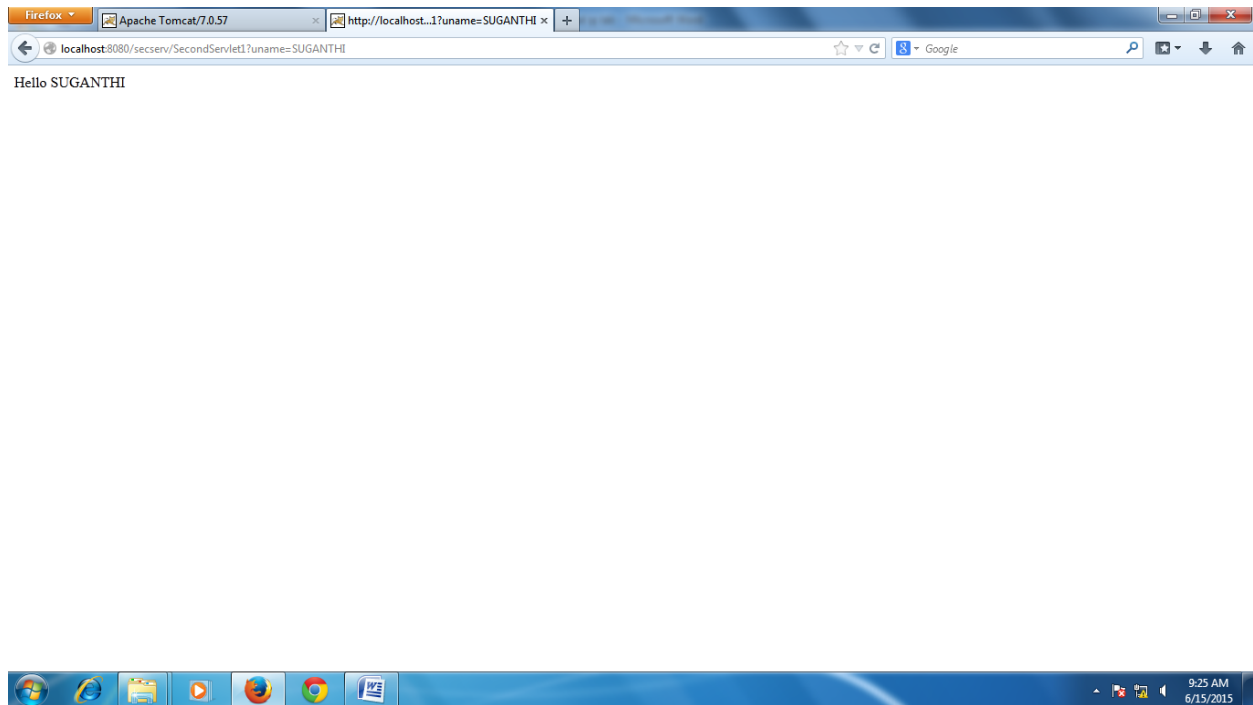
Hidden Field.html



First Servlet:



Second Servlet:



RESULT:

Thus the java servlet program for session tracking using hidden form fields was executed successfully and output is verified.

EX.NO: 4.3 SESSION TRACKING FOR A HIT COUNT

DATE:

AIM:

To write a java servlet program to track the session for a hit count.

PROCEDURE:

- Step 1: Import all the necessary packages.
- Step 2: Declare a class PAGEHIT Counter that extends HttpServlet
- Step 3: Use init parameter and reset hit count as 0 hitcount=0;
- Step 4: Set response content type and this method excutes whenever the servlet is hit, by Incrementing a hitcount.
- Step 5: Stop the program.

PROGRAM:

PageHitCounter.java

```
import java.io.*;
import java.sql.Date;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

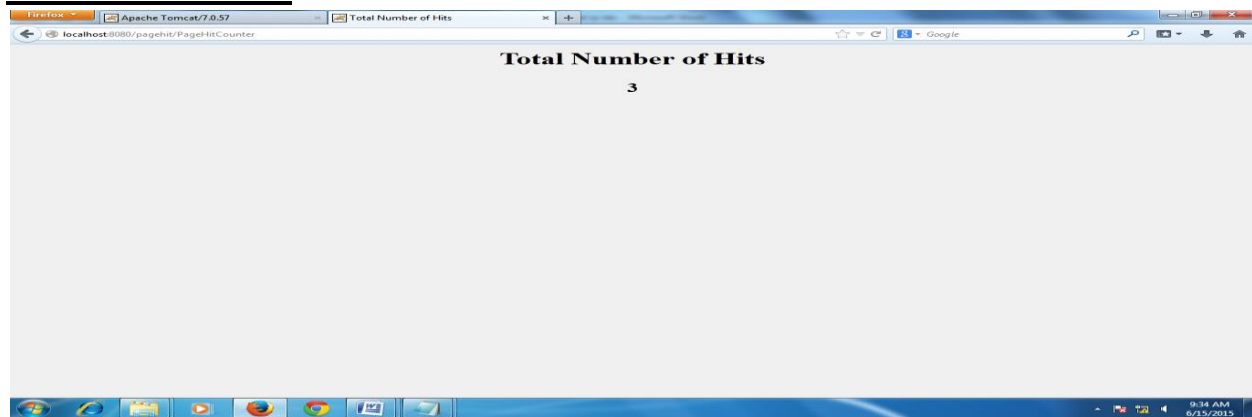
public class PageHitCounter extends HttpServlet{
    private int hitCount;
    public void init()
    {
        // Reset hit counter.
        hitCount = 0;
    }
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException
    {
        // Set response content type
        response.setContentType("text/html");
        // This method executes whenever the servlet is hit
        // increment hitCount
        hitCount++;
        PrintWriter out = response.getWriter();
        String title = "Total Number of Hits";
        String docType =
            "<!doctype html public \"-//w3c//dtd html 4.0 \" +
            \"transitional//en\">\n";
        out.println(docType + "<html>\n" + "<head><title>" + title + "</title></head>\n" +
            "<body bgcolor=\"#f0f0f0\">\n" + "<h1 align=\"center\">" + title + "</h1>\n" +
            "<h2 align=\"center\">" + hitCount + "</h2>\n" +
            "</body></html>");
    }
    public void destroy()
    {
    }
}
```

Web.xml:

```
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
    version="3.0"
    metadata-complete="true">
```

```
<display-name>Welcome to Tomcat</display-name>
<description>
Welcome to Tomcat
</description>
<servlet>
<servlet-name>PageHitCounter</servlet-name>
<servlet-class>PageHitCounter</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>PageHitCounter</servlet-name>
<url-pattern>/PageHitCounter</url-pattern>
</servlet-mapping>
</web-app>
```

SAMPLE OUTPUT:



RESULT:

Thus the java servlet program for session tracking session tracking for a hit count was executed successfully and output is verified.

EX.NO:5 THREE TIER ARCHITECTURE USING SERVLETS

DATE:

AIM:

To write a java servlet program to conduct online examination and to display student mark list available in a database which has been stored in a database server.

PROCEDURE:

Client:

Step 1: In online.html on the client side declare the questions for online exam in which True/False options are

displayed that you like to transfer to the server using html form and input type tags.

Step 2: create a submit button and close all the included tags.

Servlet:

Step 1: Import all necessary packages

Step 2: Define a class that extends servlet

Step 3: In the do Post() method, do the following:

- i) Set the content type of the response to "text/html" by using setContenttype
- ii) Create a writer to the response
- iii) Get a parameter from the request
- iv) If its value is equal to right answer then add 5 to mark variable
- v) Similarly repeat step for all parameters
- vi) Display the result in an html format using response.getWriter().

PROGRAM:

online.html

```
<html>
<head>
<title> Database test</title>
</head>
<body>
<center>
<form action="http://localhost:8080/exam/exam" method=POST>
<div align="left"><br>
<b>seat number:</b> <input type="text" name="Seat_no">
<div align="right"><br>
<b>Name:</b> <input type="text" name="Name" size="50"><br>
</div>
<br>
<br>
<b>1.Is JAVA a platform independency</b><br>
<input type="radio" name="group1" value="True">True
<input type="radio" name="group1" value="False">False<br>
<b>2.ASP .NET is a client side programming</b><br>
<input type="radio" name="group2" value="True">True
<input type="radio" name="group2" value="False">False<br>
<b>3.MATHEMATICS is the backbone of engineering</b><br>
<input type="radio" name="group3" value="True">True
<input type="radio" name="group3" value="False">False<br>
<b>4.You are working in IBM machines</b><br>
<input type="radio" name="group4" value="True">True
<input type="radio" name="group4" value="False">False<br>
<b>5.C++ is a fully OOP's language</b><br>
<input type="radio" name="group5" value="True">True
<input type="radio" name="group5" value="False">False<br>
<br><br><br>
<center>
<input type="submit" value="submit"><br><br>
</center>
</form>
</body>
</html>
```

Exam.java:

```
import java.sql.*;
```

COURSE: CS8661 INTERNET PROGRAMMING LABORATORY

```
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class exam extends HttpServlet
{
String message,Seat_no,Name,ans1,ans2,ans3,ans4,ans5;
int Total=0;
Connection connect;
Statement stmt =null;
ResultSet rs=null;
public void doPost(HttpServletRequest request,HttpServletResponse response)
throws ServletException,IOException
{
try
{
String url="jdbc:odbc:StudentDB2";
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
connect=DriverManager.getConnection(url," "," ");
message="Connection Successful";
}
catch(ClassNotFoundException cnfex)
{
cnfex.printStackTrace();
}
catch(SQLException sqllex)
{
sqllex.printStackTrace();
}
catch(Exception excp)
{
excp.printStackTrace();
}
Seat_no=request.getParameter("Seat_no");
Name=request.getParameter("Name");
ans1=request.getParameter("group1");
ans2=request.getParameter("group2");
ans3=request.getParameter("group3");
ans4=request.getParameter("group4");
ans5=request.getParameter("group5");
if(ans1.equals("True"))
Total+=2;
if(ans2.equals("False"))
Total+=2;
if(ans3.equals("True"))
Total+=2;
if(ans4.equals("True"))
Total+=2;
```

```

if(ans5.equals("False"))
Total+=2;
try
{
Statement stmt=connect.createStatement();
String query="INSERT into StudentTable VALUES (" +Seat_no+"","+Name+"","+Total+)";
//*****"+Seat_no,Name,Marks+"")+" VALUES("
int result=stmt.executeUpdate(query);
stmt.close();
}
catch(SQLException ex)
{
}
response.setContentType("text/html");
PrintWriter out=response.getWriter();
out.println("<html>");
out.println("<body bgcolor=yellow>");
out.println("<h1>"+message+"</h1>\n");
out.println("<h3>DataBase Updated");
out.println("<br><br>");
out.println("<b>"+ "The Student Database is as follows");
out.println("<table border=5>");
try
{
Statement stmt=connect.createStatement();
String query="SELECT * FROM StudentTable";
rs=stmt.executeQuery(query);
out.println("<th>"+ "Seat_no"+ "</th>");
out.println("<th>"+ "Name"+ "</th>");
out.println("<th>"+ "Marks"+ "</th>");
while(rs.next())
{
out.println("<tr>");
out.println("<td>"+rs.getInt(1)+"</td>");
out.println("<td>"+rs.getString(2)+"</td>");
out.println("<td>"+rs.getInt(3)+"</td>");
out.println("</tr>");
}
out.println("</table>");
}
catch(SQLException ex)
{}
finally
{
try
{
if(rs!=null)
rs.close();

```



```
if(stmt!=null)
stmt.close();
if(connect!=null)
connect.close();
}

catch(SQLException e)
{}}
out.println("<center>");
out.println("<h1>Thanks!</h1>\n");
out.println("</center>");
out.println("</body></html>");
}}
```

Web.xml:

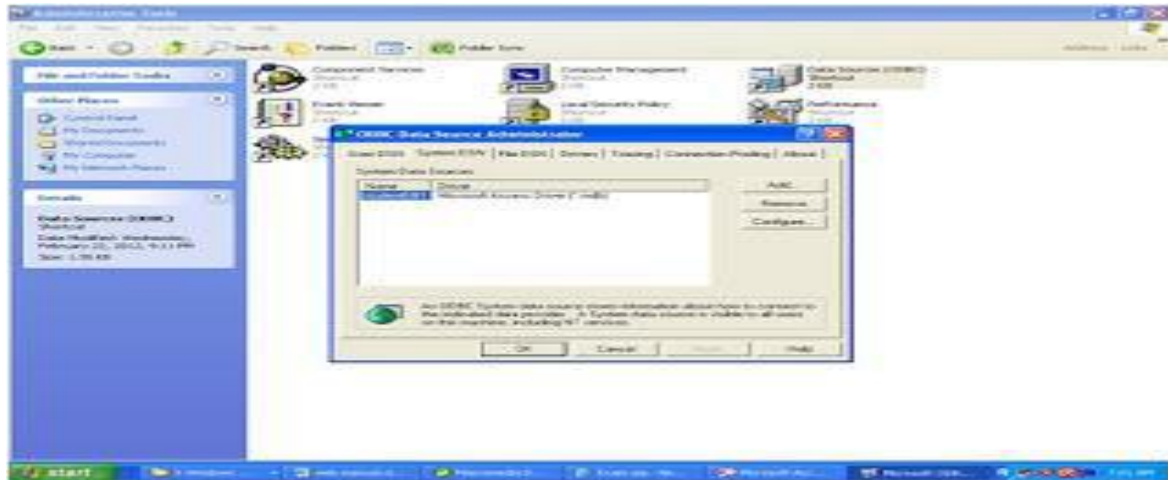
```
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
version="3.0" metadata-complete="true">
<display-name>Welcome to Tomcat</display-name>
<description>
Welcome to Tomcat
</description>
<servlet>
<servlet-name>exam</servlet-name>
<servlet-class>exam</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>exam</servlet-name>
<url-pattern>/exam</url-pattern>
</servlet-mapping>
</web-app>
```

Execution steps:

1. Create the database StudentDB2.mdb in which Studenttable is created. Note that we have to create an empty database by specifying simply the field names such as Seat_no, Name and Marks.
2. Create the DSN for Student database:

Open the Control panel and double click on the Administrative Tools icon. Then double click on the Datasources(ODBC) icon. The window appears as:

Database(Database1.mdb) –DSN CONNECTION(studentDB2)



3. Click on User DSN tab and select Microsoft Access Driver (*.mdb) and then click on the finish button. Then type the Data Source Name as StudentDB2 and click on select button to select the database file for corresponding DSN. Click on OK button. Now connection to the database is done using JDBC-ODBC driver.
4. Create a html program as online.html in which True/False questions are displayed.
5. Then Create a java program as Exam.java which is a servlet code for computing the total score of each student and it displays the database contents.
6. Compile the java program by using javac exam.java. then class file will be created and place the corresponding class file to the location c:\ "Program Files\apache-tomcat-7.0.29\webapps\exam\WEB-INF\class.
7. Then edit the web.xml file in WEB-INF for servlet name and servlet class as exam
8. Now open online.html file in a browser window the web page opens and if we click on submit button the student information should get updated in the database by invoking a servlet.

COURSE: CS8661 INTERNET PROGRAMMING LABORATORY

SAMPLE OUTPUT:

Database test

file:///C:/Program Files/apache-tomcat-7.0.29/webapps/exam/online.html

Search

seat number:

Name:

1.Is JAVA a platform independency
☐ True ☐ False

2.ASP.NET is a client side programming
☐ True ☐ False

3.MATHEMATICS is the backbone of engineering
☐ True ☐ False

4.You are working in IBM machines
☐ True ☐ False

5.C++ is a fully OOP's language
☐ True ☐ False

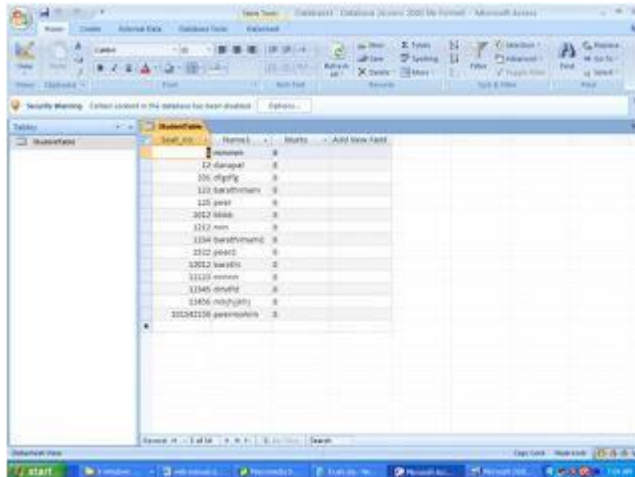
9:57 AM
12/29/2014

COURSE: CS8661 INTERNET PROGRAMMING LABORATORY



Students Marksheet

Seat_no	Name	Marks
101	Ravi	75



RESULT:

Thus the java servlet program to conduct online examination and to display student mark list available in a database was executed successfully and output is verified.

EX.NO: 6

SHOPPING CART

AIM:

Install TOMCAT web server. Convert the static WebPages into dynamic WebPages using servlets and cookies. Hint: Users information (user id, password, credit card number) would be stored in web.xml. Each user should have a separate Shopping Cart.

PROCEDURE:

- First install the tomcat into the system.
- Then make a sub directly (eg., tr) in the \tomcat\webapps.
- Under tr create WEB-INF directory and also place the html files in this tr directory only.
- Next under WEB-INF create two subclasses lib, classes and web.xml
- Next place all the class files under the classes and jar files(servlet-api.jar,classes12.jar etc...) under lib subdirectories.
- After this start tomcat by giving the following command at the instll_dir>tomcat>bin Catalina.bat run
- At the I.E(web browser) give the url as http://localhost:8080//tr/htmlfile or servlet url pattern Port no 8080 is assigned for the tomcat.

PROGRAM:

Web.xml

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

```
<!DOCTYPE web-app
PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
  <display-name>Servlet 2.4 Examples</display-name>
  <description>
    Servlet 2.4 Examples.
  </description>
  <servlet>
    <servlet-name>reg</servlet-name>
    <servlet-class>reg</servlet-class>
  </servlet>
  <servlet>
    <servlet-name>login</servlet-name>
    <servlet-class>login</servlet-class>
  </servlet>
  <servlet>
    <servlet-name>profile</servlet-name>
    <servlet-class>profile</servlet-class>
  </servlet>
  <servlet>
    <servlet-name>catalog</servlet-name>
    <servlet-class>catalog</servlet-class>
    <servlet-mapping>
      <servlet-name>order</servlet-name>
      <url-pattern>/order</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
      <servlet-name>catalog</servlet-name>
      <url-pattern>/catalog</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
      <servlet-name>profile</servlet-name>
      <url-pattern>/profile</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
      <servlet-name>login</servlet-name>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<body bgcolor="pink">
<br /><br /><br /><br /><br />
<h1 align="center"><U>ONLINE BOOK STORAGE</U></h1><br /><br /><br />
<h2 align="center"><pre>
<b>Welcome to online book storage.
Press LOGIN if you are having id
otherwise press REGISTRATION
</b></pre></h2>
<br /><br /><pre>
<div align="center"><a href="/tr/login.html">LOGIN</a> <a href="/tr/reg.html">
REGISTRATION</a></div></pre>
</body>
</html>
```

[illegible]

COURSE: CS8661 INTERNET PROGRAMMING LABORATORY

[illegible]

Order.html

[illegible]

Login.java

```
import java.sql.*;
import java.io.*;
import java.util.*;
```

COURSE: CS8661 INTERNET PROGRAMMING LABORATORY

```
import javax.servlet.*;
import javax.servlet.http.*;
public class login extends HttpServlet
{
    public void service(HttpServletRequest req,HttpServletResponse resp)
    throws ServletException,IOException
    {
        PrintWriter pw=resp.getWriter();
        pw.println("<html><body bgcolor=\"pink\"");
        String id=req.getParameter("id");
        String pwd=req.getParameter("pwd");
        try
        {
            Driver d=new oracle.jdbc.driver.OracleDriver();
            DriverManager.registerDriver(d);
            Connection
            con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger")
            ;
            Statement stmt=con.createStatement();
            String sqlstmt="select id,password from login";
            ResultSet rs=stmt.executeQuery(sqlstmt);
            int flag=0;
            while(rs.next())
            {
                if(id.equal(rs.getString(1))&&pwd.equals(rs.getString(2)))
                {
                    flag=1;
                }
            }
            if(flag==0)
            {
                pw.println("SORRY INVALID ID TRY AGAIN ID<br><br>");
                pw.println("<a href=\"/tr/login.html\">press LOGIN to RETRY</a>");
            }
            else
            {
                pw.println("VALID LOGIN ID<br><br>");
                pw.println("<h3><ul>");
                pw.println("<li><ahref=\"profile.html\"><fontcolor=\"black\">USER
                PROFILE</font> </a></li><br><br>");
                pw.println("<li><ahref=\"catalog.html\"><fontcolor=\"black\">BOOKS
                CATALOG</font></a></li><br><br>");
                pw.println("<li><ahref=\"order.html\"><fontcolor=\"black\">ORDER
                CONFIRMATION</font> </a></li><br><br>");
            }
        }
    }
}
```

```
}  
pw.println("</body></html>");  
}  
catch(Exception e)  
{ resp.sendError(500,e.toString());  
}  
}
```

Reg.html

```
import java.sql.*;  
import java.io.*;  
import java.util.*;  
import javax.servlet.*;  
import javax.servlet.http.*;  
public class login extends HttpServlet  
{  
    public void service(HttpServletRequest req,HttpServletResponse resp)  
        throws ServletException,IOException  
    {  
        PrintWriter pw=resp.getWriter();  
        pw.println("<html><body bgcolor=\"pink\"");  
        String name=req.getParameter("name");  
        String addr=req.getParameter("addr");  
        String phno=req.getParameter("phno");  
        String id=req.getParameter("id");  
        String pwd=req.getParameter("pwd");  
        int no=Integer.parseInt(phno);  
        try  
        {  
            Driver d=new oracle.jdbc.driver.OracleDriver();  
            DriverManager.registerDriver(d);  
            Connection  
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger");  
            Statement stmt=con.createStatement();  
            String sqlstmt="select id,password from login";  
            ResultSet rs=stmt.executeQuery(sqlstmt);  
            int flag=0;  
            while(rs.next())  
            {  
                if(id.equals(rs.getString(1))&&pwd.equals(rs.getString(2)))  
                {  
                    flag=1;  
                } }  
        } }
```

```

if(flag==1)
{
pw.println("SORRY INVALID ID ALREADY EXISTS TRY AGAIN WITH NEW
ID<br><br>");
pw.println("<a href='/tr/reg.html'>press REGISTER to RETRY</a>");
}
else
{
Statement stmt1=con.createStatement();
stmt1.executeUpdate("insertintologin
values("+names+","+addr+","+no+","+id+","+pwd+")");
pw.println("YOUR DETAILS ARE ENTERED<br><br>");
pw.println("<a href='/tr/login.html'>press LOGIN to login</a>");
}
pw.println("</body></html>");
}
catch(Exception e)
{
resp.sendError(500,e.toString());
}
}
}

```

Catlog.java

```

import java.sql.*;
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class login extends HttpServlet
{
public void service(HttpServletRequest req,HttpServletResponse resp)
throws ServletException,IOException
{
PrintWriter pw=resp.getWriter();
pw.println("<html><body bgcolor='pink'>");
String title=req.getParameter("title");
try
{
Driver d=new oracle.jdbc.driver.OracleDriver();
DriverManager.registerDriver(d);
Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger");
Statement stmt=con.createStatement();
String sqlstmt="select id,password from login";
ResultSet rs=stmt.executeQuery(sqlstmt);

```

```

int flag=0;
while(rs.next())
{
    pw.println("div align=\"center\">");
    pw.println("TITLE :"+rs.getString(1)+"<br>");
    pw.println("AUTHOR :"+rs.getString(2)+"<br>");
    pw.println("VERSION :"+rs.getString(3)+"<br>");
    pw.println("PUBLISHER :"+rs.getString(4)+"<br>");
    pw.println("COST :"+rs.getString(5)+"<br>");
    pw.println("</div>");
    flag=1;
}
if(flag==0)
{
    pw.println("SORRY INVALID TITLE TRY AGAIN <br><br>");
    pw.println("<a href=\"/tr/catalog.html\">press HERE to RETRY</a>");
}
pw.println("</body></html>");
}
catch(Exception e)
{
    resp.sendError(500,e.toString());
}
}
}

```

Profile.java

```

import java.sql.*;
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class login extends HttpServlet
{
    public void service(HttpServletRequest req,HttpServletResponse resp)
    throws ServletException,IOException
    {
        PrintWriter pw=resp.getWriter();
        pw.println("<html><body bgcolor=\"pink\">");
        String id=req.getParamenter("id");
        try
        {

```

COURSE: CS8661 INTERNET PROGRAMMING LABORATORY

```
Driver d=new oracle.jdbc.driver.OracleDriver();
DriverManager.registerDriver(d);
Connection con=DriverManager.getConnection("jdbc:oracle:thin:
@localhost:1521:orcl","scott","tiger");
Statement stmt=con.createStatement();
String sqlstmt="select * from login where id="+id+"";
ResultSet rs=stmt.executeQuery(sqlstmt);
int flag=0;
pw.println("<br><br><br>");
while(rs.next())
{
pw.println("<div align=\"center\">");
pw.println("NAME :"+rs.getString(1)+"<br>");
pw.println("ADDRESS :"+rs.getString(2)+"<br>");
pw.println("PHONE NO :"+rs.getString(3)+"<br>");
pw.println("</div>");
flag=1;
}
if(flag==0)
{
pw.println("SORRY INVALID ID TRY AGAIN ID<br><br>");
pw.println("<a href=\"/tr/profile.html\">press HERE to RETRY</a>");
}
pw.println("</body></html>");
}
catch(Exception e)
{
resp.sendError(500,e.toString());
}
}
}
```

Order.java

```
import java.sql.*;
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class login extends HttpServlet
{
public void service(HttpServletRequest req,HttpServletResponse resp)
```

```
throws ServletException,IOException
{
    PrintWriter pw=resp.getWriter();
    pw.println("<html><body bgcolor=\"pink\"");
    String id=req.getParameter("id");
    String pwd=req.getParameter("pwd");
    String title=req.getParameter("title");
    String count1=req.getParameter("no");
    String date=req.getParameter("date");
    String cno=req.getParameter("cno");
    int count=Integer.parseInt(count1);
    try
    {
        Driver d=new oracle.jdbc.driver.OracleDriver();
        DriverManager.registerDriver(d);
        Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger");
        Statement stmt=con.createStatement();
        String sqlstmt="select id,password from login";
        ResultSet rs=stmt.executeQuery(sqlstmt);
        int flag=0,amount,x;
        while(rs.next())
        {
            if(id.equals(rs.getString(1))&&pwd.equals(rs.getString(2)))
            {
                flag=1;
            }
        }
        if(flag==0)
        {
            pw.println("SORRY INVALID ID TRY AGAIN ID<br><br>");
            pw.println("<a href=\"\"/tr/order.html\">press HERE to RETRY</a>");

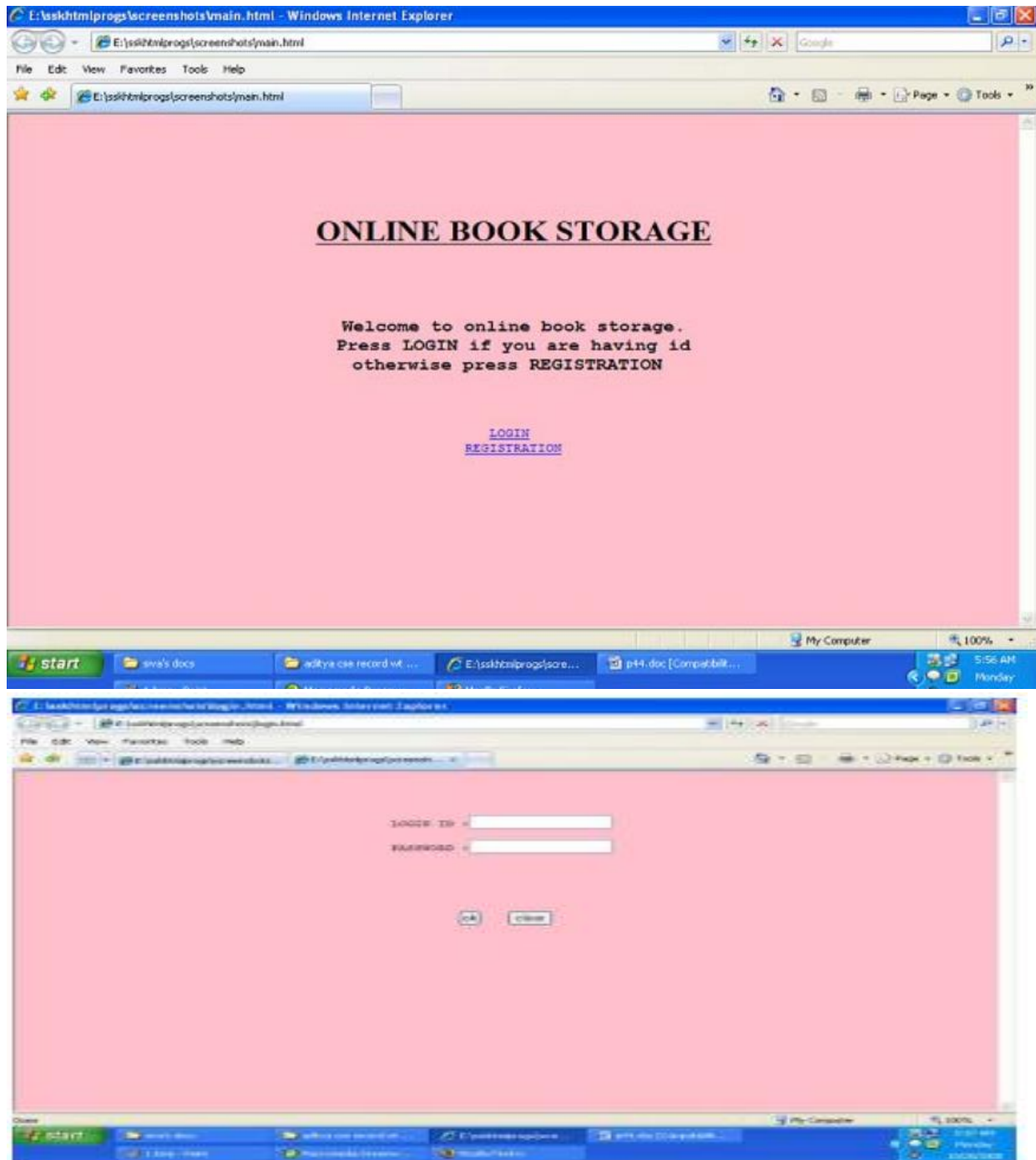
        }
        else
        {
            Statement stmt2=con.createStatement();
            String s="select cost from book where title="+title+"";
            ResultSet rs1=stmt2.executeQuery(s);
            int flag1=0;
            while(rs1.next())
            {
```

COURSE: CS8661 INTERNET PROGRAMMING LABORATORY

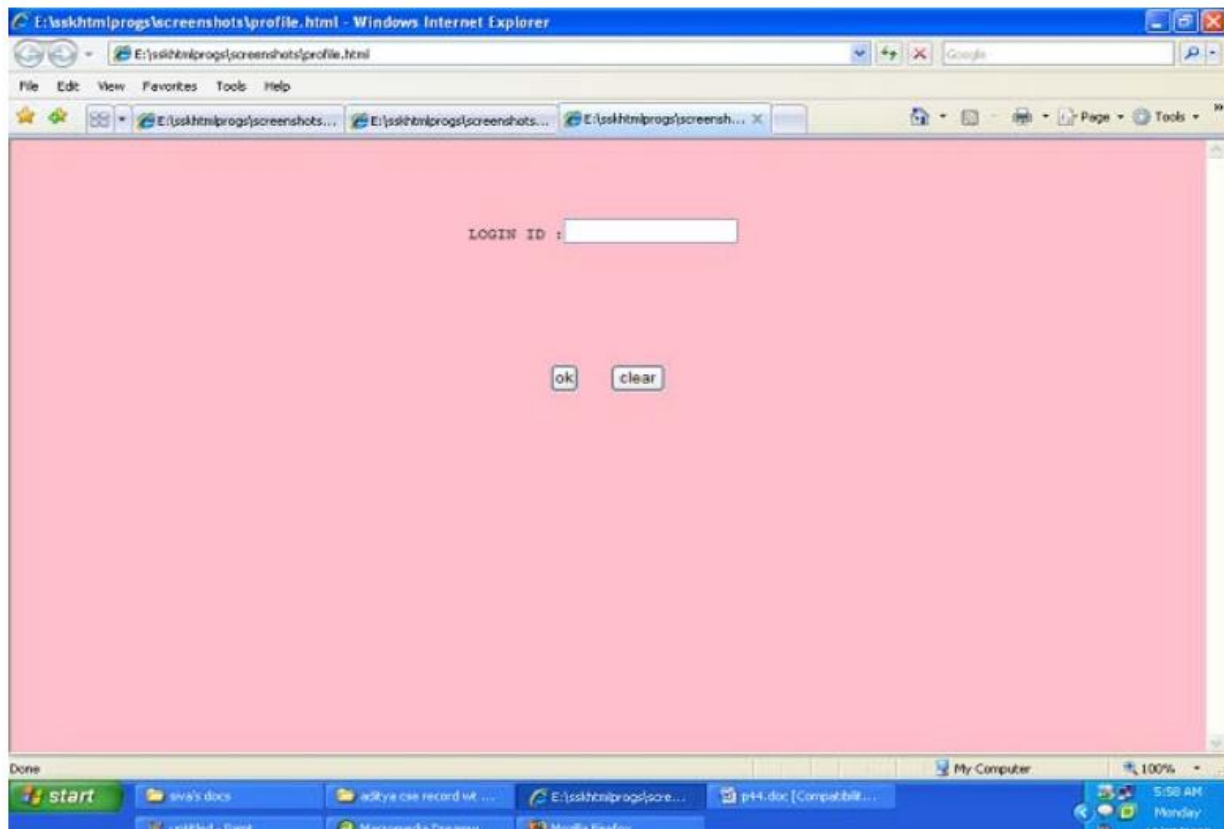
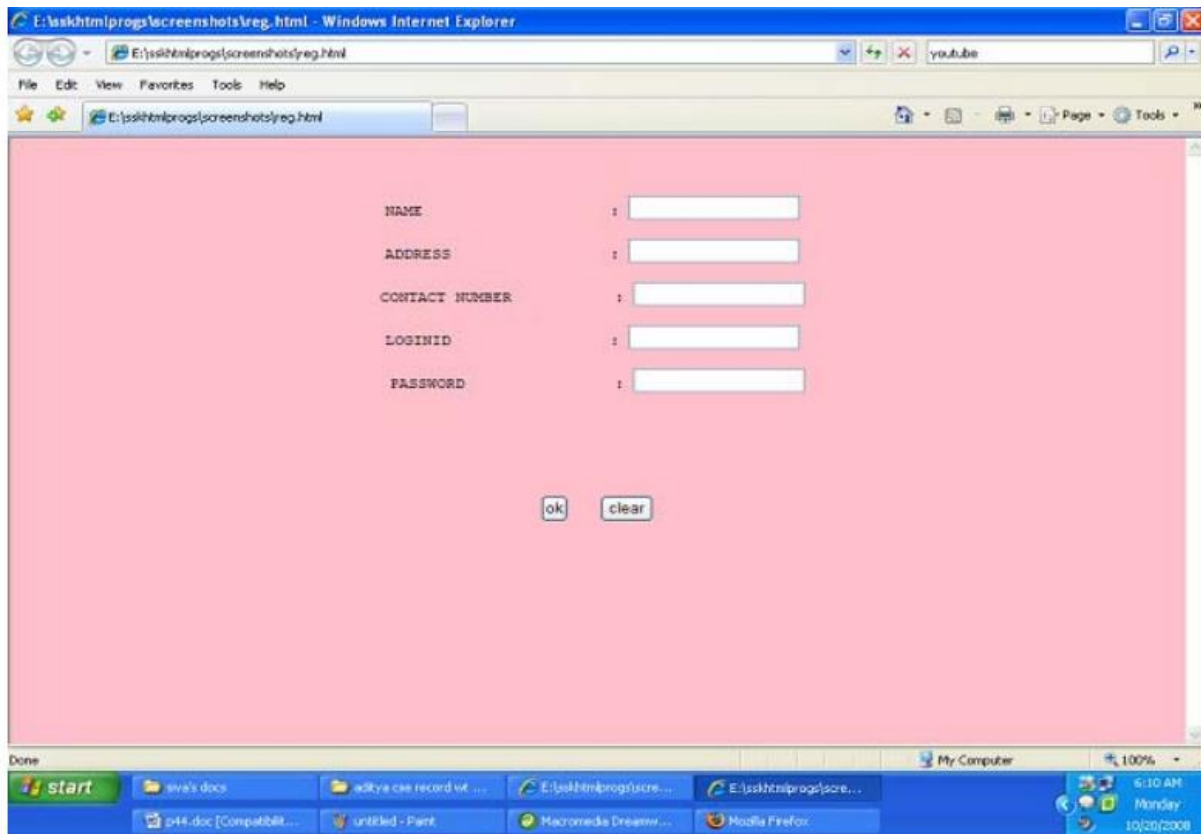
```
flag1=1;
x=Integer.parseInt(rs1.getString(1));
amount=count*x;
pw.println("AMOUNT :"+amount+"<br><br><br><br>");
Statement stmt1=con.createStatement();
stmt1.executeUpdate("insertintodetails
values('"+id+"','"+title+"'+amount+'','"+cno+"')");
pw.println("YOUR ORDER has taken<br>");
}
if(flag1==0)
{
pw.println("SORRY INVALID ID TRY AGAIN ID<br><br>");
pw.println("<a href=\\\"/tr/order.html\\\">press HERE to RETRY</a>");

}
}
pw.println("</body></html>");
con.close();
}
catch(Exception e)
{
resp.sendError(500,e.toString());
}
}
```

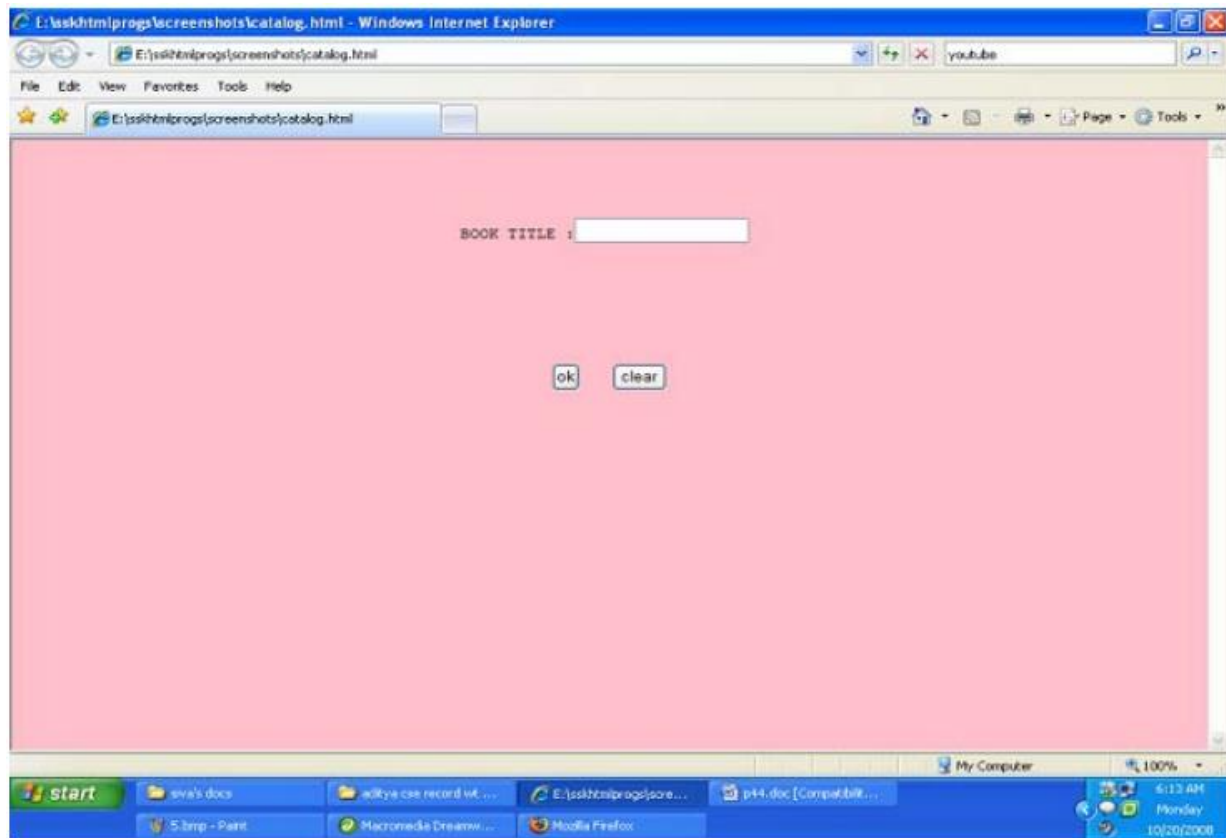

COURSE: CS8661 INTERNET PROGRAMMING LABORATORY

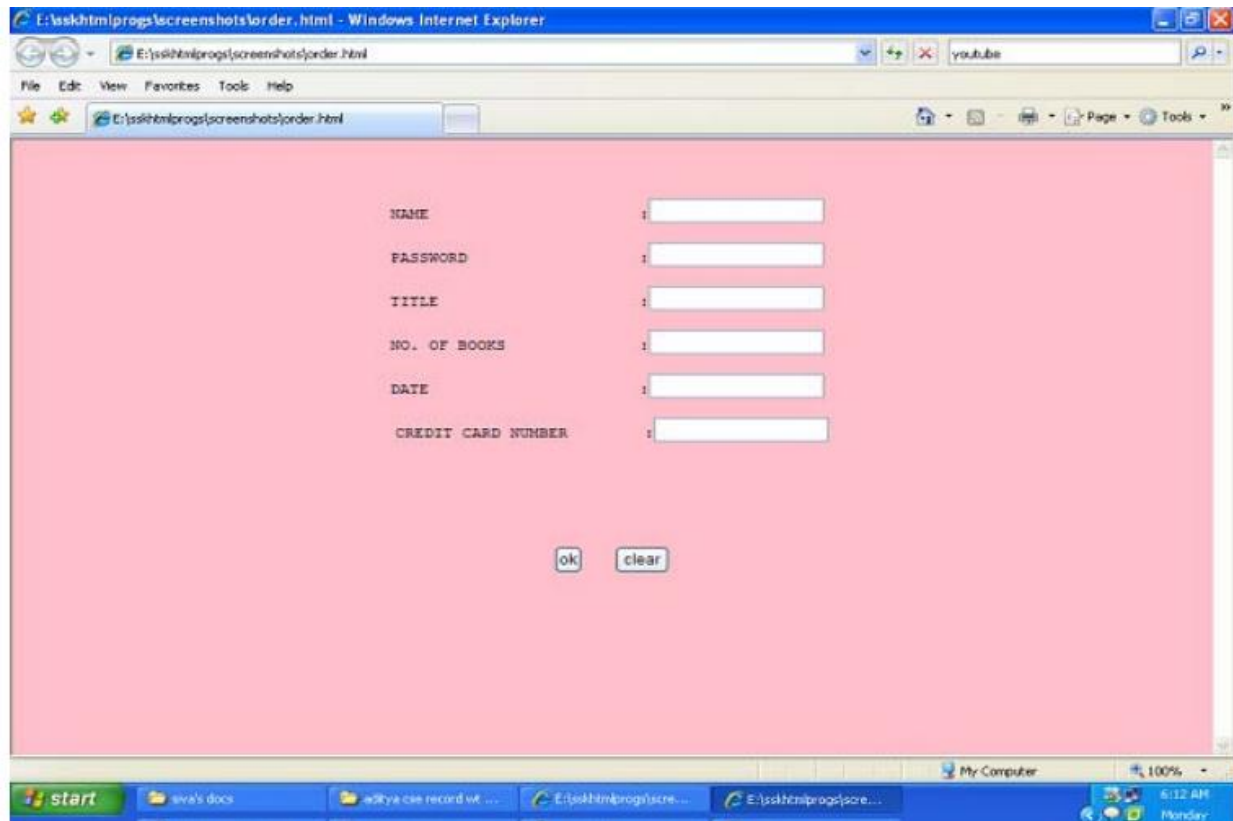


COURSE: CS8661 INTERNET PROGRAMMING LABORATORY



COURSE: CS8661 INTERNET PROGRAMMING LABORATORY





RESULT

Thus the Conversion of static WebPages into dynamic WebPages using servlets and cookies has been excuted and verified successfilly.

EX.NO:7 CONVERSION OF STATIC WEB PAGES INTO DYNAMIC WEB PAGES
DATE:

AIM:

To Redo the previous task using JSP by converting the static web pages into dynamic web pages. Create a database with user information and books information and books information. The books catalogue should be dynamically loaded from the database. Follow the MVC architecture while doing the website.

PROCEDURE:

- 1) Create your own directory under tomcat/webapps (e.g. tr1)
- 2) Copy the html files in tr1
- 3) Copy the jsp files also into tr1
- 4) Start tomcat give the following command Catalina.bat run At install-dir/bin at I.E give url as <http://localhost:8081/tr1/main.html>

PROGRAM:

Main.html:

```
<html>
<body bgcolor="pink">
<br><br><br><br><br><br>
<h1 align="center"><u>ONLINE BOOK STORAGE</u></h1><br><br><br>
<h2 align="center"><PRE>
<b> Welcome to online book storage.
    Press LOGIN if you are having id
    Otherwise press REGISTRATION
</b></PRE></h2>
<br><br><pre>
<div align="center"><a href="/tr/login.html">LOGIN</a>
href="/tr/login.html">REGISTRATION</a></div></pre>
</body></html>
```

Login.html:

```
<html>
<body bgcolor="pink"><br><br><br>
<form name="myform" method="post" action="/tr1/login.jsp">
<div align="center"><pre>
LOGIN ID : <input type="password" name="pwd"></pre><br><br>
PASSWORD : <input type="password" name="pwd"></pre><br><br>
</div>
<br><br>
<div align="center">
```

```
<input type="submit" value="ok"
onClick="validate()">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="reset"
value="clear">
</form>
</body>
</html>
```

Reg.html:

```
<html>
<body bgcolor="pink"><br><br>
<form name="myform" method="post" action="/tr1/reg.jsp">
  <div align="center"><pre>
    NAME      :<input type="text" name="name"><br>
    ADDRESS   :<input type="text" name="addr"><br>
    CONTACT NUMBER :<input type="text" name="phno"><br>
    LOGIN ID   :<input type="text" name="id"><br>
    PASSWORD  :<input type="password" name="pwd"></pre><br><br>
  </div>
  <br><br>
  <div align="center">
    <input type="submit" value="ok"
onClick="validate()">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="reset"
value="clear">
  </div>
</form>
</body>
</html>
```

Profile.html:

```
<html>
<body bgcolor="pink"><br><br>
<form name="myform" method="post" action="/tr1/profile.jsp">
  <div align="center"><pre>
    LOGIN ID   :<input type="text" name="id"><br>
  </pre><br><br>
  </div>
  <br><br>
  <div align="center">
    <input type="submit" value="ok"
onClick="validate()">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="reset"
value="clear">
  </div>
</form>
</body>
</html>
```

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

Catalog.html:

```
<html>
<body bgcolor="pink"><br><br><br>
<form method="post" action="/tr1/catalog.jsp">
  <div align="center"><pre>
    BOOK TITLE : <input type="text" name="title"><br>
  </pre><br><br>
</div>
<br><br>
<div align="center">
  <input type="submit" value="ok"
name="button1">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="reset" value="clear"
name="button2">
</form>
</body>
</html>
```

Order.html:

```
<html>
<body bgcolor="pink"><br><br><br>
<form method="post" action="/tr1/order.jsp">
  <div align="center"><pre>
    LOGIN ID      :<input type="text" name="id"><br>
    PASSWORD     :<input type="password" name="pwd"><br>
    TITLE        :<input type="text" name="title"><br>
    NO. OF BOOKS  :<input type="text" name="no"><br>
    DATE         :<input type="text" name="date"><br>
    CREDIT CARD NUMBER :<input type="password" name="cno"><br></pre><br><br>
  </div>
  <br><br>
  <div align="center">
    <input type="submit" value="ok"
name="button1">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="reset"
value="clear" name="button2">
  </div>
</form>
</body>
</html>
```

```

</form>
</body>
</html>

```

Login.jsp:

```

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

    out.println("<html><body bgcolor=\"pink\">");
    String id=request.getParameter("id");
    String pwd=request.getParameter("pwd");
    Driver d=new oracle.jdbc.driver.OracleDriver();
    DriverManager.registerDriver(d);
    Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger");
    Statement stmt=con.createStatement();
    String sqlstmt="select id,password from login where id="+id+" and password="+pwd+"";
    ResultSet rs=stmt.executeQuery(sqlstmt);
    int flag=0;
    while(rs.next())
    {
        flag=1;
    }
    if(flag==0)
    {
        out.println("SORRY INVALID ID TRY AGAIN ID<br><br>");
        out.println(" <a href=\"/tr1/login.html\">press LOGIN to RETRY</a>");
    }
    else
{
    out.println("VALID LOGIN ID<br><br>");
        out.println("<h3><ul>");
        out.println("<li><a href=\"/profile.html\"><fontcolor=\"black\">USER
PROFILE</font></a></li><br><br>");
        out.println("<li><a href=\"/catalog.html\"><fontcolor=\"black\">BOOKS
CATALOG</font></a></li><br><br>");
        out.println("<li><a href=\"/order.html\"><fontcolor=\"black\">ORDER
CONFIRMATION</font></a></li><br><br>");
        out.println("</ul>");
    }
}

```



```
out.println("<body></html>");
%>
```

Reg.jsp:

```
%@page import="java.sql.*"%
%@page import="java.io.*"%
<%
out.println("<html><body bgcolor=\"pink\">");
String name=request.getParameter("name");
String addr=request.getParameter("addr");
String phno=request.getParameter("phno");
String id=request.getParameter("id");
String pwd=request.getParameter("pwd");
int no=Integer.parseInt(phno);
Driver d=new oracle.jdbc.driver.OracleDriver();
DriverManager.registerDriver(d);
Connection con=
DriverManager.getConnection ("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger");
Statement stmt=con.createStatement();
String sqlstmt="select id from login";
ResultSet rs=stmt.executeQuery(sqlstmt);
int flag=0;
while(rs.next())
{
    if(id.equals(rs.getString(1)))
    {
        flag=1;
    }
}
if(flag==1)
{
    out.println("SORRY LOGIN ID ALREADY EXISTS TRY AGAIN WITH NEW ID
<br><br>");
    out.println("<a href=\"/tr1/reg.html\">press REGISTER to RETRY</a>");
}
else
{
    Statement stmt1=con.createStatement ();
    stmt1.executeUpdate ("insert into login values
("+name+", "+addr+", "+no+", "+id+", "+pwd+)");
    out.println ("YOU DETAILS ARE ENTERED <br><br>");
}
```

```

        out.println("<a href =\"/tr1/login.html\">press LOGIN to login</a>");
    }
    out.println("</body></html>");
%>

```

Profile.jsp:

```

<%@page import="java.sql.*"%>
<%@page import="java.io.*"%>
<%
    out.println("<html><body bgcolor=\"pink\">");
    String id=request.getParameter("id");
    Driver d=new oracle.jdbc.driver.OracleDriver();
    DriverManager.regiserDriver(d);
Connection con=
DriverManager.getConnection ("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger");
    Statement stmt=con.createStatement ();
    String sqlstmt="select * from login where id="+id+"";
    ResultSet rs=stmt.executeQuery (sqlstmt);
    int flag=0;
    while(rs.next())
    {
        out.println("<div align=\"center\">");
        out.println("NAME      :"+rs.getString(1)+"<br>");
        out.println("ADDRESS   :"+rs.getString(2)+"<br>");
        out.println("PHONE NO  :"+rs.getString(3)+"<br>");
        out.println("</div>");
        flag=1;
    }
    if(flag==0)
    {
        out.println("SORRY INVALID ID TRY AGAIN ID <br><br>");
        out.println("<a href =\"/tr1/profile.html\">press HERE to RETRY </a>");
    }
    out.println("</body></html>");
%>

```

Catalog.jsp:

```

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

```

COURSE: CS8661 INTERNET PROGRAMMING LABORATORY

```
out.println("<html><body bgcolor=\"pink\">");
String title=request.getParameter("title");
Driver d=new oracle.jdbc.driver.OracleDriver ();
DriverManager.regiserDriver (d);
Connection con=
DriverManager.getConnection ("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger");
Statement stmt=con.createStatement ();
String sqlstmt="select * from book where title="+title+"";
ResultSet rs=stmt.executeQuery (sqlstmt);
int flag=0;
while(rs.next())
{
    out.println("<div align=\"center\">");
    out.println("TITLE      :"+rs.getString(1)+"<br>");
    out.println("AUTHOR   :"+rs.getString(2)+"<br>");
    out.println("VERSION:"+rs.getString(3)+"<br>");
    out.println("PUBLISHER  :"+rs.getString(4)+"<br>");
    out.println("COST    :"+rs.getString(5)+"<br>");
    out.println("</div>");
    flag=1;
}
if(flag==0)
{
    out.println("SORRY INVALID ID TRY AGAIN ID <br><br>");
    out.println("<a href=\"/tr1/catalog.html\">press HERE to RETRY </a>");
}
out.println("</body></html>");
%>
```

Order.jsp:

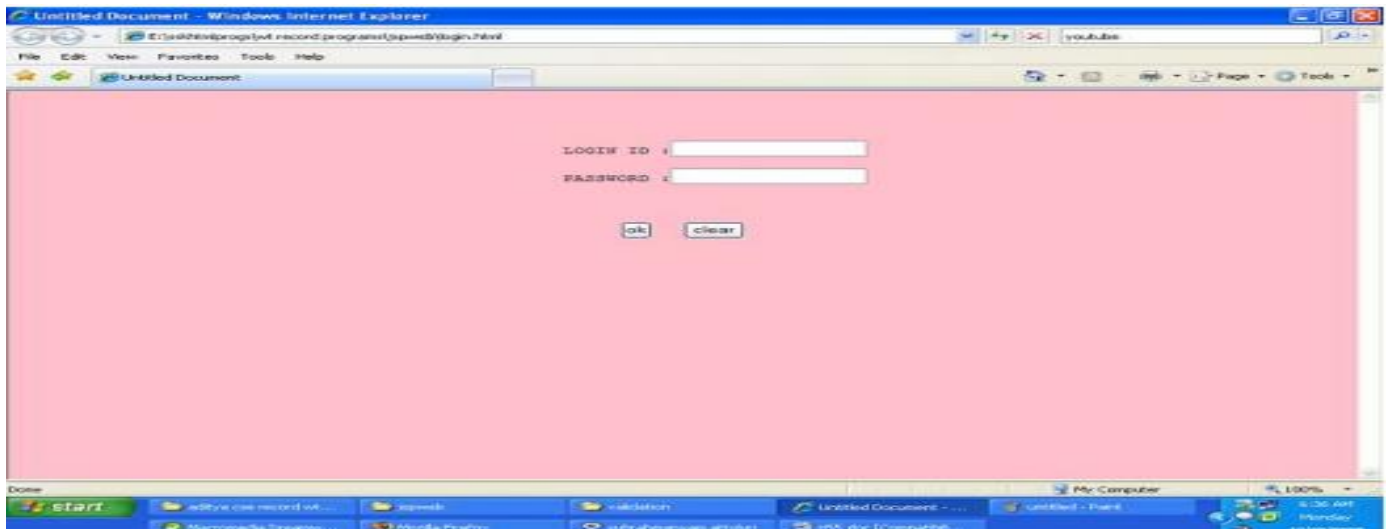
```
<%@page import="java.sql.*"%>
<%@page import="java.io.*"%>
<%
    out.println("<html><body bgcolor=\"pink\">");
    String id=request.getParameter("id");
    String pwd=request.getParameter("pwd");
    String title=request.getParameter("title");
    String count1=request.getParameter("no");
    String date=request.getParameter("date");
    String cno=request.getParameter("cno");
    int count=Integer.parseInt(count1);
```

COURSE: CS8661 INTERNET PROGRAMMING LABORATORY

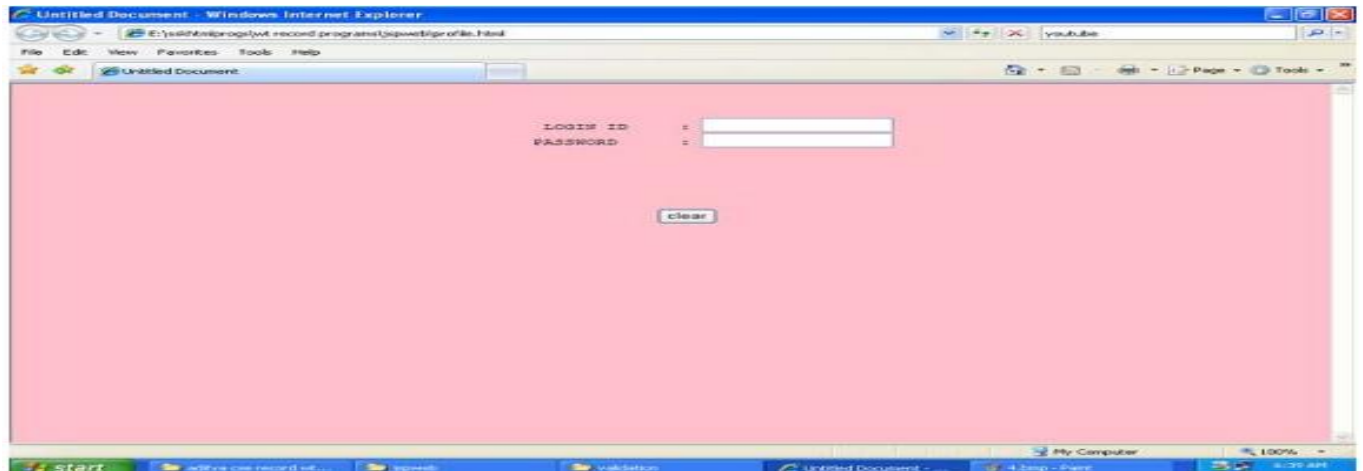
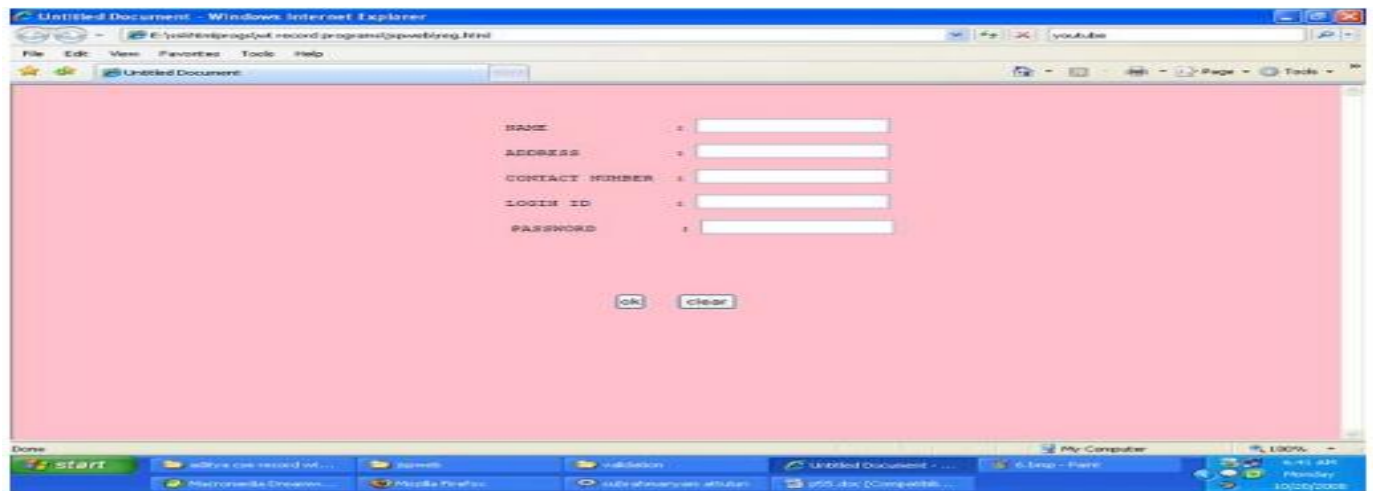
```
Driver d=new oracle.jdbc.driver.OracleDriver ();
DriverManager.regiserDriver (d);
Connection con=
DriverManager.getConnection ("jdbc:oracle:thin:@localhost:1521:orcl","scott","tiger");
Statement stmt=con.createStatement ();
String sqlstmt="select id, password from login";
ResultSet rs=stmt.executeQuery (sqlstmt);
int flag=0,amount,x;
while(rs.next())
{
    if(id.equals(rs.getString(1))&& pwd.equals(rs.getString(2)))
    {
        flag=1;
    }
}
if(flag==0)
{
    out.println("SORRY INVALID ID TRY AGAIN ID <br><br>");
    out.println("<a href='\"/tr1/order.html\"'>press HERE to RETRY </a>");
}
else
{
    Statement stmt2=con.createStatement();
    String s="select cost from book where title='"+title+"'";
    ResultSet rs1=stmt2.executeQuery(s);
    int flag1=0;
    while(rs1.next())
    {
        flag1=1;
        x=Integer.parseInt(rs1.getString(1));
        amount=count*x;
        out.println("AMOUNT  :"+amount+"<br><br><br><br>");
        Statement stmt1=con.createStatement ();
        stmt1.executeUpdate ("insert into details
(+id+','"+title+','"+amount+','"+date+','"+cno+'))");
        out.println ("YOU ORDER HAS TAKEN<br>");
    }
}
if(flag1==0)
{
    out.println("SORRY INVALID BOOK TRY AGAIN <br><br>");
    out.println("<a href='\"/tr1/order.html\"'>press HERE to RETRY </a>");
}
}
```

```
} out.println (“</body></html>”);%>
```

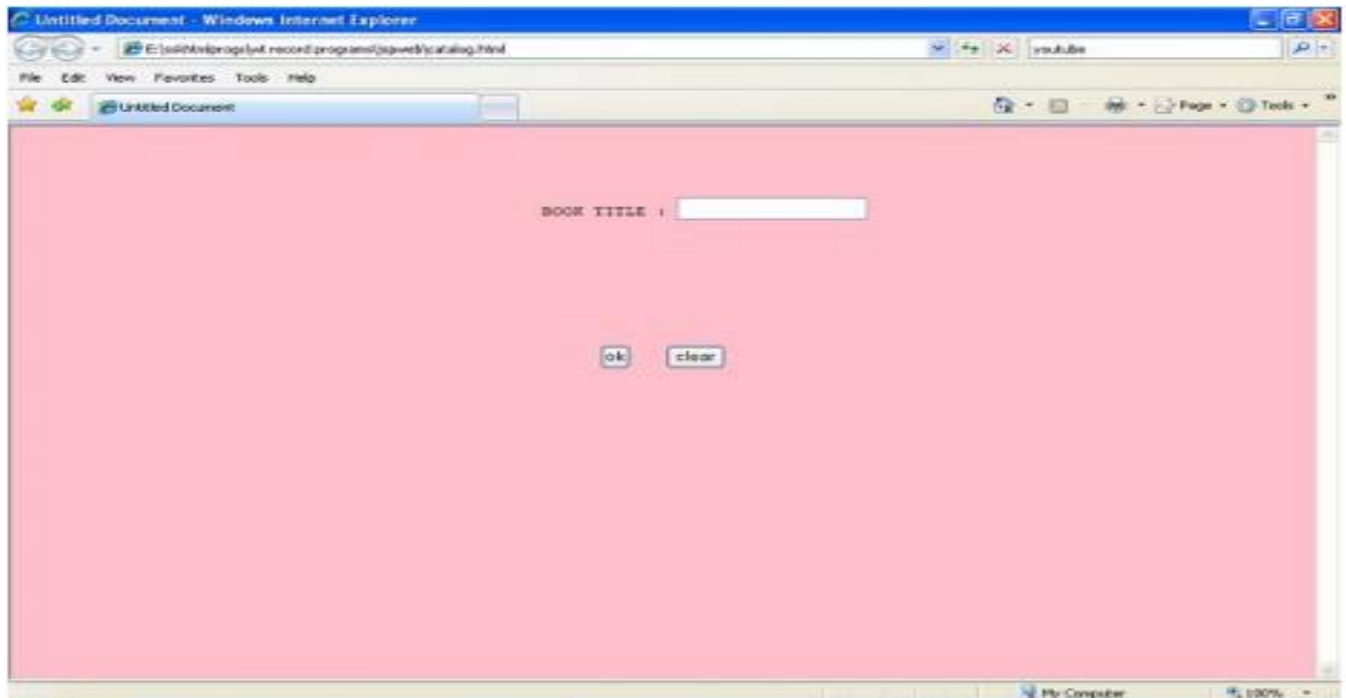
SAMPLE OUTPUT:



COURSE: CS8661 INTERNET PROGRAMMING LABORATORY



COURSE: CS8661 INTERNET PROGRAMMING LABORATORY



RESULT:

Thus the database with user information and books information has been created the books catalogue has been dynamically loaded from the database.

EX.NO: 8 USER INFORMATION AND INFORMATION RETRIEVAL FROM XML
DATE:

AIM:

To write a program, that takes user id as input and displays the user details by taking the user information from the XML document.

PROCEDURE:

Step 1: Start the xml program.

Step 2: Create the 10 user information in the employee tag and insert the information about the employee in <cse> tag.

Step 3: Then save an XML document by Info.xml which contains 10 user information..

Step 3: Next to retrieve the information by an user id use the java script tag by specifying the script language.

Step 4: Use the function readXMLData () in which information of 10users can be retrieved by using an id node.

Step 5: Then validate idnode, namenode, department node, designation node by using xmlDocumentObject.getElementsByTagName

Step 6: Next save the file as elements.html and open in the new browser window.

Step 7: Enter id number in the search prompt then the corresponding user details is displayed.

Step 8: Stop the program.

PROGRAM:

User Information

Info.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<cse>
<id>2</id>
<name>Mani</name>
<department>cse</department>
<designation>AP</designation>
<id>1</id>
<name>Raj</name>
<department>cse</department>
<designation>AP</designation>
<id>3</id>
<name>jaya</name>
<department>cse</department>
<designation>AP</designation>
<id>4</id>
<name>madhan</name>
<department>cse</department>
```



```
<designation>AP</designation>
<id>5</id>
<name>jaga</name>
<department>cse</department>
<designation>AP</designation>
<id>6</id>
<name>kareem</name>
<department>cse</department>
<designation>AP</designation>
<id>7</id>
<name>amala</name>
<department>cse</department>
<designation>AP</designation>
<id>8</id>
<name>geetha</name>
<department>cse</department>
<designation>AP</designation>
<id>9</id>
<name>dipthi</name>
<department>cse</department>
<designation>AP</designation>
<id>10</id>
<name>bala</name>
<department>cse</department>
<designation>AP</designation>
</cse>
```

Information Retrieval **Elements.html**

```
<HTML>
<HEAD>
<TITLE>Searching for Elements </TITLE>
<SCRIPT LANGUAGE="JavaScript">
function readXMLData()
{
var xmlDocumentObject, idnode, namenode, designationnode, departmentnode;
xmlDocumentObject=new XMLHttpRequest();
xmlDocumentObject.open("GET","info.xml",false);
xmlDocumentObject.send();
xmlDocumentObject=xmlDocumentObject.responseXML;
idnode = xmlDocumentObject.getElementsByTagName("id");
departmentnode = xmlDocumentObject.getElementsByTagName("department");
designationnode = xmlDocumentObject.getElementsByTagName("designation");
namenode = xmlDocumentObject.getElementsByTagName("name");
for (i = 0; i < idnode.length; i++)
{
output=idnode[i].firstChild.nodeValue;
if (output == document.getElementById("myText").value)
{
```

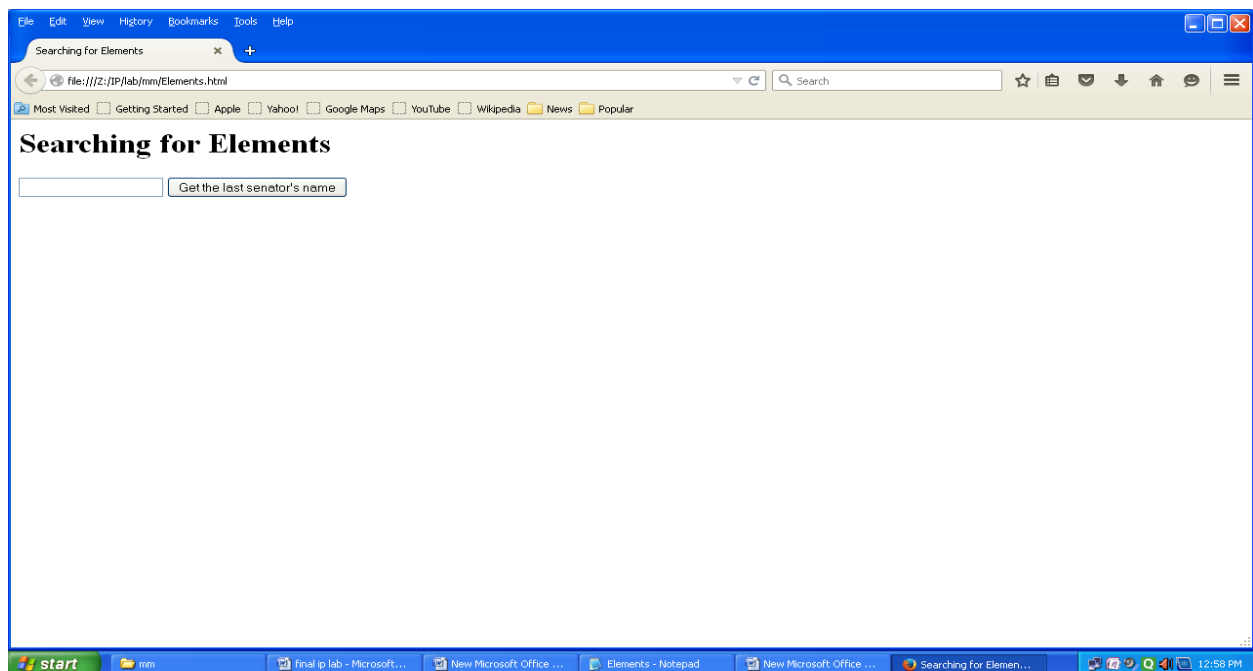
```
displayDIV.innerHTML = idnode[i].firstChild.nodeValue + " " +  
namenode[i].firstChild.nodeValue + " " + departmentnode[i].firstChild.nodeValue + " " +  
designationnode[i].firstChild.nodeValue  
}  
}  
}  
</SCRIPT>  
</HEAD>  
<BODY>  
<H1>Searching for Elements</H1>  
<input type="text" id="myText" value="">  
<input type="BUTTON" VALUE="Get the last senator's name" ONCLICK="readXMLData()">  
<P>  
<DIV ID="displayDIV"> </DIV>  
</BODY>  
</HTML>
```

Execution Steps:

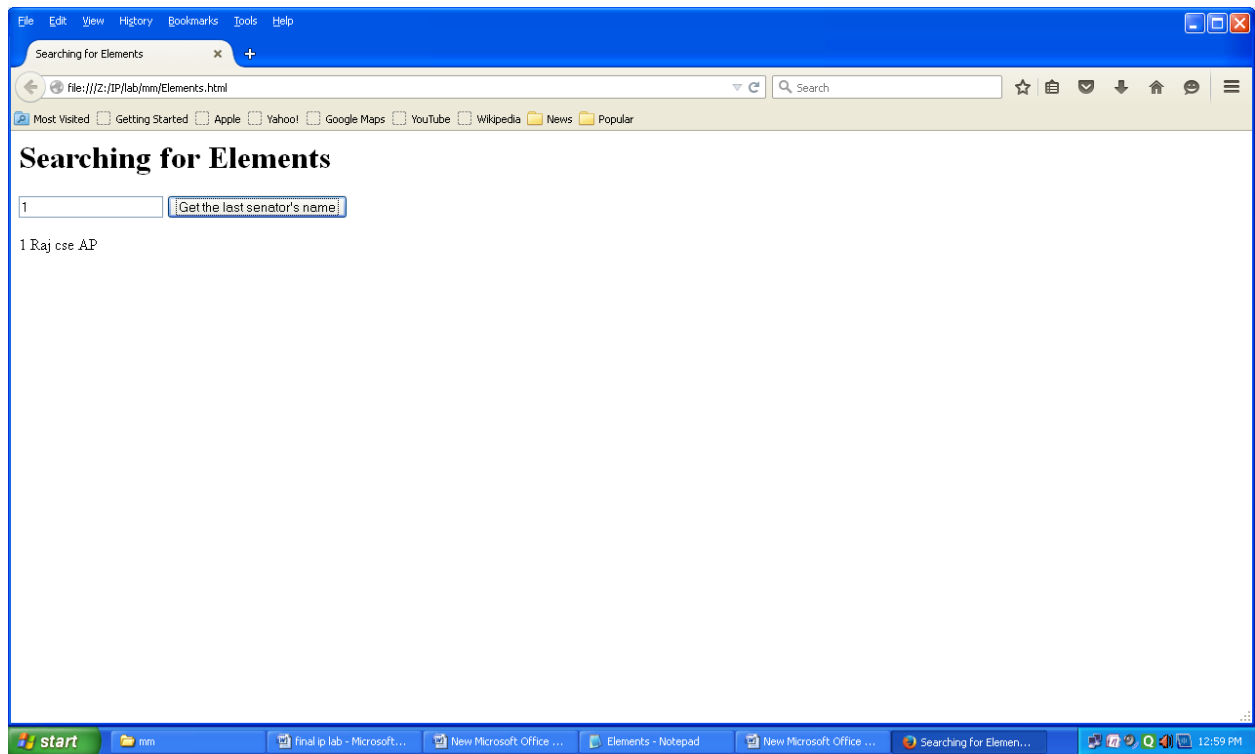
1. Create and save xml program named Info.xml which contains 10 users information.
2. Then use a html program by specifying the javascript which takes user Id as an input and returns the User details by taking the user information from the xml document.
3. Open Elements.html in a new browser window it asks for an id number.
- 4. If an id number is given the corresponding user details is retrieved from the xml document.**

SAMPLE OUTPUT:

**Output screens for XML document
Elements.html**



COURSE: CS8661 INTERNET PROGRAMMING LABORATORY



RESULT:

Thus the program which takes user Id as an input and returns the user details by taking the user information from the XML document is done successfully and output was verified.

Course Teacher: Mr.D.Vaduganathan., Mr.T.Arunkumar., Assistant Professor, Department of CSE,
Excel Engineering College, Komarapalayam-637303

EX.NO:9.1 VALIDATE THE FORM USING PHP REGULAR EXPRESSION

DATE:

AIM:

To write a program, that takes users information as a form and validates the form using PHP regular expression.

PROCEDURE:

Step1:Create the form using <form> tag

Step2: Form tag should contain users input fields

Step3: Validate the form using Regular expression

PROGRAM:

```
<!DOCTYPE HTML>
<html>
<head>
<style>
.error {color: #FF0000;}
</style>
</head>
<body>

<?php
// define variables and set to empty values
$nameErr = $emailErr = $genderErr = $websiteErr = "";
$name = $email = $gender = $comment = $website = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["name"])) {
        $nameErr = "Name is required";
    } else {
        $name = test_input($_POST["name"]);
        // check if name only contains letters and whitespace
        if (!preg_match("/^[a-zA-Z ]*$/",$name)) {
            $nameErr = "Only letters and white space allowed";
        }
    }
}
```

```

if (empty($_POST["email"])) {
    $emailErr = "Email is required";
} else {
    $email = test_input($_POST["email"]);
    // check if e-mail address is well-formed
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $emailErr = "Invalid email format";
    }
}

if (empty($_POST["website"])) {
    $website = "";
} else {
    $website = test_input($_POST["website"]);
    // check if URL address syntax is valid
    if (!preg_match("/^b(?:(:?https?|ftp):\\\/\\\/www\\.)[-a-z0-9+&@#\/%?~_!:.,:]*[-a-z0-9+&@#\/%?~_!]/i",$website)) {
        $websiteErr = "Invalid URL";
    }
}

if (empty($_POST["comment"])) {
    $comment = "";
} else {
    $comment = test_input($_POST["comment"]);
}

if (empty($_POST["gender"])) {
    $genderErr = "Gender is required";
} else {
    $gender = test_input($_POST["gender"]);
}

function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
?>

```

```
<h2>PHP Form Validation Example</h2>
<p><span class="error">* required field</span></p>
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
  Name: <input type="text" name="name">
  <span class="error">* <?php echo $nameErr;?></span>
  <br><br>
  E-mail: <input type="text" name="email">
  <span class="error">* <?php echo $emailErr;?></span>
  <br><br>
  Website: <input type="text" name="website">
  <span class="error"><?php echo $websiteErr;?></span>
  <br><br>
  Comment: <textarea name="comment" rows="5" cols="40"></textarea>
  <br><br>
  Gender:
  <input type="radio" name="gender" value="female">Female
  <input type="radio" name="gender" value="male">Male
  <input type="radio" name="gender" value="other">Other
  <span class="error">* <?php echo $genderErr;?></span>
  <br><br>
  <input type="submit" name="submit" value="Submit">
</form>

<?php
echo "<h2>Your Input:</h2>";
echo $name;
echo "<br>";
echo $email;
echo "<br>";
echo $website;
echo "<br>";
echo $comment;
echo "<br>";
echo $gender;
?>

</body>
</html>
```

SAMPLE OUTPUT:

PHP Form Validation Example

*** required field**

Name: *** Name is required**

E-mail: *** Email is required**

Website:

Comment:

Gender: ☐ Female ☐ Male ☐ Other *** Gender is required**

Your Input:

RESULT

Thus the program to takes users information as a form and validates the form using PHP regular expression has been executed and verified successfully.

EX.NO:9.2

PHP STORES A FORM DATA INTO DATABASE

DATE:

AIM:

To write a program, that takes users information as a form data and stores into database.

PROCEDURE:

Step1: Create the form using <form> tag with user details

Step2: Form tag should contain users input fields

Step3: Create the database fields with user details in mysql.

Step4: Connect the database with front end using msqconnect ().

Step5:Stores the user data by insert query.

PROGRAM:

insert.php

```
<!DOCTYPE html>
<html>
<head>
<title>PHP insertion</title>
<link href="css/insert.css" rel="stylesheet">
</head>
<body>
<div class="maindiv">
<!--HTML Form -->
<div class="form_div">
<div class="title">
<h2>Insert Data In Database Using PHP.</h2>
</div>
<form action="insert.php" method="post">
<!-- Method can be set as POST for hiding values in URL-->
<h2>Form</h2>
<label>Name:</label>
<input class="input" name="name" type="text" value="">
<label>Email:</label>
<input class="input" name="email" type="text" value="">
<label>Contact:</label>
<input class="input" name="contact" type="text" value="">
```



```
<label>Address:</label>
<textarea cols="25" name="address" rows="5"></textarea><br>
<input class="submit" name="submit" type="submit" value="Insert">
</form>
</div>
</div>
</body>
</html>
```

PHP Code segment

```
<?php
$connection = mysql_connect("localhost", "root", ""); // Establishing Connection with Server
$db = mysql_select_db("colleges", $connection); // Selecting Database from Server
if(isset($_POST['submit'])){ // Fetching variables of the form which travels in URL
$name = $_POST['name'];
$email = $_POST['email'];
$contact = $_POST['contact'];
$address = $_POST['address'];
if($name != "" || $email != ""){
//Insert Query of SQL
$query = mysql_query("insert into students(student_name, student_email, student_contact,
student_address) values ('$name', '$email', '$contact', '$address')");
echo "<br/><br/><span>Data Inserted successfully...!!</span>";
}
else{
echo "<p>Insertion Failed <br/> Some Fields are Blank....!!</p>";
}
}
mysql_close($connection); // Closing Connection with Server
?>
```

insert.css

```
maindiv {
margin:30px auto;
width:980px;
height:500px;
background:#fff;
padding-top:20px;
font-family:'Droid Serif',serif;
font-size:14px
}
.title {
width:500px;
height:70px;
```

COURSE: CS8661 INTERNET PROGRAMMING LABORATORY

```
text-shadow:2px 2px 2px #cfcfcf;
font-size:16px;
text-align:center
}
.form_div {
width:70%;
float:left
}
form {
width:300px;
border:1px dashed #aaa;
padding:10px 30px 40px;
margin-left:70px;
background-color:#f0f8ff
}
form h2 {
text-align:center;
text-shadow:2px 2px 2px #cfcfcf
}
textarea {
width:100%;
height:60px;
border-radius:1px;
box-shadow:0 0 1px 2px #123456;
margin-top:10px;
padding:7px;
border:none
}
.input {
width:100%;
height:30px;
border-radius:2px;
box-shadow:0 0 1px 2px #123456;
margin-top:10px;
padding:7px;
border:none;
margin-bottom:20px
}
.submit {
color:#fff;
border-radius:3px;
background:#1F8DD6;
padding:5px;
margin-top:40px;
border:none;
width:100%;
height:30px;
box-shadow:0 0 1px 2px #123456;
```

```
font-size:18px
}
p {
color:red;
text-align:center
}
span {
text-align:center;
color:green
}
```

OUTPUT



The screenshot shows a web form titled "Form" with a light blue background and a dashed border. It contains four input fields labeled "Name:", "Email:", "Contact:", and "Address:". Below the input fields is a blue button labeled "Insert". At the bottom, a green message reads "Data Inserted successfully...!!".

RESULT

Thus the program has been executed and verified successfully.

Ex. No: 10

OPINIONS FOR CONSUMER PRODUCT

DATE:

AIM:

To write a web service for finding what people think by asking 500 people's opinion for any consumer product

PROCEDURE:

Step1: Create the form using <form> tag

Step2: Form tag should contain consumer products' opinion as input fields

Step3: Validate the form using Regular expression

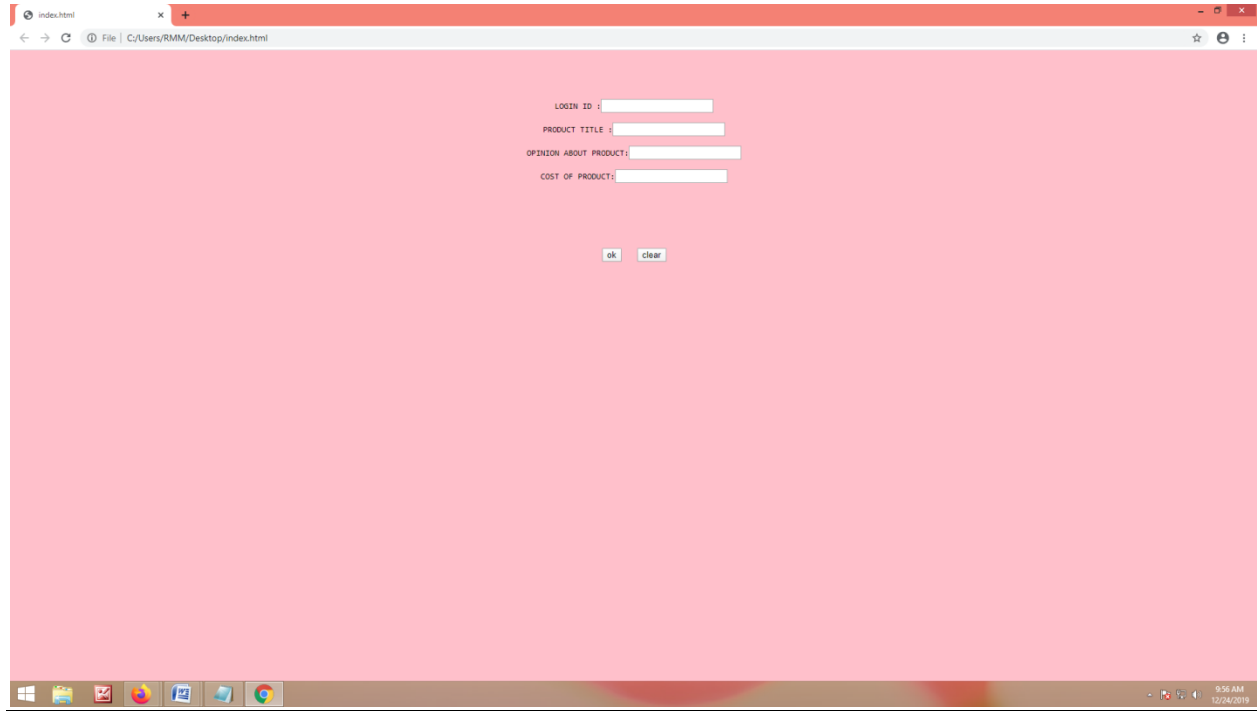
Step4: Get the 500 people opinion for consumer

Step5: Store the data into Database.

PROGRAM

```
<html>
<body bgcolor="pink"><br><br><br>
<script language="javascript">
function validate()
{
var flag=1;
if(document.myform.id.value=="||
document.myform.title.value=="||
document.myform.opinion.value=="||
document.myform.cost.value=="")
{
flag=0;
}
str=document.myform.title.value;
var str1=document.myform.cost.value;
if(!((str=="c"&& str1==444) || (str=="jsp" && str1==555)))
{
flag=0;
```


SAMPLE OUTPUT:



A screenshot of a web browser window displaying a form for product feedback. The browser's address bar shows the file path "C:/Users/RMM/Desktop/index.html". The form is centered on a light pink background and contains the following fields and buttons:

- LOGIN ID :
- PRODUCT TITLE :
- OPINION ABOUT PRODUCT:
- COST OF PRODUCT:
- Two buttons labeled "ok" and "clear" are positioned below the input fields.

The Windows taskbar is visible at the bottom of the screen, showing the Start button and several application icons. The system clock in the bottom right corner indicates the time is 9:56 AM on 12/24/2019.

RESULT:

Thus the program to web service for finding what people think by asking 500 people's opinion for any consumer product

AUGMENTED EXRIMENTS

1. Programs using AJAX
2. Consider a case where we have two web Services- an airline service and a travel agent and the travel agent is searching for an airline. Implement this scenario using Web Services and Data base.

Ex. No 1

DATE:

AJAX.

AIM:

To write a java script program for a AJAX.

ALGORITHM:

1. Start the program.
2. A scripting language that is commonly hosted in a browser to add Interactivity to HTML PAGES.
3. Defines the structure of a webpage as a set of programmable objects that can be accessed through javascript.
4. Allows a client-side script to perform and httprequest.
5. AJAX applications use xmlhttprequest object to perform asynchronous requests to the server as opposed to performing a full page refresh.
6. Display the result.
7. Stop the program.

PROGRAM:

```
<html>
<head>
<script type="text/javascript">
functionloadXMLDoc()
{
if (window.XMLHttpRequest)
xmlhttp=new XMLHttpRequest();
else
xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
xmlhttp.onreadystatechange=function()
{
if (xmlhttp.readyState==4 &&xmlhttp.status==200)
{
document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
}
}
xmlhttp.open("GET","new.txt",true);
xmlhttp.send();
}
```

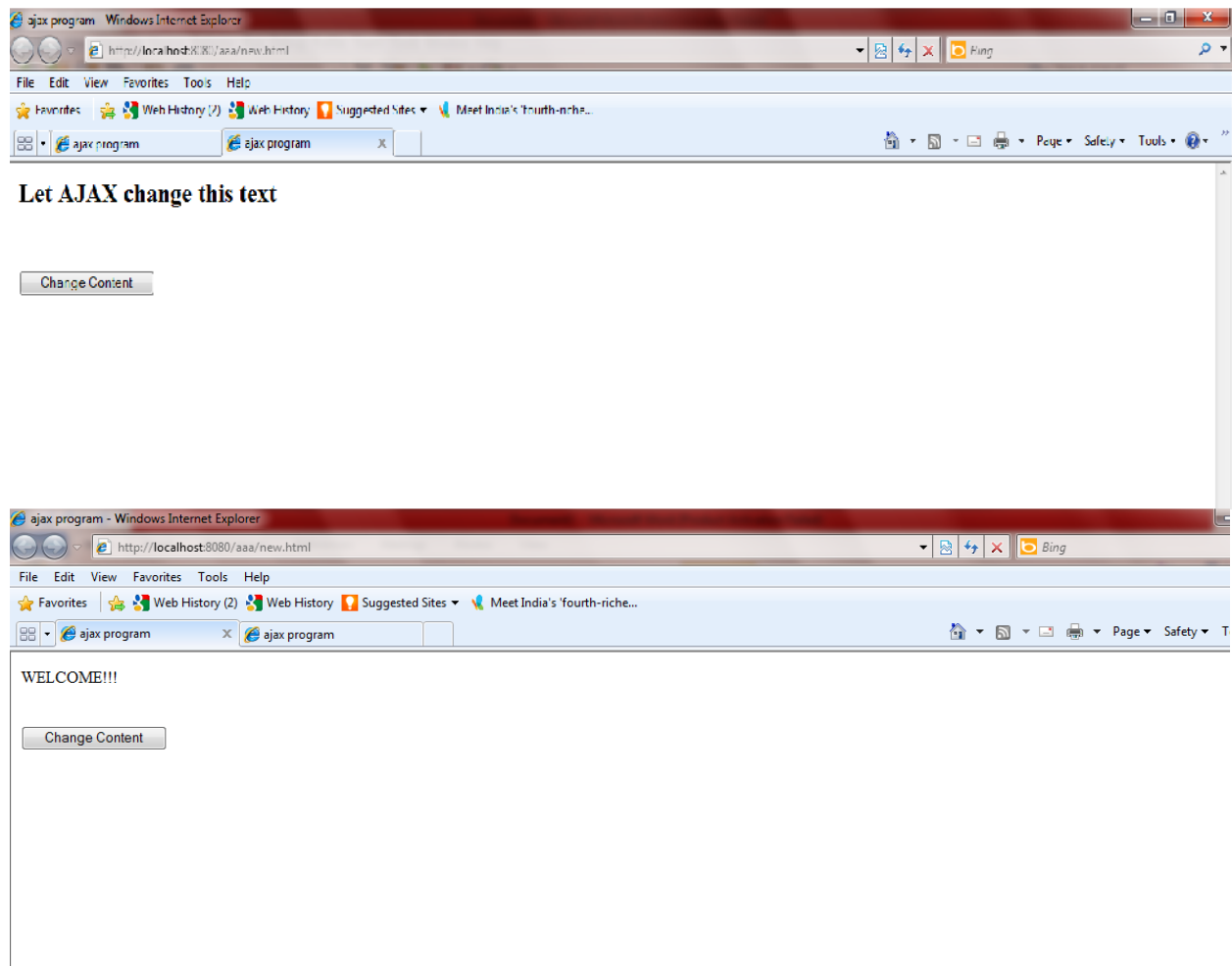
COURSE: CS8661 INTERNET PROGRAMMING LABORATORY

```
</script>
<title>ajax program</title>
</head>
<body>
<div id="myDiv"><h2>Let AJAX change this text</h2></div>
<button type="button" onclick="loadXMLDoc()">Change Content</button>
</body>
</html>
```

New.txt:

WELCOME!!!

OUTPUT:



RESULT:

Thus the program for AJAX was executed and the output was verified.

EX.NO: 2

DATE:

IMPLEMENTING AN APPLICATION USING THE WEB SERVICES

AIM:

To implement a application using the web services.

ALGORITHM:

- 1.Start the program
- 2.Create a root rocess for reservation
- 3.Create a service with focus on each item
- 4.Run the program, display the result
- 5.Stop the program.

SOURCE CODE:

```
<?xml version = "1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!-- Solution11.16 -->
<!-- Airline Reservation System-->
<html xmlns = "http://www.w3.org/1999/xhtml">
<head>
<title>Airline Reservation System</title>
<script type = "text/javascript">
<!--
var input;
var secondInput;
var element;
var secondElement;
var firstCount = 0;
var economyCount = 0;
var seats = [ ,0,0,0,0,0,0,0,0,0]; //allocate 10-element Array
function startArray()
{
for(var i=0; i<11; i++)
{
input = window.prompt("Please type 1 for First Class and Please type 2 for Economy.", "0");
if (input == 1 || input == 2)
{
element = linearSearch(seats);
if(element== -1 && input==1)
{
document.writeln("The First Class is already fully
booked<br/>");
secondQuestion(seats);
}
else if (element == -1 && input == 2)
```

```

{
document.writeln("The Economy Class is already fully
booked<br/>");
secondQuestion(seats);
}
else
boarding Pass(input);
}
//to terminate the program
else
{
window.status="Bye-bye!";
System.exit(0);
}
}
}
function linear Search(the Array)
{
if (input == 1)
{
for (var n=0; n<6 ; n++)
if (the Array [n] == 0)
return n;
}
else if (input == 2)
{
for (var n=6; n<11 ; n++)
if (the Array [n] == 0)
return n;
}
return -1;
}
function boarding Pass(the Input)
{
if (input ==1)
{
document.writeln("-----BOARDING PASS-----<br/>");
document.writeln("You are allocated in the First Class<br/>");
document.writeln("Your seat number is "+ element+"<br/>");
document.writeln("-----<br/>");
seats[element]= 1;
firstCount++;
}
else if (input ==2)

{
document.writeln("-----BOARDING PASS-----<br/>");
document.writeln("You are allocated in the EconomyClass<br/>"); document.writeln("Your

```

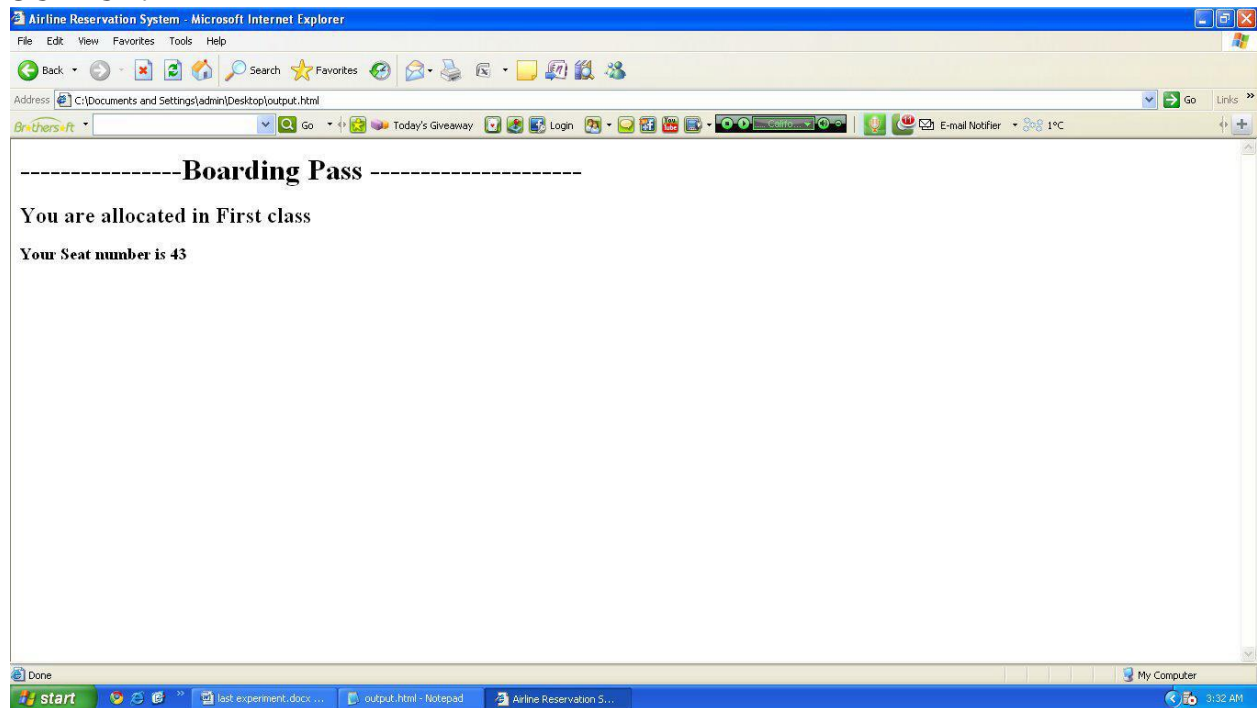
```

seat
number is "+ element + "<br/>";
document.writeln("-----<br/>");
seats[element]= 1;
economyCount++;
}
}
functionsecondQuestion(theArray)
{
if (input == 1)
{
for (var n=6; n<11 ;n++)
{
if (theArray [n] == 0)
{
second Input = window.prompt("Do you want to move to Economy Class?
(If YES, please press 1. If NO, please press 2)","0");
if ( second Input == 1)
{
input = 2;
element=linear Search(seats);
document.writeln("You have been allocated to Economy
Class<br/>");
boardingPass(input);
break;
}
else if (secondInput == 2)
{
document.writeln("Next flight leaves in 3 hours<br/>"); break;
}
}
}
}
else if (input == 2)
{
for (var n=0; n<6 ;n++)
{
if (theArray [n] == 0)
{
secondInput = window.prompt("Do you want to move to First Class? (If YES,
please press 1. If NO, please press 2)","0");
for (var n=0; n<6 ;n++)
{
if (theArray [n] == 0)
{
secondInput = window.prompt("Do you want to move to First Class? (If YES, please press 1. If
NO, please press 2)","0");

```

```
boarding Pass(input); break;
}
else if (secondInput == 2)
{
document.writeln("Next flight leaves in 3 hours<br/>");
break;
}
} }
}
}
//-->
</script>
</head>
<body onload = "startArray()"></body>
</html>
```

OUTPUT:



RESULT:

Thus the program is executed and verified successfully.