**IT8511 WEB TECHNOLOGY LABORATORY L T P C**

**0 0 4 2**

**OBJECTIVES:**

* To design interactive web pages using Scripting languages.
* To learn server side programming using servlets and JSP
* To develop web pages using XML/XSLT.

**LIST OF EXPERIMENTS**

1. Create a web page with the following using HTML.

i) To embed an image map in a web page.

ii) To fix the hot spots.

iii) Show all the related information when the hot spots are clicked

2. Create a web page with all types of Cascading style sheets.

3. Client Side Scripts for Validating Web Form Controls using DHTML.

4. Installation of Apache Tomcat web server.

5. Write programs in Java using Servlets:

To invoke servlets from HTML forms.

Session Tracking.

6. Write programs in Java to create three-tier applications using JSP and databases

• 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.

7. Programs using XML – Schema – XSLT/XSL.

8. Programs using DOM and SAX parsers.

9. Programs using AJAX.

10. 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.

**TOTAL: 60 PERIODS**

**OUTCOMES:**

**Upon Completion of the course, the students will be able to:**

• Design simple web pages using markup languages like HTML and XHTML.

• Create dynamic web pages using DHTML and java script that is easy to navigate and use.

• Program server side web pages that have to process request from client side web pages.

• Represent web data using XML and develop web pages using JSP.

• Understand various web services and how these web services interact.

**LAB REQUIREMENTS FOR A BATCH OF 30 STUDENTS**

**SOFTWARE REQUIRED:**

• Dream Weaver or Equivalent, MySQL or Equivalent, Apache Server, WAMP/XAMPP

**HARDWARE REQUIRED:**

* Standalone dsktops 30 Nos

**Ex. No.:1 CREATING A WEBPAGE USING IMAGE MAP**

**Aim:** To create a web page with the following using HTML.

i) To embed an image map in a web page.

ii) To fix the hot spots.

iii) Show all the related information when the hot spots are clicked

**Algorithm:**

**Step 1:** Create a html file with map tag.

**Step 2:** Set the source attribute of the img tag to the location of the image and also set the use

map attribute.

**Step 3:** Specify an area with name, shape and href set to the appropriate values.

**Step 4:** Repeat step 3 as many hot spots you want to put in the map.

**Step 5:** Create html files for each and every hot spots the user will select.

**Program:**

**India.html**

<html>

<head><title>IMAGE MAP USING HTML</title>

<body bgcolor="blue">

<center><font face=" Times New Roman" color="yellow"><h1>India</h1></font></center>

<center><**img** src="india.png" **usemap**="#India"></center>

<**map** name="India">

<**area shape**="rect" **coords**="278,671,310,718" **href**="Chennai.html" **title**="CHENNAI">

<**area shape**="circle" **coords**="141,429,25" **href**="Mumbai.html" **title**="MUMBAI">

<**area shape**="rect" **coords**="157,547,199,566" **href**="Goa.html" **title**="GOA">

<**/map**>

</body>

</html>

**Chennai.html**

<html>

<head>

<title>CHENNAI</title>

</head>

<body bgcolor="yellow">

<center><font face=" Algerian" size="12" color="red">CHENNAI</font></center><br>

<marquee direction=”right”> Vanakam Chennai </marquee>

welcome to chennai

</body>

</html>

**Mumbai.html**

<html>

<head>

<title>MUMBAI</title>

</head>

<body bgcolor="pink">

<center><font face="Magneto Bold" size="12" color="orange">MUMBAI</font></center><br>

<marquee>namsathy</marquee>

sorry for the inconvenience mumbai official site under maintainence

</body>

</html>

**Goa.html**

<html>

<head>

<title>GOA</title>

</head>

<body bgcolor="red">

<center><font face="Magneto Bold" size="12" color="green">GOA</font></center><br>

<marquee>tourist spot</marquee>

<ul type="disc">

<li>arabic sea</li>

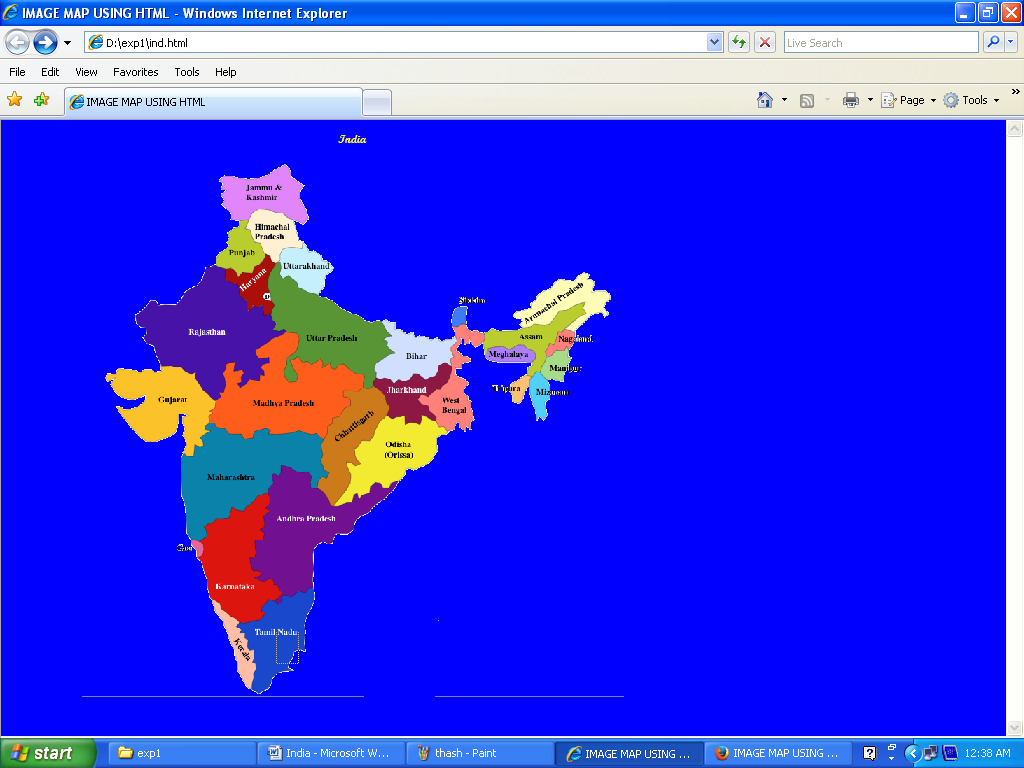
<li>church</li>

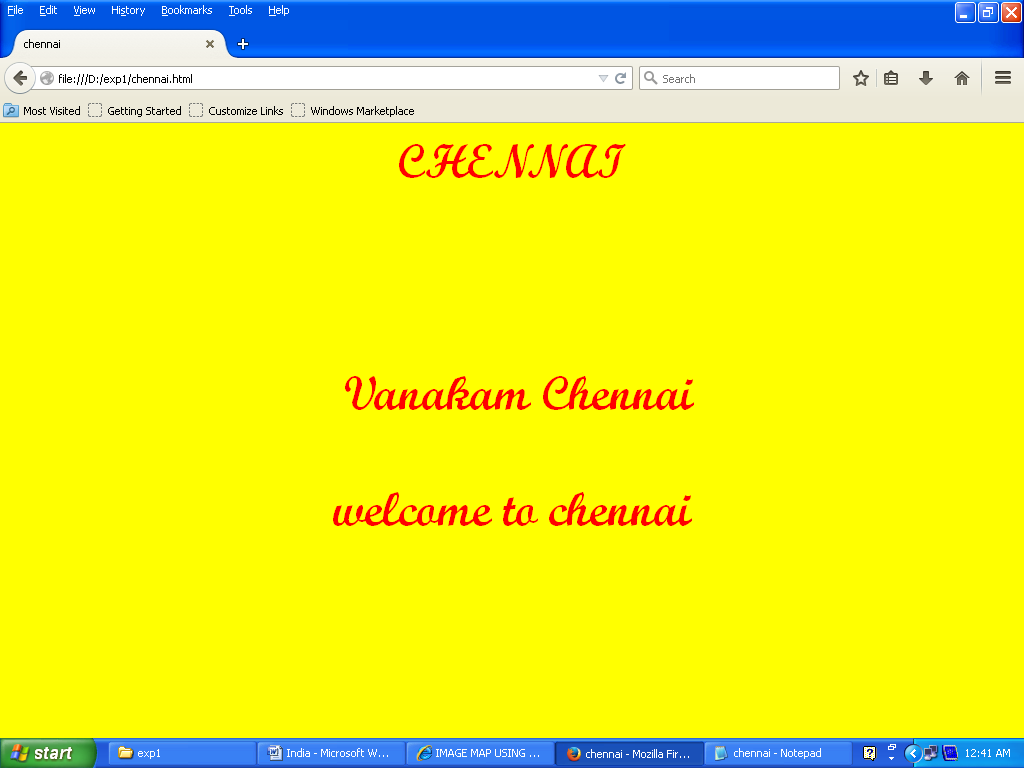
</ul>

</body>

</html>

**Output**





**Ex. No.:2 CREATING A WEBPAGE WITH CASCADING STYLE SHEETS**

**Aim:**

To create a web page with all types of cascading style sheets

**Algorithm:**

**Step 1:** Declare as individual element by using style attribute (inline style)

**Step 2:** Embed the entire style rules in an HTML document head section (embedded style)by

using <style> element follows <style type=”text/css”>

**Step 3:** Create separate document that contains only css rule(external style)save as

filename.css

**Step 4:** Create a link between external style rule and HTML document by using <link>element

as follows<link rel=”stylesheet” type=”text/css” href=”extstyle.css”>

**Step 5:** Open the browser and display the webpage.

**Program:**

**main page.html**

<html>

<head>

<title>COLLEGE</title>

<style type="text/css">

ul{list-style-type:square}

body{

background-image: url("tree.jpg");

background-repeat:repeat-x;

background-position:bottom

}

p{font-family:Papyrus;font-size:20px;color:green}

h1{font-family:Script MT Bold;font-size:25px}

</style>

</head>

<body>

<p><marquee direction="right"><center>CHENNAI INSTITUTE OF TECHNOLOGY</center></marquee></p>

<center><p>DEPARTMENT</p></center>

<ul>

<li><h1><a href=it.html>IT</a></h1></li>

<li><h1><a href=cse.html>CSE</a></h1></li>

<li><h1><a href=ece.html>ECE</a></h1></li>

<li><h1><a href=mech.html>MECH</a></h1></li>

</ul>

</body>

</html>

**It.html**

</body>

</html>

<html>

<head>

<title>INFORMATION TECHNOLOGY</title>

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

</head>

<body>

<h1>IT is interesting department with lot of interesting subjects</h1>

</body>

</html>

**Style.css**

h1{font-family:Script MT Bold;font-size:50px;color:red}

body{background-color:green}

**cse.html**

<head>

<title>CSE</title>

<style type="text/css">

p{font-family:Times New Roman;font-size:55px;text-transform:uppercase;font-weight:bold;color:sky blue}

h1{font-family:Old English Text MT;font-weight:bold}

body{background-color:yellow}

table,th,td{border:1px solid black}

th{color:green;background-color:red}

</style>

</head>

<body>

<marquee><p>Computer science engineering</p></marquee>

<table>

<tr>

<th>SEMESTER</th>

<th>SUBJECT</th>

</tr>

<tr>

<td>1</td>

<td>PYTHON PROGRAMMING</td>

</tr>

<tr>

<td>2</td>

<td>OOPS</td>

</tr>

<tr>

<td>3</td>

<td>JAVA PROGRAMMING</td>

</tr>

</body>

</html>

**Ece.html**

<html>

<head>

<title>ECE</title>

</head>

<body bgcolor="sky blue">

<center>

<p style="font-family:Pristina;text-transform:uppercase;FONT-SIZE:30">ECE<br><br>

<marquee direction="down">It is also a department </marquee></p></center>

</body>

</html>

**Mech.html**

<html>

<head>

<title>mech</title><link rel="stylesheet" type="text/css" href="style.css">

<style type="text/css">

p{font-family:Palace Script MT;font-size:55px;text-transform:uppercase;font-weight:bold;color:blue}

body{background-color:grey}

</style>

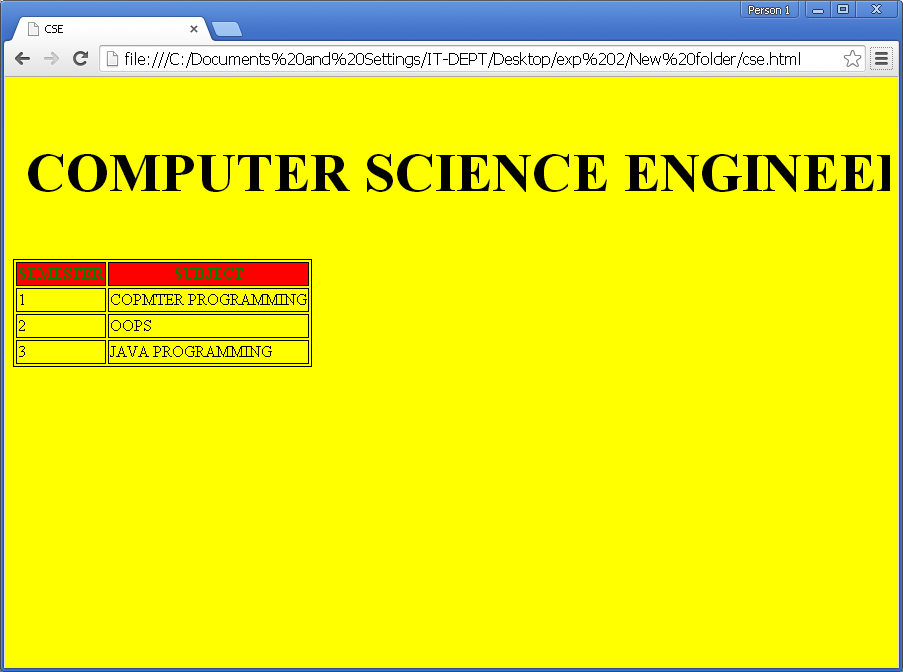
</head>

<body>

<marquee><h1>MECANICAL ENGINEERING</h1></marquee>

<p>only boys</p><h2 style=background-color:red;font-family:Pristina;font-size:30px;text-transform:uppercase>awesome department</h2></body></html>

**Output**

****

**Ex. No.:3 WEB FORM VALIDATION USING DHTML**

**Aim:**

To develop a client side scripts for validating web form controls using DHTML.

**Algorithm:**

1. Start the program

2. Insert <form> element in the HTML document with following attributes:

**name** attribute:to identify the form element

**action** attribute : specify the target of the forma data to be submitted

**onsubmit**: to call the javascript to valid the form when submit button s pressed.

3.Insert all the necessary form elements such as(<fieldset>,<legend>,<input>,<select>,<option>)

4.Insert various types of field such as(text,radio,checkbox,submit)by using “type” attribute of <input>element

5. Embed validate form() function within the head section of HTML document using <scipt>element.

6.Validate the input data given and alert the user if the field is blank by using alert()

7. Stop the program.

**Program:**

**Form.html**

<html>

<head>

<title>Form validation</title>

<script type="text/javascript">

function validateForm()

{

var x=document.myForm.fname.value

if(x==null || x=="")

{

alert("first name must be filled out");

document.myForm.fname.focus();

return false;

}

var x=document.myForm.lname.value

if(x==null ||x =="")

{

alert("last name must be filled out");

document.myForm.lname.focus();

return false;

}

var z=document.myForm.age.value;

if(z<18||z>60)

{

alert("Please give age range between 18 and 60!");

document.myForm.age.focus();

return false;

}

if((document.myForm.gender[0].checked==false)&&(document.myForm.gender[1].checked==false))

{

alert("please choose your gender!");

return false;

}

var x=document.myForm.email.value

if(x==null || x=="")

{

alert("plz provide an email id");

document.myForm.email.focus();

return false;

}

if(!x.match(/^[\w]+@[\w]+\.[a-z|A-Z]/))

{

alert("u have entered an invalid email address");

document.myForm.email.focus();

return false;

}

var a=document.getElementById("cntry");

if(a.selectedIndex==0)

{

alert("select a country");

document.myForm.country.focus();

return false;

}

if((document.myForm.hobby[0].checked==false)&&

(document.myForm.hobby[1].checked==false)&&

(document.myForm.hobby[2].checked==false)&&

(document.myForm.hobby[3].checked==false)&&

(document.myForm.hobby[4].checked==false))

{

alert("please choose ur hobby!");

return false;

}

var x=document.myForm.phone.value

if(x=="")

{

alert("Phone number is required field!");

document.myForm.phone.focus();

return false;

}

if(x.length!=10)

{

alert("Oops not a 10 digit number");

document.myForm.phone.focus();

return false;

}

if(x.match(/^\d{10}/))

{

return true;

}

else

{

alert("Invalid character in ur phone no!!! please try again");

document.myForm.phone.focus();

return false;

}

}

</script>

</head>

<body style="background-color:pink">

<center><h1 style="color:green">FORM VALIDATION USING JAVASCRIPT</h1></center>

<form name="myForm" action="register.html" onsubmit="return validateForm();" method="post">

<fieldset>

<legend>Registration Form</legend>

First name:<input type="text" name="fname" placeholder="First Name"><br><br>

Last name:<input type="text" name="lname" placeholder="Last Name"><br><br>

Age<input type="text" name="age" placeholder="ur age"><br><br>

Gender:

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

<input type="radio" name="gender" value="male">male<br><br>

Email:<input type="text" name="email" placeholder="Ur Email-Id"><br><br>

Choose ur country:

<select id="cntry" name="country"><br><br>

<option value="0">Select</option>

<option value="1">India</option>

<option value="2">Canada</option>

<option value="3">London</option>

</select><br><br>

Hobby:

<input type="checkbox" name="hobby">fishing

<input type="checkbox" name="hobby">gardening

<input type="checkbox" name="hobby">stamp collection

<input type="checkbox" name="hobby">foreign currency collection

<input type="checkbox" name="hobby">reading books<br><br>

Phone no:<input type="text" name="phone" placeholder="Ur contact numbr"><br><br>

<input type="submit" value="submit">

</fieldset>

</form>

</body>

</html>

**register.html**

<html>

<head>

<title>Registration Successfull</title>

</head>

<body bgcolor="pink">

<h3><center>Registered Successfully.......!</center></h3>

</body>

</html>

**Output**



**Ex. No.:4 INSTALLATION OF APACHE TOMCAT WEB SERVER**

**Aim :**

Installation of Apache Tomcat web server.

**Procedure:**

## What is Apache Tomcat

It is an application server or web server or servlet container developed by the Apache Software Foundation (ASF) and released under the Apache License version 2. HTTP web servers provide an environment for Java code to run in. It includes tools for configuration and management, but can also be configured by editing XML configuration files. Most of the modern Java web frameworks are based on servlets and JavaServer Pages and can run on Apache Tomcat, for example Struts, JavaServer Faces, Spring, etcetera.

**Apache Tomcat7.0.XX new released**

 The Apache Tomcat team has released version 7.0.40 of Apache. They removed several fixes that stop Tomcat attempting to parse text, improved handling and reporting if a ConcurrentModificationException occurs while checking for memory leaks, etcetera.

**How to Install Tomcat 7**

 There are certain steps we must follow for configuring Apache Tomcat 7.

**Step 1**

**Download and Install Tomcat**

1. Go to <http://tomcat.apache.org/download-70.cgi> then go to the Binary Distribution/Core/ and download the "zip" package (for example "apache-tomcat-7.0.40.zip", about 8MB).
2. Now **unzip** the downloaded file into a directory of our choice. Don't unzip onto the dekstop (since its path is hard to locate). I suggest using "e:\myserver". Tomcat will be unzipped into the directory "e:\myserver\tomcat-7.0.40".

**Step 2**

 Check the installed directory to ensure it contains the following sub-directories:

* bin folder
* logs folder
* webapps folder
* work folder
* temp folder
* conf folder
* lib folder

**Step 3**

 Now, we need to create an Environment Variable JAVA\_HOME.

 We need to create an environment variable called "JAVA\_HOME" and set it to our JDK installed directory.

1. To create the JAVA\_HOME environment variable in Windows XP/Vista/7 we need to push the "Start" button then select "Control Panel" / "System" / "Advanced system settings".  Then switch to the "Advanced" tab and select "Environment Variables" / "System Variables" then select "New" (or "Edit" for modification). In "Variable Name", enter "JAVA\_HOME". In "Variable Value", enter your JDK installed directory (e.g., "c:\Program Files\Java\jdk1.7.0\_{xx}").
2. For ensuring that it is set correctly, we need to start a command shell (to refresh the environment) and issue:  
   set JAVA\_HOME  
   JAVA\_HOME=c:\Program Files\Java\jdk1.7.0\_{xx} <== Check that this is OUR JDK installed directory
3. Sometimes we need to set JRE\_HOME also. So for creating JRE\_HOME we need to use the same procedure. Push the "Start" buttonthen select "Control Panel" / "System" / "Advanced system settings".  Then switch to the "Advanced" tab and select "Environment Variables" / "System Variables" then select "New" (or "Edit" for modification). In "Variable Name", enter "JRE\_HOME". In "Variable Value", enter your JRE installed directory (e.g., "C:\Program Files\Java\jre7\").

**Step 4**

## Configure Tomcat Server

The configuration files of the Apache Tomcat Server are located in the "conf" sub-directory of our Tomcat installed directory, for example "E:\myserver\tomcat7.0.40\conf". There are 4 configuration XML files:

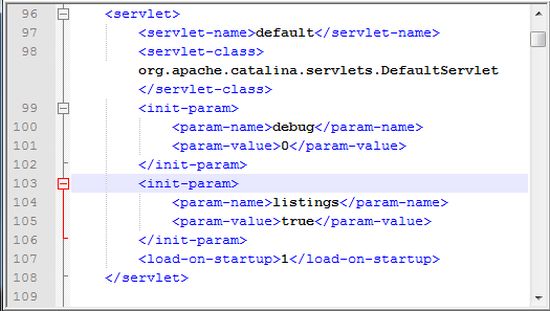
1. context.xml file
2. tomcat-users.xml file
3. server.xml file
4. web.xml file

Before proceeding, make a BACKUP of the configuration files.

**Step 4(a) "conf\web.xml"; Enabling a Directory Listing**

 Open the configuration file "web.xml". We shall enable the directory listing by changing "listings" from "false" to "true" for the "default" servlet.

 <param-value>**true**</param-value> like:



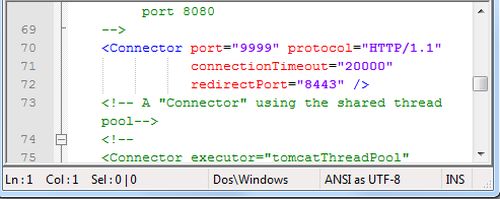
**Step 4(b) "conf\server.xml file"; set the TCP Port Number**

Open the file "server.xml" in a text editor.

 The default port number of Tomcat is 8080. Now we need to change the TCP port number for Tomcat, since the same port number can be used by other servers like SQL Server. We may choose any number between 1024 and 65535. We shall choose 9999 in this article.

Locate the following lines, and change port="8080" to port="9999". Like:

<Connector port="9999" protocol="HTTP/1.1" Like



**Step 4(c) "conf\context.xml"; Enabling Automatic Reload**

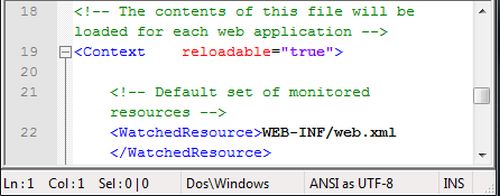
In that we set reloadable="true" to the <Context> element to enable automatic reload after code changes.

 Add reloadable="true" as in the following:

 <Context reloadable="true">

......

</Context> Like



**Step 4(d) (Optional) "conf\tomcat-users.xml"**

 It is used to manage Tomcat by adding the highlighted lines, inside the <tomcat-users> elements.

 In that we can add a password and username as an optional step.

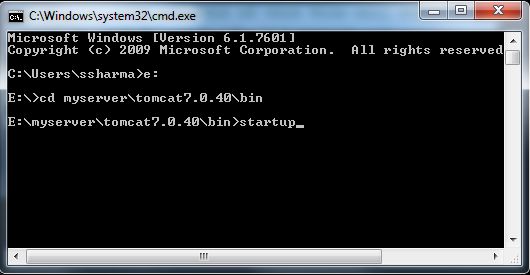
**Step 5**

 Now, start the tomcat server

 Executable programs and scripts are kept in the "bin" sub-directory of the Tomcat installed directory, e.g., "E:\myserver\tomcat7.0.40\bin".

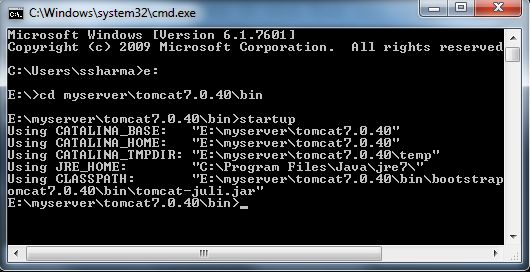
**Step 5(a) Start Server**

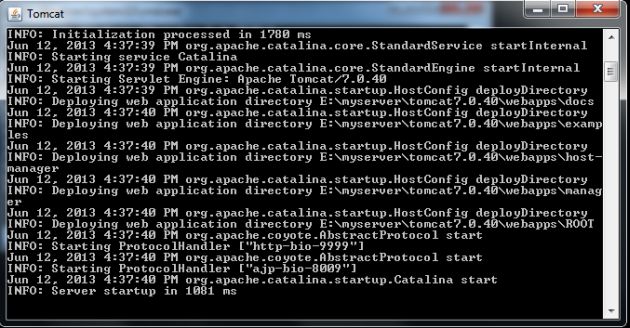
Launch a command shell. Set the current directory to "<TOMCAT\_HOME>\bin" like E:\myserver\tomcat7.0.40\bin, and run "startup.bat" as follows:



After that a new Tomcat console window appears. Read the messages on the console. Look out for the Tomcat's port number (double check that Tomcat is running on port 9999).......

We saw a figure like:

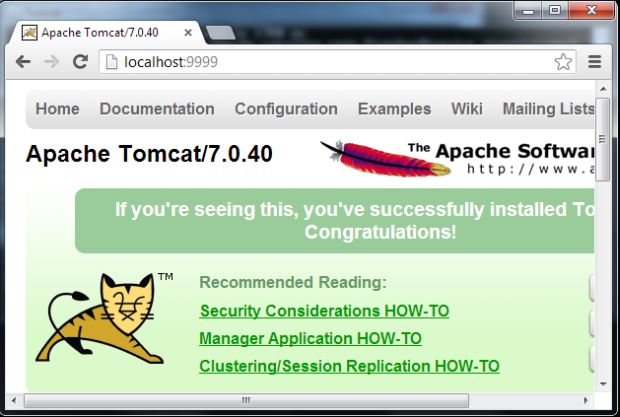




**Step 5(b) Access the Server**

Open a browser then enter the URL "http://localhost:9999" to access the Tomcat server's welcome page.

 If we get this type of page then it means we are done.



Now try the URL http://localhost:9999/examples to view JSP and servlet examples.

**Step 5(c) How to Shutdown Server**

We can stop the server using one of the following:

1. Press ctrl-c on the Tomcat console; or
2. Run "<TOMCAT\_HOME>\bin\shutdown.bat" script:  
   // Change the current directory to Tomcat's "bin"  
   > e: // Change the current drive  
   e:\> cd E:\myserver\tomcat7.0.40\bin // Change Directory to YOUR Tomcat's "bin" directory  
   // Shutdown the server  
   E:\myserver\tomcat7.0.40\bin> shutdown

**Ex. No.:5a INVOKE SERVLETS FROM HTML FORMS**

**Aim**

To write a program in Java using servlets to invoke servlets from HTML forms.

**Algorithm:**

Step 1.Create a index page , which accepts user name, phone number, gender, address and age as user input. Invoke the servlet page by submit button trigger.

Step 2.Create a servlet called **helloservlet** which accepts the values given from form and prints them.

Step 3.Save the appropriate files in the following tomcat hierarchy for depolyment

C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps\[project-root-directory]\index.html

C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps\[project-root-directory]\ helloservlet.java

Create two new folders namely WEB-INF within [project-root-directory] folder which has the web.xml and classes folder within WEB-INF which has .class of the java file

Step 4: Compile the servlet program and create servlet class file , save this class file

into (E:\Program Files (x86)\Apache Software Foundation\Tomcat 7.0\webapps\sample\WEB-INF\classes)classes folder.

Step5: Run the Tomcat server services, call the html file from the browser URL <http://localhost:8080/sample/index.html> and configure web.xml for url link and servlet name , class link

Step 6: Call the servlet program from html and display the result

Step 7: Stop the services

**Program**

**Index.html**

<html>

<body>

<form name=f1 action="http://localhost:8080/sample/helloservlet" method="GET">

USERNAME <input type="text" name="name"><br><br>

PHONE NO <input type="text" name="phno"><br><br>

SEX

<input type="radio" name="sex" value="F">female

<input type="radio" name="sex" value="M">male<br><br>

ADDRESS:<textarea rows="10" cols="10" name="address"></textarea><br><br>

AGE

<select name="age">

<option value="18" > 18 </option>

<option value="19" > 19 </option>

<option value="20" > 20 </option>

<option value="21" > 21 </option>

</select><br></br>

<input type="submit" value="submit">

<input type="button" value="clear">

</form>

</body>

</html>

**Helloservlet.java**

import java.io.\*;

import javax.servlet.\*;

import java.util.\*;

import javax.servlet.http.\*;

public class helloservlet extends HttpServlet

{

public void doGet(HttpServletRequest req,HttpServletResponse res) throws ServletException,IOException

{

res.setContentType("text/html");

PrintWriter pw =res.getWriter();

Enumeration e=req.getParameterNames();

while(e.hasMoreElements())

{

String name=(String)e.nextElement();

String value=req.getParameter(name);

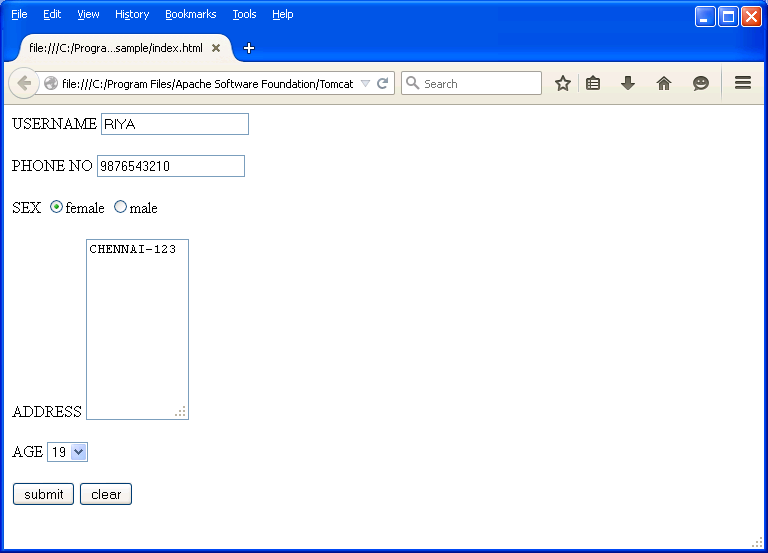
pw.println(name+"="+value+"<br>");

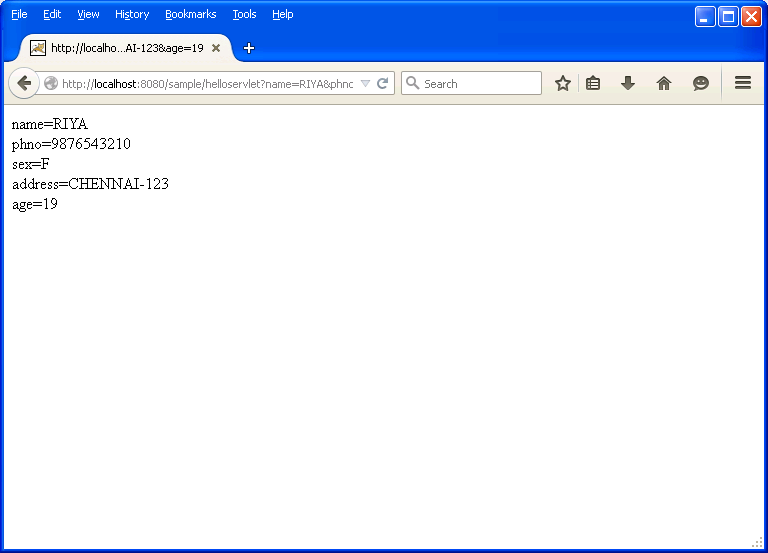
}

}

}

**Output**

****

****

**Ex. No.:5b SESSION TRACKING USING HIT COUNT**

**Aim**

To write a program in java to implement session tracking in servlets using hit count.

.

**Algorithm:**

Step 1: Write a servlet program with HttpSession object to trcack the session count

Step 2: Compile the servlet program and create servlet class file , save this class file

into(**E:\Program Files\Apache Software Foundation\Tomcat**

**7.0\webapps\sessioncount\WEB- INF\classes**) folder.

Step3: Run the Tomcat server services, call theservlet program from the browser

URL (**http://localhost:8080/sessioncount/SessionServlet**)

and configure web.xml for url link and servlet name , class link

Step 4: Call the servlet program from browser and check the staus of session count

Step 5: Every time visit by client will get increment the session count.

Step 6: Stop the Services.

**Program**

**SERVLET CODE: SessionServlet.java**

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

public class SessionServlet extends HttpServlet

{

protected void processRequest(HttpServletRequest request, HttpServletResponse response)throws ServletException, IOException

{

response.setContentType("text/html;charset=UTF-8");

HttpSession sess=request.getSession();

PrintWriter out = response.getWriter();

String head;

Integer cnt=(Integer)sess.getAttribute("cnt");

if (cnt==null)

{

cnt=new Integer(0);

head="Welcome You are Accessing the WebPage for the First Time";

}

else

{

head="Welcome Once again";

cnt=new Integer(cnt.intValue()+1);

}

sess.setAttribute("cnt",cnt);

try

{

out.println("<html>");

out.println("<head>");

out.println("<title>Servlet SessionServlet</title>");

out.println("</head>");

out.println("<body bgcolor=#FFFF99><br><br><center>");

out.println("<h1>"+head+"</h1>");

out.println("<h2>The number of Previous Access="+cnt+"</h2>");

out.println("</center></body>");

out.println("</html>");

}

finally

{

out.close();

}

}

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)throws ServletException, IOException

{

processRequest(request, response);

}

}

**web.xml**

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

<web-app>

<servlet>

<servlet-name>SessionServlet</servlet-name>

<servlet-class>SessionServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>SessionServlet</servlet-name>

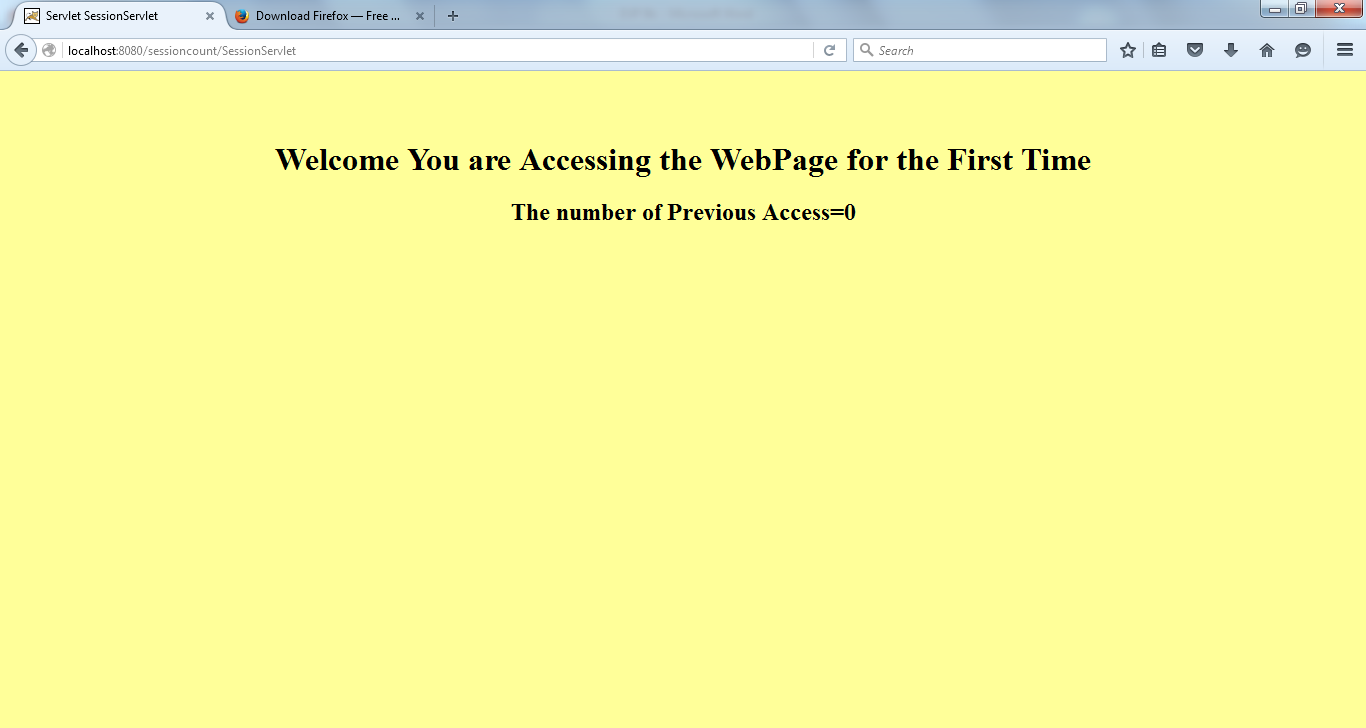
<url-pattern>/SessionServlet</url-pattern>

</servlet-mapping>

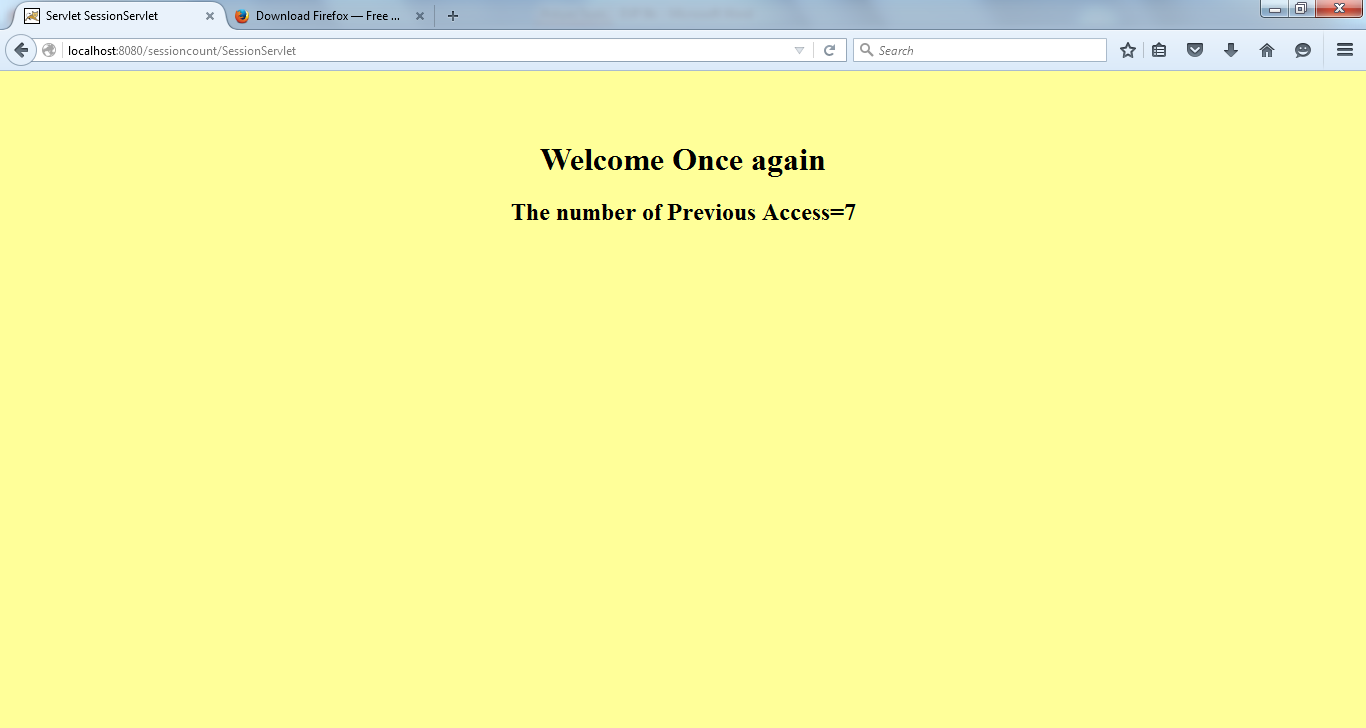
</web-app>

**OUTPUT**

Session Servlet Fist time run



Seesion Servlet Multiple time access



**Ex. No.:6a ONLINE EXAMINATION USING JSP AND DATABASE**

**Aim**

. To write programs in Java to create three-tier applications using JSP and Databases

for conducting on-line examination.

**Algorithm:**

**Client:**

1. In client side design the contents that you like to transfer to the server using html form with input type tags.
2. In the HTML program the action parameter in the form specifies the

name of the jsp which is to be invoked.

1. The method specified is GET. That means a GET request is made by HTML to the jsp

**Server:**

1. import java.io.\* package which provides the interface for performing simple input and output programs.
2. import java.sql.\* package provides the classes and interfaces for the operation on databases.
3. get the name and seatno from the request and calculate the total for the answers checked by the user in the form.
4. Load the DriverManager class using the following syntax Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
5. Make the DriverManager to connect to the DSN by using the following syntax con=DriverManager.getConnection("jdbc:odbc:dsn");
6. Insert the Seat\_no, Name, Marks of the particular user in the database using the object of Statement class with insert query.
7. Retrieve the records from the database using the object of ResultSet query.
8. Finally close connection to the database by closing the object of connection class and statement class, ResultSet class.

**Program:**

**ExamServer.jsp**

<%@page contentType="text/html" language="java" import="java.sql.\*"%>

<html>

<head>

<title>Online Exam Server</title>

<style type="text/css">

body{background-color:black;font-family:courier;color:blue}

</style>

</head>

<body>

<h2 style="text-align:center">ONLINE EXAMINATION</h2>

<p>

<a href="ExamClient.html">Back To Main Page</a>

</p>

<hr/>

<%

String str1=request.getParameter("ans1");

String str2=request.getParameter("ans2");

String str3=request.getParameter("ans3");

int mark=0;

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Connection con=DriverManager.getConnection("jdbc:odbc:examDS");

Statement stmt=con.createStatement();

ResultSet rs=stmt.executeQuery("SELECT \* FROM examTab");

int i=1;

while(rs.next())

{

if(i==1)

{

String dbans1=rs.getString(1);

if(str1.equals(dbans1))

{

mark=mark+5;

}

}

if(i==2)

{

String dbans2=rs.getString(1);

if(str2.equals(dbans2))

{

mark=mark+5;

}

}

if(i==3)

{

String dbans3=rs.getString(1);

if(str3.equals(dbans3))

{

mark=mark+5;

}

}

i++;

}

if(mark>=10)

{

out.println("<h4>Your Mark Is : "+mark+"</h4>");

out.println("<h3>Congratulations....! You Are Eligible For The Next Round...</h3>");

}

else

{

out.println("<h4>Your Mark is : "+mark+"</h4>");

out.println("<h3>Sorry....!! You Are Not Eligible For The Next Round...</h3>");

}

%>

</form>

</body>

</html>

**ExamClient.html**

<html>

<head>

<title>Online Exam Client</title>

<style type="text/css">

body{background-color:black;font-family:courier;color:blue}

</style>

</head>

<body>

<h2 style="text-align:center">ONLINE EXAMINATION</h2>

<h3>Answer the following questions (5 marks for each correct answer)</h3>

<hr/>

<form name="examForm" method="post" action="ExamServer.jsp">

1. All computers must have <br/>

<input type="radio" name="ans1" value="Operating System">Operating System

<input type="radio" name="ans1" value="Application Software">Application Software

<input type="radio" name="ans1" value="CD Drive">CD Drive

<input type="radio" name="ans1" value="Microsoft word">Microsoft word

<br/><br/>

2. The term PC means <br/>

<input type="radio" name="ans2" value="Private Computer">Private Computer

<input type="radio" name="ans2" value="Professional Computer">Professional Computer

<input type="radio" name="ans2" value="Personal Computer">Personal Computer

<input type="radio" name="ans2" value="Personal Calculator">Personal Calculator

<br/><br/>

3.C was developed by?<br/>

<input type="radio" name="ans3" value="Dennis Ritchie">Dennis Ritchie

<input type="radio" name="ans3" value="Stroustrup">Stroustrup

<input type="radio" name="ans3" value="David Ritchie">David Ritchie

<input type="radio" name="ans3" value="Charles Babbage">Charles Babbage

<br/><br/>

<input type="submit" value="Check Your Result"/>

</form>

</body>

</html>

**Output**



**Ex. No.: 6b STUDENT INFORMATION SYSTEM USING JSP AND DATABASE**

**Aim**

To write jsp program to retrieve and display student marklist available in a database.

**Algorithm:**

**Client:**

1. In client side design the contents that you like to transfer to the server

using html form with input type tags.

1. In the HTML program the action parameter in the form specifies the name of the jsp which is to be invoked.
2. The method specified is GET. That means a GET request is made by HTML to the jsp

**Server:**

1. import java.io.\* package which provides the interface for performing simple input and output programs.
2. import java.sql.\* package provides the classes and interfaces for the operation on databases.
3. get the parameters from the request
4. Load the DriverManager class using the following syntax Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
5. Make the DriverManager to connect to the DSN by using the following syntax con=DriverManager.getConnection("jdbc:odbc:dsn");
6. if the parameter from the request matches with database column then Retrieve the records of the particular user from the database using the object of ResultSet query.

**Program:**

**self.jsp**

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

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

<%@ page import="java.util.\*" %>

<% Class.forName("sun.jdbc.odbc.JdbcOdbcDriver"); %>

<html>

<head>

<title>Fetching Data From a Database</title>

</head>

<body bgcolor="yellow">

<form action="self.jsp" method="POST">

<h1 style="color:red">Please enter the register no of student u want to find:</h1>

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

<input type="submit" value="Submit">

</form>

<%

Connection connection = DriverManager.getConnection("jdbc:odbc:data", " ", " ");

Statement statement = connection.createStatement();

String id = request.getParameter("id");

ResultSet resultset = statement.executeQuery("select \* from stud where Regno =" + id + " ") ; if(!resultset.next())

{

out.println("<h3>"+"Sorry,could not find that student detail!!!!Please try again"+"</h3>");

}

else { %>

<h3 style="color:green">Fetching student detail from a Database </h3>

<table border="1">

<tr style="color:blue">

<th>Regno</th>

<th>Name</th>

<th>Department</th>

<th>Mark1</th>

<th>Mark2</th>

<th>Mark3</th>

</tr>

<tr>

<td> <%= resultset.getString(1) %> </td>

<td> <%= resultset.getString(2) %> </td>

<td> <%= resultset.getString(3) %> </td>

<td> <%= resultset.getString(4) %> </td>

<td> <%= resultset.getString(5) %> </td>

<td> <%= resultset.getString(6) %> </td>

</tr>

</table><br>

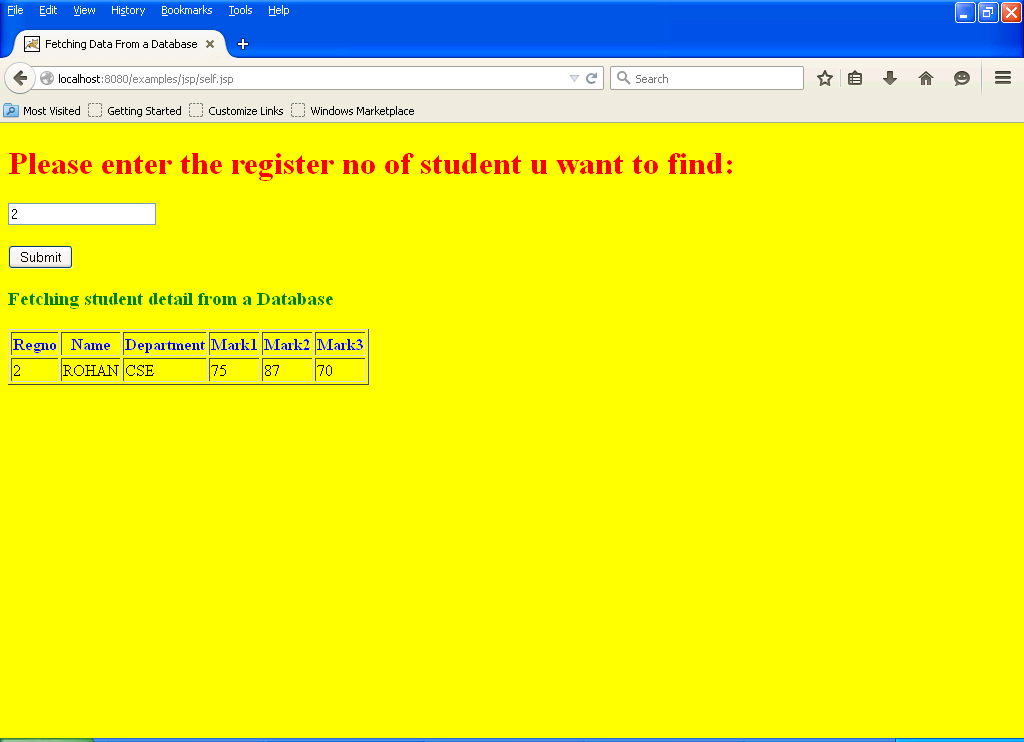
<%

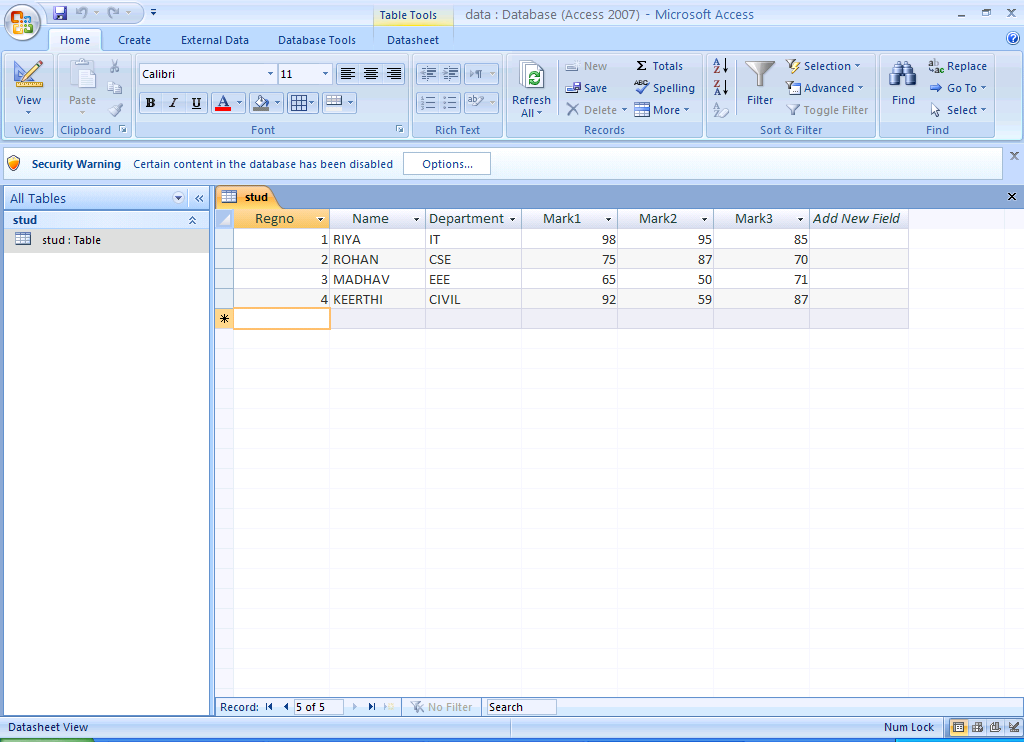
}

%>

</body> </html>

**Output**





**Ex. No.: 7 CREATING WEB PAGES USING XML DOCUMENT**

**Aim**

To write a program using XML – Schema –XSLT/XSL for displaying the contents available in XML document

**Algorithm:**

1. Create an XML document.
2. When developing XML code, include the following line as the first line in the XML file:

<?xml version="1.0" encoding="ISO-8859-1"?>  
<?xml-stylesheet type="text/xsl" href="student.xsl"?>

1. Enclose the coding for this XML file within the

<student></student>tag.

4. Save the XML file using .xml extension.

5. Now create a XSLT file and develop the code with reference to the XML

file student.xml such as

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

<xsl:stylesheet version="1.0"

xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

6.The <xsl : for-each > element can be used to select every XML element of a specified node-set.

<xsl : for-each select=”Student/Person-Details” > Hence under the root node Student there are the nodes named Person-Details which get selected each time.

7.The <xsl : value-of> element can be used to extract the value of the desired elementThe name of the element must be assigned to the select clause.

8. Save the XSLT file using .xsl extension. The XSLT transformation is

done by the browser, when the browser reads the XML file.

**Program**

**Student.xml**

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

<?xml-stylesheet type="text/xsl" href="student.xsl"?>

<student>

<Person-Details>

<name>Aarthi</name>

<address>chennai</address>

<std>second</std>

<marks>70 percent</marks>

</Person-Details>

<Person-Details>

<name>priyanka</name>

<address>andhra</address>

<std>second</std>

<marks>80 percent</marks>

</Person-Details>

<Person-Details>

<name>Amni</name>

<address>mumbai</address>

<std>forth</std>

<marks>90 percent</marks>

</Person-Details>

<Person-Details>

<name>Tinu</name>

<address>delhi</address>

<std>tenth</std>

<marks>75 percent</marks>

</Person-Details>

</student>

**Student.xsl**

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

<xsl:stylesheet version="1.0"

xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">

<html>

<body>

<h2>Student database</h2>

<table border="1">

<tr bgcolor="gray">

<th>name</th>

<th>address</th>

<th>standard</th>

<th>marks</th>

</tr>

<xsl:for-each select="student/Person-Details">

<xsl:sort select="marks"/>

<tr bgcolor="pink">

<td><xsl:value-of select="name"/> </td>

<td><xsl:value-of select="address"/></td>

<td><xsl:value-of select="std"/> </td>

<td><xsl:value-of select="marks"/> </td>

</tr>

</xsl:for-each>

</table>

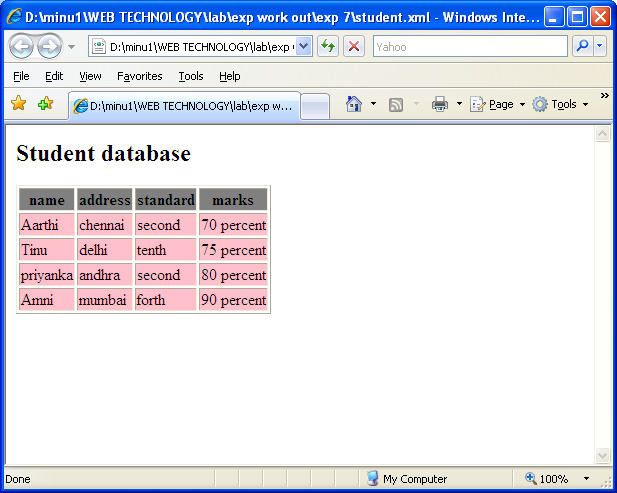
</body>

</html>

</xsl:template>

</xsl:stylesheet>

**Output**

****

**Ex. No.: 8a PROGRAM USING DOM PARSER**

**Aim**

To write a XML document for checking well formedness of XML document using DOM API

**Algorithm:**

1. Start the program

2. Import **java.io** package for performing simple input and outout operations,**javax.xml.parses.\*** package for

processing the XML documents, **org.w3c.dom\*;** package provides the interface for DOM,

**org.xml.sax.\*;**provides classes and interface for API for XML(SAX)

3. Read the name of XML document using **BufferedReader** class.

4**. Document BuilderFactory** is a factory API an application can obtain parser.

5. **Document Builder** is used to invoke a method parse.

6.In the try block we are calling the method **parse** for parsing the XML document. If the XML document is not well formed then the control of the program will go to catch block.

6. Stop the program.

**PROGRAM:**

**DOM.java**

import java.io.\*;

import java.xml.parsers.\*;

import org.w3c.dom\*;

import org.xml.sax.\*;

public class DOM

{

Public static void main(String[] args)throws IOException

{

try

{

System.out.println(“Enter the name of XML document”);

BufferedReader input=new BufferedReader(new InputStreamReader(System.in));

String file\_name=input.readLine();

File fp=new file(file\_name);

if(fp.exists())

{

try

{

Document BuilderFactory obj= Document BuilderFactory.newInstance();

Document Builder builder=obj.new Document Builder();

InputSource src=new InputSource(file\_name);

Document doc=builder.parse(src);

System.out.println(file\_name+”is well-formed”);

}

Catch(Exception e)

{

System.out.println(file\_name+”is not well-formed”);

System.exit(1);

}

}

else

{

System.out.println(“File not found”);

}

}

Catch(IOException e)

{

e.printStackTrace();

}

}

}

**Student.xml**

<?xml version=”1.0”?>

<student>

<Rollno>3</Rollno>

<personal\_info>

<name>Rohan</name>

<addr>Delhi</addr>

</personal\_info>

<sub>English</sub>

<mark>96

</student>

**OUTPUT:**

|  |
| --- |
| javac DOM.java  java DOM  Enter the name of XML document  Student.xml  Student.xml is not a well-formed document |

**Ex. No.: 8b PROGRAM USING SAX PARSER**

**Aim**

To write a XML document for checking well formedness of XML document using SAX API

**Algorithm:**

1. Start the program

2. Import **java.io** package for performing simple input and outout operations,

**org.xml.sax.\*;**provides classes and interface for API for XML(SAX), **org.xml.sax.helpers.\*;**

provides the helper classes for simple API for XML

3. Read the name of XML document using **BufferedReader** class.

4**. XMLReader Factory** helps in creating an XML reader,this reader parses xml document..

5. **createXMLReader** an object **reader** is created using which we can calla method parse.

6.In the try block we are calling the method **parse** for parsing the XML document. If the XML document is not well formed then the control of the program will go to catch block.

6. Stop the program.

**PROGRAM:**

**SAX.java**

import java.io.\*;

import org.xml.sax.\*;

import org.xml.sax.helpers.\*;

public class SAX

{

Public static void main(String[] args)throws IOException

{

try

{

System.out.println(“Enter the name of XML document”);

BufferedReader input=new BufferedReader(new InputStreamReader(System.in));

String file\_name=input.readLine();

File fp=new file(file\_name);

if(fp.exists())

{

try

{

XMLReader reader= XMLReader Factory.createXMLReader();

Reader.parse(file\_name);

System.out.println(file\_name+”is well-formed”);

}

Catch(Exception e)

{

System.out.println(file\_name+”is not well-formed”);

System.exit(1);

}

}

else

{

System.out.println(“File not found”);

}

}

Catch(IOException e)

{

e.printStackTrace();

}

}

}

**Student.xml**

<?xml version=”1.0”?>

<student>

<Rollno>3</Rollno>

<name>Rohan</name>

<addr>Delhi</addr>

</student>

**OUTPUT:**

|  |
| --- |
| javac SAX.java  java SAX  Enter the name of XML document  Student.xml  Student.xml is a well-formed document |

**Ex. No.: 9 PROGRAM USING AJAX**

**Aim**

To write an AJAX program to retrieve the contents of text file.

**Algorithm:**

1. 1 A client event is created.
2. Use div section to display information returned from a server.
3. Create button to call a function named loadDoc(), if it is clicked.
4. Add a <script> tag to the page's head section containing the loadDoc() function.
5. Use XMLHttpRequest object to exchange data with a server behind the scenes.
6. Use the open() and send() methods of the XMLHttpRequest object to send a request to a server.
7. Click the button to view the changes in the browser.

**Program**

**Ajax.html**

<html>

<head>

<script type=”text/javascript”>

function loadDoc()

{

var req;

if (window.XMLHttpRequest)

{

req=new XMLHttpRequest();

}

else

{

req=new ActiveXObject("Microsoft.XMLHTTP");

}

req.onreadystatechange=function()

{

if (req.readyState==4 && req.status==200)

{

document.getElementById("myDiv").innerHTML=req.responseText;

}

}

req.open("GET","ajaxinfo.txt",true);

req.send();

}

</script>

</head>

<body>

<div id="myDiv"><h2>Let AJAX change this text</h2></div>

<input type="button" onclick="loadDoc()" value="Change Content">

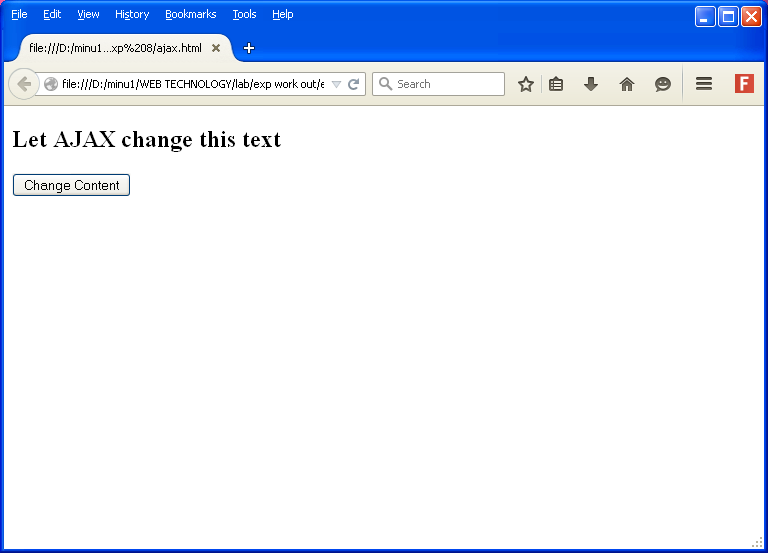
</body>

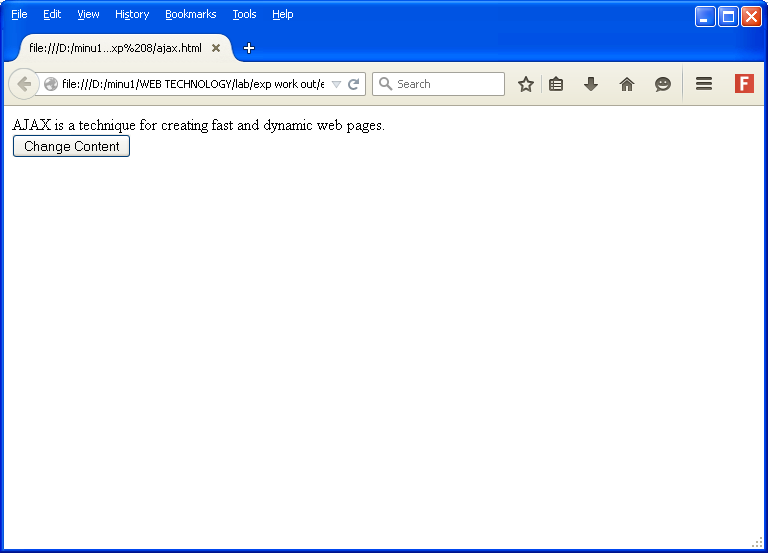
</html>

**Ajaxinfo.txt**

AJAX is a technique for creating fast and dynamic web pages.

**Output**

****

****

**Ex. No.: 10 PROJECT WORK**

**ONLINE FLIGHT TICKET RESERVATION**

**INTRODUCTION:**

This is a web application used for booking flight tickets online.  It adopts the three-tier architecture. This application is used by the travel agency to fetch suitable flights for their customers and helping them book online.

**PROJECT DESCRIPTION:**

In this web application, the front end application is used by the Customer to provide the travel details to the travel agency, and the agency in turn searches for the suitable flights available from the databases of different airlines for the customer available at the back-end.  When found, the agency provides the options (if any), and the customer selects from them. Then the customer is provided the cost of travel for the selected flight. He then makes payment using the credit card, where the transactions are recorded in a separate database at the back-end.  Then the agency makes the payment to the respective airlines booking the tickets for the customer.

**MINI PROJECT: ONLINE FLIGHT TICKET RESERVATION**

**Reservationform.html:** This is a HTML file which generates theflight reservation form, from the travel agency.

**User.java:**  This is a Java Servlet application that interacts with the database to provide customer with the suitable flight option.  And the customer selects an option that is appropriate to his travel plan.

**Payment.java:** This is also a Java Servlet application that interacts with the database and provides the cost of travel along with the flight details that has been selected by the customer.  And asks for customer confirmation.

**Payment.html:** This is a HTML page, which gets the payment details from the user like his name, address, credit card number, etc. and makes the appropriate transactions and updates the PAYMENT database in the mean while recording the transaction made.

**Passenger.java:** This is a Java Servlet page that confirms the transaction made by the customer and provides the link for the ticket.

**Ticket.java:** It is a Java Servlet page which generates the ticket to the user.

**Rformvalidation.js:** This is a JavaScript file that is used to perform the validation for the reservationform.html page.

**Ticket.css:** This is a Cascading StyleSheet file that is used by Reservationform.html and Payment.html files to enhance their presentation.

**FLIGHTINFO – TABLE:**  This table maintains the flight available from different airlines.

**PAYMENT – TABLE:**   This table maintains the customer transaction to the travel agency.

**AIRLINE\_PAYMENT – TABLE:** This table maintains the transactions made between the airlines and travel agency.

**PROGRAM:**

**Reservation form.html**

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>

<head>

<title>Bon Voyage Travel Agency</title>

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

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

<script type="text/javascript" src="rform validation.js">

</script>

</head>

<body><br> <br> <br>

<center> <h1 class ="f1">  Bon Voyage Travel Agency </h1> </center><br> <br> <br>

<center> <h1 class="fp1"> <u class="fp3"> Flight Ticket Booking </u> </h1> </center><br><br>

<form name="userapp" action="http://localhost:8084/Flight\_Eticketing/user" onsubmit="return validateForm();">

<table border="0" cellpadding="1" class="fp1" align="center">

<caption> <u> <b> Select your Travel points </b> </u> </caption>

<tbody><tr>

<td> Source point:</td>

<td><select name="src">

<option value="1"> Chennai </option>

<option value="2"> Bangalore </option>

<option value="3"> Hyderabad </option>

<option value="4"> Goa </option>

<option value="5"> Kolkatta </option>

<option value="6"> Delhi </option>

<option value="7"> Mumbai </option>

</select></td></tr>

<tr>

<td> Destination point:</td>

<td ><select name="dest">

<option value="1"> Chennai </option>

<option value="2"> Bangalore </option>

<option value="3"> Hyderabad </option>

<option value="4"> Goa </option>

<option value="5"> Kolkatta </option>

<option value="6"> Delhi </option>

<option value="7"> Mumbai </option>

</select></td>

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

<center class="fp1">

<u> <b> Travel Date </b> </u><br>

<select name="sdate">

<option value="1"> 1 </option>

<option value="2"> 2 </option>

<option value="3"> 3 </option>

<option value="4"> 4 </option>

<option value="5"> 5 </option>

<option value="6"> 6 </option>

<option value="7"> 7 </option>

<option value="8"> 8 </option>

<option value="9"> 9 </option>

<option value="10"> 10 </option>

<option value="11"> 11 </option>

<option value="12"> 12 </option>

<option value="13"> 13 </option>

<option value="14"> 14 </option>

<option value="15"> 15 </option>

<option value="16"> 16 </option>

<option value="17"> 17 </option>

<option value="18"> 18 </option>

<option value="19"> 19 </option>

<option value="20"> 20 </option>

<option value="21"> 21 </option>

<option value="22"> 22 </option>

<option value="23"> 23 </option>

<option value="24"> 24 </option>

<option value="25"> 25 </option>

<option value="26"> 26 </option>

<option value="27"> 27 </option>

<option value="28"> 28 </option>

<option value="29"> 29 </option>

<option value="30"> 30 </option>

<option value="31"> 31 </option>

</select> <select name="smonth">

<option value="1"> January </option>

<option value="2"> February </option>

<option value="3"> March </option>

<option value="4"> April </option>

<option value="5"> May </option>

<option value="6"> June </option>

<option value="7"> July </option>

<option value="8"> August </option>

<option value="9"> September </option>

<option value="10"> October </option>

<option value="11"> November </option>

<option value="12"> December </option>

</select> <select name="syear">

<option value="2011"> 2011 </option>

<option value="2012"> 2012 </option>

</select> <br> <br> <br>

<u> <b> Choose your class </b> </u><br>

Class:

<select name=ctype>

<option value=1> Business </option>

<option value=2> Economy </option>

</select> </center><br> <br> <br>

<table border="0" cellpadding="1" class="fp1" align="center">

caption> <u> <b> Choose No. of seats </b> </u> </caption>

<tbody><tr>

<td> No. of Adults:</td>

<td> <select name="adults">

<option value="0"> 0 </option>

<option value="1"> 1 </option>

<option value="2"> 2 </option>

<option value="3"> 3 </option>

<option value="4"> 4 </option>

<option value="5"> 5 </option>

<option value="6"> 6 </option>

</select></td></tr>

<tr>

<td> No. of Children: </td>

<td><select name="children">

<option value="0"> 0 </option>

<option value="1"> 1 </option>

<option value="2"> 2 </option>

<option value="3"> 3 </option>

<option value="4"> 4 </option>

<option value="5"> 5 </option>

<option value="6"> 6 </option>

</select> </td></tr>

<tr>

<td>No. of Infants:</td>

<td><select name="infants">

<option value="0"> 0 </option>

<option value="1"> 1 </option>

<option value="2"> 2 </option>

<option value="3"> 3 </option>

<option value="4"> 4 </option>

<option value="5"> 5 </option>

<option value="6"> 6 </option>

</select></td></tr>

</tbody>

</table><br><br>

<center> <input type="submit" name="submit">

</center> </form>

</body>

</html>

**user.java:**

package agent;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.\*;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.sql.\*;

public class user extends HttpServlet  {

Connection theConnection;

Statement theStatement;

ResultSet theResult;

ServletContext sc= null;

int amount;

public void init() throws ServletException

{

sc = getServletContext();

}

@Override

protected void service(HttpServletRequest request, HttpServletResponse response)throws ServletException, IOException

{

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

try {

String src1 = request.getParameter("src");

String des1 = request.getParameter("dest");

String classtype = request.getParameter("ctype");

String adults = request.getParameter("adults");

String children = request.getParameter("children");

String infants = request.getParameter("infants");

String src = null, des = null;

int ctype = Integer.parseInt(classtype);

int srcpt = Integer.parseInt(src1);

switch(srcpt)

{

case 1:

src = "Chennai";

break;

case 2:

src = "Bangalore";

break;

case 3:

src = "Hyderabad";

break;

case 4:

src = "Goa";

break;

case 5:

src = "Kolkatta";

break;

case 6:

src = "Delhi";

break;

case 7:

src = "Mumbai";

break;

}

int despt = Integer.parseInt(des1);

switch(despt)

{

case 1:

des = "Chennai";

break;

case 2:

des = "Bangalore";

break;

case 3:

des = "Hyderabad";

break;

case 4:

des = "Goa";

break;

case 5:

des = "Kolkatta";

break;

case 6:

des = "Delhi";

break;

case 7:

des = "Mumbai";

break;

}

sc.setAttribute("src",src);

sc.setAttribute("des",des);

sc.setAttribute("sdate",request.getParameter("sdate") +"-"+ request.getParameter("smonth")+"-"+ request.getParameter("syear"));

sc.setAttribute("ctype",classtype);

sc.setAttribute("adults",adults);

sc.setAttribute("children",children);

sc.setAttribute("infants",infants);

out.println("<html> \n" +

                    " <head> \n" +

                    " <title> Choose Your Flight </title> " +

                    " </head>");

out.println("<body text = purple>\n <center> ");

out.println("<form name =user action=http://localhost:8084/Flight\_Eticketing/Payment onsubmit =validate(); >");

out.println("<table border = 1 width = 40 cellpadding = 5 cellspacing = 5 > <caption> <h1> <u> Choose Your Flight </h1> </u><br><br><br><br> </caption> <br><br><br><br><br><br> " );

out.println("<tr>");

out.println("<th> Flight # </th>");

out.println("<th> Start Point </th>");

out.println("<th> Destination </th>");

out.println("<th> ETD </th>");

out.println("<th> ETA </th>");

out.println("<th> Adult Fare </th>");

out.println("<th> Children Fare </th>");

out.println("<th> Infant Fare </th>");

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

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

theConnection = DriverManager.getConnection("jdbc:odbc:iplab", "system", "newZEPHY6191");

theStatement = theConnection.createStatement();

theResult = theStatement.executeQuery("select \* from flightinfo");

int count=0;

while(theResult.next())

{

String temp1 = theResult.getString(2);

String temp2 = theResult.getString(3);

if(src.equalsIgnoreCase(temp1) && des.equalsIgnoreCase(temp2))

{

count++;

out.println("<tr>");

String fnum = theResult.getString(1);

out.println(" <td> <input type=radio name=fno value=" + fnum +"> " +fnum+"</td> ");

out.println("<td>" + temp1 + "</td>");

out.println("<td>" + temp2 + "</td>");

out.println("<td>" + theResult.getString(4) + "</td>");

out.println("<td>" + theResult.getString(5) + "</td>");

if(ctype == 2)

{

out.println("<td>" + theResult.getString(6) + "</td>");

out.println("<td>" + theResult.getString(10) + "</td>");

out.println("<td>" + theResult.getString(11) + "</td>");

}

else if(ctype == 1)

{

out.println("<td>" + theResult.getString(7) + "</td>");

out.println("<td>" + theResult.getString(8) + "</td>");

out.println("<td>" + theResult.getString(9) + "</td>");

}

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

}}

if(count==0)

{

out.println("<tr>");

out.println(" <td colspan = 12> <center> Sorry! No flights available. . .</center> </td>");

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

}

out.println("</table> ");

out.println("<br> <br> <a href= http://localhost:8084/Flight\_Eticketing/reservationform.html  > Click here to get back to home page </a> \n <input type = submit value = Submit > \n </center> \n</form> \n  </body> \n </html>");

theResult.close();

theStatement.close();

theConnection.close();

}

catch(Exception e)

{

e.getMessage();

}

finally

{

out.close();

}}

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)throws ServletException, IOException   {

service(request, response);

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)throws ServletException, IOException

{

service(request, response);

}

@Override

public String getServletInfo() {

return "Short description";

}

}

**Payment.java:**

package agent;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.\*;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.sql.\*;

public class Payment extends HttpServlet {

Connection theConnection;

Statement theStatement;

ResultSet theResult;

int ctype, adults, children, infants, amount;

ServletContext sc = null;

@Override

public void init() throws ServletException

{

sc = getServletContext();

}

@Override

protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

try {

out.println("<html> \n " +

" <head> \n" +

" <title> Confirm Details </title>\n" +

" </head>\n");

out.println("<body text = olive>\n <center> <br> <br> <br> \n ");

String fno = request.getParameter("fno");

sc.setAttribute("fno", request.getParameter("fno"));

String ct = (String)sc.getAttribute("ctype");

String ad = (String)sc.getAttribute("adults");

String chd = (String)sc.getAttribute("children");

String inft = (String)sc.getAttribute("infants");

String fnum = null;

amount = calcPayment(fno, ct, ad, chd, inft);

String amt = "Amount: "+amount;

sc.setAttribute("amt", amt);

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

theConnection = DriverManager.getConnection("jdbc:odbc:iplab", "system", "newZEPHY6191");

theStatement = theConnection.createStatement();

theResult = theStatement.executeQuery("select \* from flightinfo;");

while(theResult.next())

{

fnum = theResult.getString(1);

if(fno.equals(fnum))

{

out.println("<h1> <u> Your Ticket Details </u> </h1>");

out.println(" <br> <br> Flight Number: "+ fnum );

out.println(" <br>Start Point: "+theResult.getString(2) );

out.println(" <br>Destination Point: "+theResult.getString(3) );

out.println(" <br>ETD: "+theResult.getString(4) );

out.println(" <br>ETA: "+theResult.getString(5) );

if(ctype == 2)

{

out.println(" <br> Economy Adult Fare:  "+theResult.getString(7)+ "\t No. of adults booked:"+ adults);

out.println(" <br> Economy Children Fare:  "+theResult.getString(8)+ "\t No. of children booked:"+ children );

out.println(" <br> Economy Infant Fare:  "+theResult.getString(9)+ "\t No. of infants booked:"+ infants );

}

else if(ctype == 1)

{

out.println(" <br>Business Adult Fare: "+theResult.getString(6)+ "\t No. of adults booked:"+ adults  );

out.println(" <br>Business Children Fare: "+theResult.getString(10)+ "\t No. of children booked:"+ children );

out.println(" <br>Business Infant Fare: "+theResult.getString(11)+ "\t No. of infants booked:"+ infants );

}

out.println("<br> <br> <b> The total Ticket price: "+amount+ "</b>");

}

}

out.println( "<br> <br> <br> <br>  <a href=http://localhost:8084/Flight\_Eticketing/reservationform.html > Not Satisfied! Click here to move back and make another selection <a> ");

out.println( "<br> <br> <br> <br>  <a href=http://localhost:8084/Flight\_Eticketing/payment.html > Yup! Satisfied!  Lemme make the payment! <a> ");

out.println(" </body> \n </html>");

theResult.close();

theStatement.close();

theConnection.close();

}

catch(Exception e)

{

e.getMessage();

}

finally {

out.close();

}

}

public int calcPayment(String fno, String ct, String ad, String chd, String inft)

{

int amt = 0;

try {

ctype = Integer.parseInt(ct);

adults = Integer.parseInt(ad);

children = Integer.parseInt(chd);

infants = Integer.parseInt(inft);

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

theConnection = DriverManager.getConnection("jdbc:odbc:iplab", "system", "newZEPHY6191");

theStatement = theConnection.createStatement();

theResult = theStatement.executeQuery("select \* from flightinfo;" );

while(theResult.next())

{

String fnum = theResult.getString(1);

if(fno.equals(fnum))

{

if(ctype == 2)

amt = (adults \* Integer.parseInt(theResult.getString(6))) + (children \* Integer.parseInt(theResult.getString(10))) + (infants \* Integer.parseInt(theResult.getString(11)));

else if (ctype == 1)

amt = (adults \* Integer.parseInt(theResult.getString(7))) + (children \* Integer.parseInt(theResult.getString(8))) + (infants \* Integer.parseInt(theResult.getString(9)));

}

}

theResult.close();

theStatement.close();

theConnection.close();

}

catch(Exception e)

{

e.getMessage();

}

return amt;

}

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)throws ServletException, IOException {service(request, response);

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)throws ServletException, IOException {service(request, response);

}

@Override

public String getServletInfo() {

return "Short description";

}

}

**Payment.html:**

<html>

 <head>

<title>Make your Payment</title>

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

<script type="text/javascript">

function validateForm()

{

var name =document.passenger.pname.value;

if (name.length == 0 || name==null)

{

alert("Passenger name must be filled out!");

return false;

}

var type = document.passenger.ptype.options[document.passenger.ptype.selectedIndex].value

if (type==null)

{

alert("Please select the Lead passenger type!");

return false;

}

var addr = document.passenger.addr.value;

if(addr.length==0 || addr==null )

{

alert("Passenger address must be filled out!");

return false;

}

var pvalid = "0123456789";

var phone=document.passenger.phone.value

if(phone=="")

{

alert("Phone number is a required field!");

return false;

}

if(phone.length != 10)

{

alert("Invalid phone number! Please try again.");

return false;

}

for (var i=0; i < phone.length; i++)

{

temp = "" + phone.substring(i, i+1);

if (pvalid.indexOf(temp) == "-1")

{

alert("Invalid characters in your phone.  Please try again.");

return false;

}

}

var ctype= document.passenger.ctype.options[document.passenger.ctype.selectedIndex].value

if (ctype==null)

{

alert("Please select the Card type!");

return false;

}

var cvalid = "0123456789";

var cno=document.passenger.cnum.value

if(cno=="")

{

alert("Card number is a required field!");

return false;

}

if(cno.length != 16)

{

alert("Invalid card number! Please try again.");

return false;

}

for (i=0; i < cno.length; i++)

{

temp = "" + cno.substring(i, i+1);

if (cvalid.indexOf(temp) == "-1")

{

alert("Invalid characters in your card number field.  Please try again.");

return false;

}

}

var vdt =document.passenger.vdate.value

if (vdt==null || vdt=="")

{

alert("Card validity date must be filled out");

return false;

}

var edt =document.passenger.edate.value

if (edt==null || edt=="")

{

alert("Card expiry date must be filled out");

return false;

}

var pinno = document.passenger.pin.value;

var svalid = "0123456789";

if(pinno=="")

{

alert("PIN is a required field!");

return false;

}

if(pinno.length != 4)

{

alert("Invalid PIN! Please try again.");

return false;

}

for (i=0; i < pinno.length; i++)

{

temp = "" + pinno.substring(i, i+1);

if (svalid.indexOf(temp) == "-1")

{

alert("Invalid characters in your PIN field.  Please try again.");

return false;

}

}

}

</script>

</head>

<body>

<br> <br> <br> <h1 class ="m1">  Make Payment </h1> <br> <br> <br>

<form name = "passenger" action = "http://localhost:8084/Flight\_Eticketing/Passenger" onsubmit= "return validateForm();">

<table border="0" align="center" class="mp1">

<tr>

<td>Lead Passenger Name: </td>

<td> <input type = text name = pname> </td> </tr>

<tr> <td> Lead Passenger Type: </td>

<td> <select name = ptype>

<option value = 1> Adult </option>

<option value = 2> Child </option>

<option value = 3> Infant </option>

</select> </td> </tr>

<tr>

<td> Address: </td>

<td> <input type = textarea name = addr> </td> </tr>

<tr> <td> Phone: </td> <td> <input type = text name = phone> </td>

</tr>

<tr> <td> Card Type: </td> <td> <select name = ctype>

<option value = 1> Visa </option>

<option value = 2> Master Card </option>

</select>  </td>

</tr>

<tr> <td>

Card Number: </td> <td> <input type = text name = cnum> </td></tr>

<tr> <td>Validity date: </td> <td> <input type = text name = vdate> </td></tr>

 <tr> <td>Expiry date: </td> <td> <input type = text name = edate> </td></tr>

<tr> <td> PIN: </td> <td> <input type = text name = pin> </td></tr> </table><br> <br>

<center><input type = submit value = submit></center>

</form>

</body>

</html>

**passenger.java:**

package agent;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.\*;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.sql.\*;

public class Passenger extends HttpServlet

{

Connection theConnection;

 Statement theStatement;

ResultSet theResult;

String agency = "Bon Voyage Travel Agency"  ;

String branch = "Chennai";

ServletContext sc ;

 int res;

@Override

public void init()

{

 sc = getServletContext();

}

@Override

protected void service(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

 response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

Try

 {

String pname = request.getParameter("pname");

String ptype = request.getParameter("ptype");

String addr = request.getParameter("addr");

String phone = request.getParameter("phone");

String ctype = request.getParameter("ctype");

String cnum = request.getParameter("cnum");

String vdate = request.getParameter("vdate");

String edate = request.getParameter("edate");

String pin = request.getParameter("pin");

String amt = (String)sc.getAttribute("amt");

String fno = (String) sc.getAttribute("fno");

String amount = amt.substring(8);

sc.setAttribute("pname",pname);

sc.setAttribute("ptype",ptype);

sc.setAttribute("addr",addr);

sc.setAttribute("phone",phone);

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

theConnection = DriverManager.getConnection("jdbc:odbc:iplab", "system", "newZEPHY6191");

theStatement = theConnection.createStatement();

 res =   theStatement.executeUpdate("INSERT INTO PAYMENT(PNAME, PTYPE, PADDR, PPHONE, CARDTYPE, CARDNO, VALIDFROM, EXPIRYDATE,PIN,AMT) VALUES (\'"+pname+"\' ,\'"+ptype+"\' ,\' "+addr+"\' ,\' "+phone+"\' ,\' "+ctype+"\' ,\' "+cnum+"\' ,\' "+vdate+"\' ,\' "+edate+"\' ,\' "+pin+"\' ,\' "+amount +"\' ); " )   ;

res =   theStatement.executeUpdate("INSERT INTO AIRLINE\_PAYMENT(AGENCYNAME, BRANCH, PASSENGERNAME, FLIGHTNO, AMOUNT) VALUES (\'"+agency+"\' ,\'"+branch+"\' ,\' "+pname+"\' ,\' "+fno+"\' ,\' "+amount+"\' ); " )   ;

out.println("<html>");

out.println("<head>");

out.println("<title>Make your Payment</title>");

out.println("</head>");

out.println("<body text = teal> <br> <br> <br> <br> <br> <br> ");

out.println("<CENTER> <H1> <B> <br> <br> <br> Thanks for using our service!</B> </H1> <br> <br> <br> </CENTER>");

out.println("<center> <a href= http://localhost:8084/Flight\_Eticketing/ticket > Click here to get your Tickets </a> </center>");

out.println("<br> <br> <br> Please do take up a hard copy of the ticket which is to be generated.");

out.println("</body>");

out.println("</html>");

}

catch(Exception e)

{

System.out.println(e);

}

finally

{

out.close();

}

}

}

**ticket.java:**

package agent;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.\*;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.sql.\*;

public class ticket extends HttpServlet {

Connection theConnection;

Statement theStatement;

ResultSet theResult;

ServletContext sc;

@Override

public void init()

{

sc = getServletContext();

}

protected void processRequest(HttpServletRequest request, HttpServletResponse response)throws ServletException, IOException

{

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

String pname = (String)sc.getAttribute("pname");

String ptype = (String)sc.getAttribute("ptype");

String addr = (String)sc.getAttribute("addr");

String phone = (String)sc.getAttribute("phone");

String adults = (String)sc.getAttribute("adults");

String children = (String)sc.getAttribute("children");

String infants = (String)sc.getAttribute("infants");

String ctype = (String)sc.getAttribute("ctype");

String src = (String)sc.getAttribute("src");

String des = (String)sc.getAttribute("des");

String fno = (String)sc.getAttribute("fno");

String tdate = (String)sc.getAttribute("sdate");

String amt = (String)sc.getAttribute("amt");

try {

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

out.println("<html>");

out.println("<head>");

out.println("<title>Your Ticket</title>");

out.println("</head>");

out.println("<body bgcolor = MAROON text = BLACK > <br> <br> <br> <B> ");

out.println("<table border=0 align = center> <caption> <h1> <CENTER> <B> THE INDIAN </B> </CENTER> </h1></caption> ");

out.println("<tr> <td rowspan = 15> <img src=./images.jpg width=180 height=400> </td> <td> Lead Passenger Name: </td> <td>  "+pname+" </td> </tr>");

out.println(" <tr>  ");

out.println(" <td>Lead Passenger Type: </td> <td> "+ptype+"</td> </tr>");

out.println("<tr> <td>Lead Passenger Address:</td> <td> "+addr+"</td> </tr>");

out.println("<tr> <td>Lead Passenger Phone Number:</td> <td> "+phone+"</td> </tr>");

out.println("<tr> <td>No. of Adults:</td> <td> "+adults+"</td> </tr>");

out.println("<tr> <td>No. of Children:</td> <td> "+children+"</td> </tr>");

out.println("<tr> <td>No. of Infants:</td> <td> "+infants+"</td> </tr>");

out.println("<tr> <td>Class: </td> <td>"+ctype+"</td> </tr>");

out.println("<tr> <td>Starting Point:</td> <td> "+src+"</td> </tr>");

out.println("<tr> <td>Destination Point:</td> <td> "+des+"</td> </tr>");

out.println("<tr> <td>Flight No.:</td> <td> "+fno+"</td> </tr>");

out.println("<tr> <td>Travel Date:</td> <td> "+tdate+"</td> </tr>");

out.println("<tr> <td colspan = 2> Total "+amt+"</td> </tr> </table> </B>");

out.println("<center> This is an e-ticket. This is a computer generated copy and does not required to be signatured. </center>");

out.println("</body>");

out.println("</html>");

}

catch(Exception e)

{

System.out.println(e);

}

finally {

out.close();

}

}

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)throws ServletException, IOException

 {

processRequest(request, response);

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)throws ServletException, IOException

{

processRequest(request, response);

}

@Override

public String getServletInfo() {

return "Short description";

}

}

**rformvalidation.js:**

function validateForm()

{

var src = document.userapp.src.options[document.userapp.src.selectedIndex].value

var dest = document.userapp.dest.options[document.userapp.dest.selectedIndex].value

var d = document.userapp.sdate.options[document.userapp.sdate.selectedIndex].value

var m = document.userapp.smonth.options[document.userapp.smonth.selectedIndex].value

var y = document.userapp.syear.options[document.userapp.syear.selectedIndex].value

var a = document.userapp.adults.options[document.userapp.adults.selectedIndex].value

var c = document.userapp.children.options[document.userapp.children.selectedIndex].value

var i = document.userapp.infants.options[document.userapp.infants.selectedIndex].value

var dt = new Date();

var cd = dt.getDate();

var cm = dt.getMonth();

var cy = dt.getFullYear();

//VALIDATE SRC AND DES POINTS

if (src == dest)

{

alert("Src and destination cannot be the same!");

return false;

}

//VALIDATE TRAVEL DATE

if(y==cy)

{

if((m<=cm & d<cd) )

{

alert("Enter a Valid date!");

return false;

}

}

if(((m==2)&(d>29)) | ((m==4)&(d>30)) | ((m==6)&(d>30)) | ((m==9)&(d>30)) | ((m==11)&(d>30))) {

alert("Enter a Valid date!");

return false;

}

//VALIDATE NO. OF SEATS

 if((a==0) & (c==0) & (i==0))

{

alert("Enter Correct number of passengers!");

return false;

}

}

**ticket.css:**

.f1 {

color: #6666ff;

font-family: Verdana,Arial,Helvetica,sans-serif;

background-color: white;

text-align: center;

vertical-align: middle;

line-height: normal;

border-right-style: outset;

border-left-style: outset;

border-bottom-style: outset;

border-top-style: outset;

border-right-color: #6666ff;

border-left-color: #6666ff;

border-bottom-color: #6666ff;

border-top-color: #6666ff;

font-size: 18px;

text-decoration: underline;

font-style: italic;

font-weight: bold;

}

.fp1

{

color: #6666ff;

font-family: serif;

background-color: white;

text-align: center;

vertical-align: middle;

line-height: normal;

font-size: 18px;

}

.fp2

{

text-align: center;

vertical-align: super;

}

.fp3

{

font-size: XX-large;

}

.m1

{

color:maroon;

font-family: Verdana,Arial,Helvetica,sans-serif;

background-color: white;

text-align: center;

vertical-align: middle;

line-height: normal;

border-right-style: outset;

border-left-style: outset;

border-bottom-style: outset;

border-top-style: outset;

border-right-color: maroon;

border-left-color: maroon;

border-bottom-color: maroon;

border-top-color: maroon;

font-size: 18px;

text-decoration: underline;

font-style: italic;

font-weight: bold;

}

.mp1

{

color: maroon;

font-family: serif;

background-color: white;

text-align: center;

vertical-align: middle;

line-height: normal;

font-size: 18px;

}

.u1

{

color: green;

font-weight: bold;

font-style: italic;

text-decoration: underline;

font-size: XX-large;

font-family: sans-serif;

border-right-color: purple;

border-left-color: purple;

border-bottom-color: purple;

border-top-color: purple;

background-image: url(../fb.jpg);

background-position: top center;

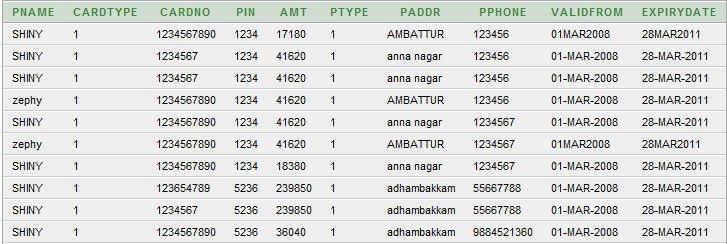
}

**DATABASE – TABLES:**

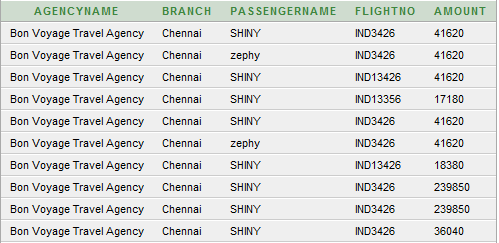
1. **Table Name:  FLIGHTINFO**



1. **Table Name:  Payment**

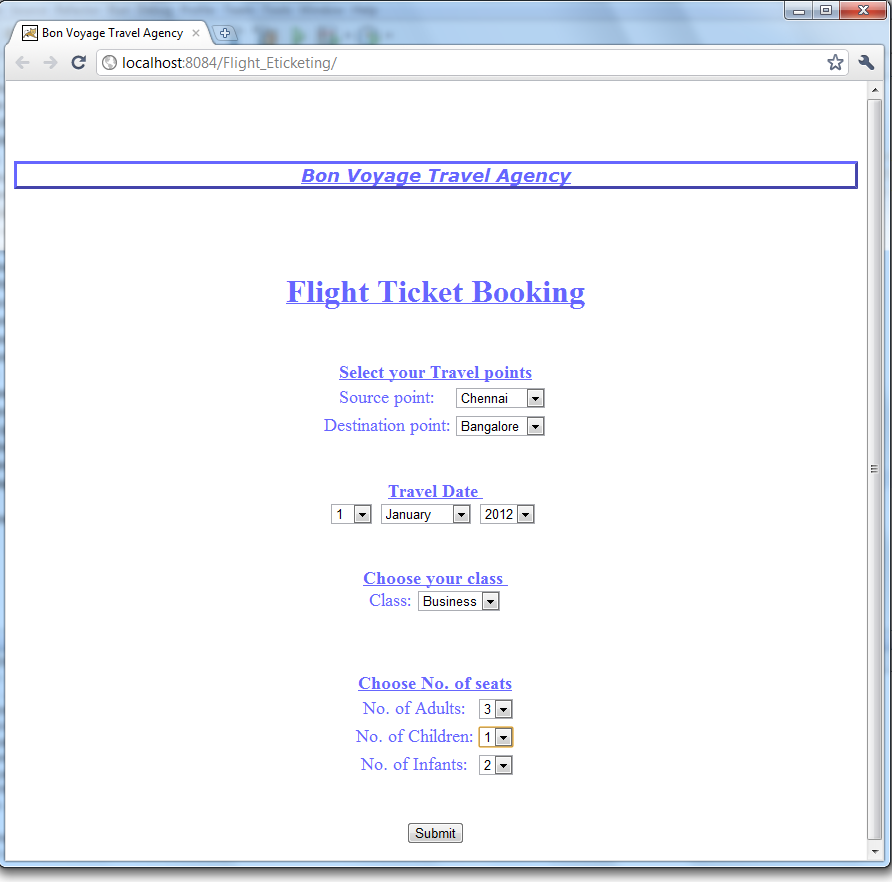


1. **Table Name:  Airline\_payment**

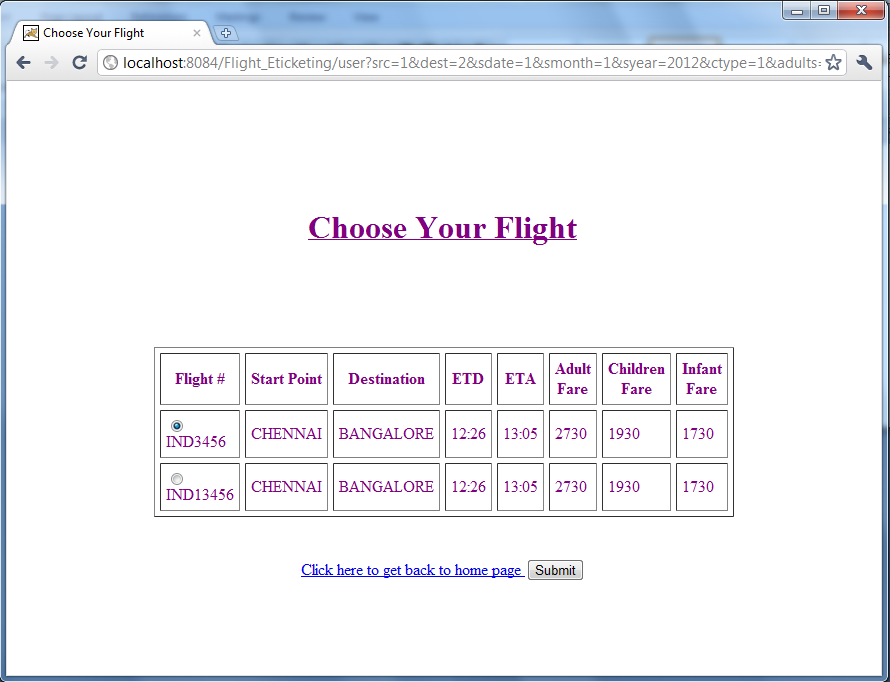


**OUTPUT:**

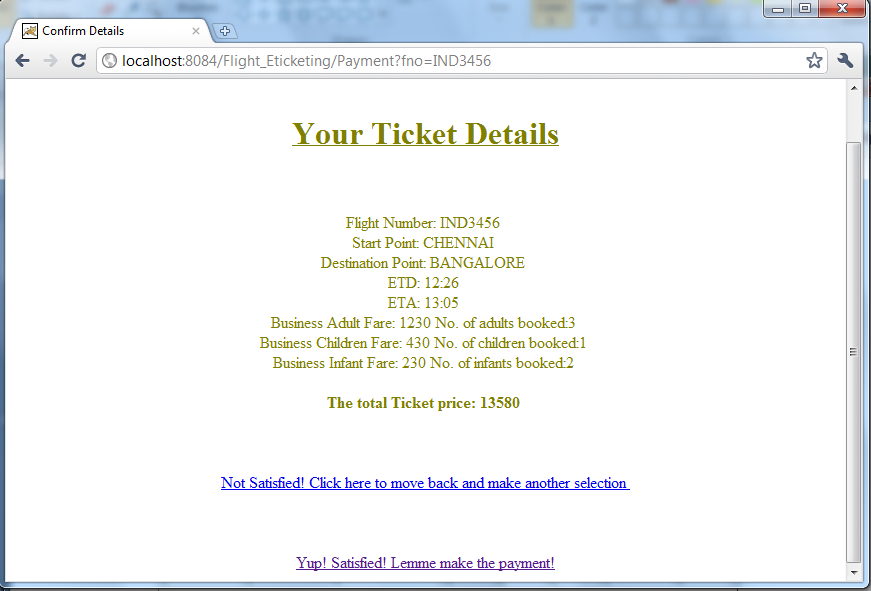
**Reservationform.html:**

****

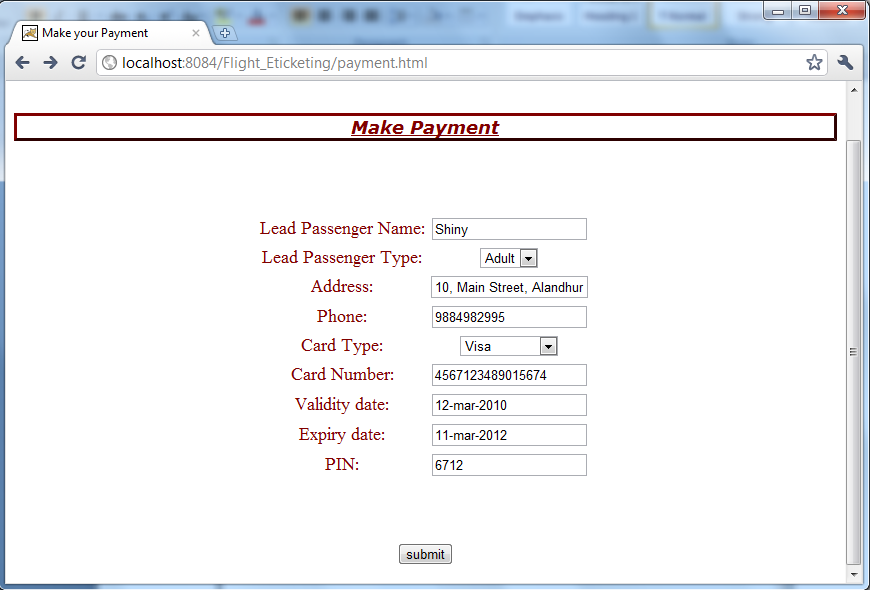
**User.java:**

****

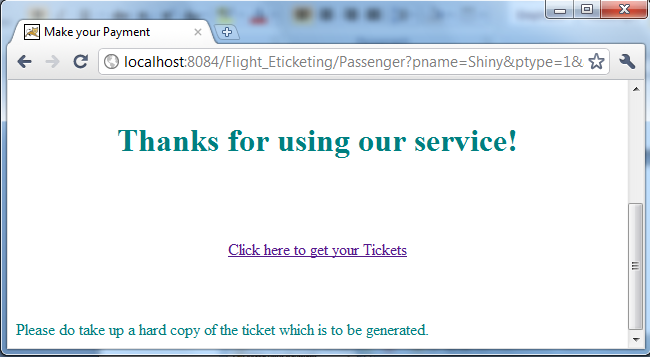
**Payment.java:**

****

**Payment.html:**

****

**Passenger.java:**

****

**Ticket.java:**

****