University Department of Information Technology

University of Mumbai
Ranade Bhavan, B-wing, Ground Floor
Vidyanagari Campus, Kalina

Santacruz (East), Mumbai - 400098

ENTERPRISE JAVA

SUBMITTED TO

G.R. PATIL COLLEGE, SONARPADA

In

Partial Fulfillment of

B.Sc. in Information Technology

BY

Utkarsh Pyati

Under the guidance of

Miss. Shubhangi Ingole

Through

University Department of Information Technology,

University of Mumbai, Mumbai

Year 2021 – 22

G.R.PATIL COLLEGE OF ARTS,COMMERCE AND SCIENCE

SONARPADA DOMBIVALI(E)

(Affiliated to University of Mumbai)

Roll NO: 2115

This is to certify that **Utkarsh Pyati** Has Satisfactory
Completed the Practical's on Enterprise JAVA to be Submitted in the Partial fulfilment of Bachelor of Science in Of **Information Technology** during the Academic year 2021-2022.

Subject –

Teacher:-Prof. Shubhangi Ingole

Principal:-Prof. Yatin Kene

INDEX

Practical No. 01

Program 1a – Create a simple calculator using Servlet

Code:

index.html

```
<html>
<head>
<title>Calculator Application</title>
</head>
<body>
<form action="CalculatorServlet" >
   Enter First Number <input type="text" name="txtN1" ><br>
  Enter Second Number  <input type="text" name="txtN2" ><br>
    Select an Operation <input type="radio" name="opr" value="+">ADDITION
 <input type="radio" name="opr" value="-">SUBTRACTION
<input type="radio" name="opr" value="*">MULTIPLY
 <input type="radio" name="opr" value="/">DIVIDE <br>
<input type="reset"> <input type="submit" value="Calculate" >
</form>
</body>
</html>
<html>
<head>
<title>Login Form</title>
</head>
   <form action="LoginServlet" >
```

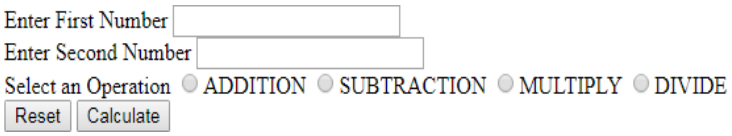
```
    Enter User ID<input type="text" name="txtId"><br>
    Enter Password<input type="password" name="txtPass"><br>
    <input type="reset">
    <input type="submit" value="Click to Login" >
  </form>
</html>
```

CalculatorServlet.java

```
package mypack;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class CalculatorServlet extends HttpServlet
{
  public void doGet(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    out.println("<html><head><title>Servlet CalculatorServlet</title></head><body>");
    double n1 = Double.parseDouble(request.getParameter("txtN1"));
    double n2 = Double.parseDouble(request.getParameter("txtN2"));
    double result =0;
    String opr=request.getParameter("opr");
    if(opr.equals("+")) result=n1+n2;
```

```
        if(opr.equals("-")) result=n1-n2;
        if(opr.equals("*")) result=n1*n2;
        if(opr.equals("/")) result=n1/n2;
        out.println("<h1> Result = "+result);
        out.println("</body></html>");
}
 }
```

Output -



Addition –

Result = 5.0

Q.1 b) Create a servlet for a login page. If the username and password are correct then it says message "Hello" else a message "login failed".

Code:

index.html

```
<html>
```

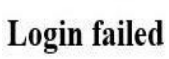
```
    <body>
    <form action="LoginServlet"  method="post">
      UserName : <input type="text" name="uname"><br>
      Password : <input type="password" name="pw"> <br>
      <input type="submit" value="LOGIN">
     </form>
  </body>
</html>
```

LoginServlet.java

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

import javax.servlet.http.*; public class

LoginServlet extends HttpServlet {

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

 {

response.setContentType("text/ht

ml");

   PrintWriter out = response.getWriter();
    String username=request.getParameter("uname");
    String password=request.getParameter("pw");
    String msg="";
    if (username .equals("admin") && password.equals("admin123"))

msg="Hello "+username;

    else     msg="Login failed";

out.println("<b>"+msg+"<b>"

);
```

```
 }
}
```

Output -



Q.1 c) Create a registration servlet in Java using JDBC. Accept the details such as Username, Password, Email, and Country from the user using HTML Form and store the registration details in the database.

Code:

index.html

```
<html>
<body>
<form action = “RegistrationServlet” method = “post”>
User name: <input type = “text” name = “uname”> <br>
Password: <input type = “password” name = “pw”> <br>
Email Id: <input type = text” name = “email” > <br>
Country: <select name = “coun”>
<option> select…
<option> India
<option> Banladesh
```

<option> Bhutan

<option> Canada

</select>

<input type = “submit” value = “Register”>

</form>

</body>

</html>

RegistrationServlet.java

import java.io.*; import java.sql.*; import

javax.servlet.*; import javax.servlet.http.*;

public class RegistrationServlet extends

HttpServlet

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

{ Connection con=null;

PreparedStatement ps=null;

response.setContentType("text/html");

PrintWriter out = response.getWriter();

String username=request.getParameter("uname");

String password=request.getParameter("pw");

String emailid=request.getParameter("email");

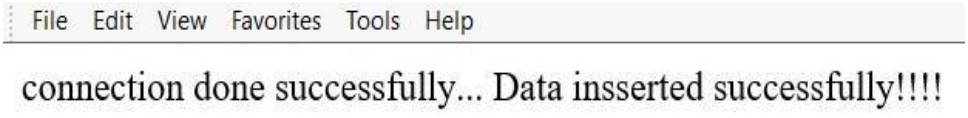
String country=request.getParameter("coun");

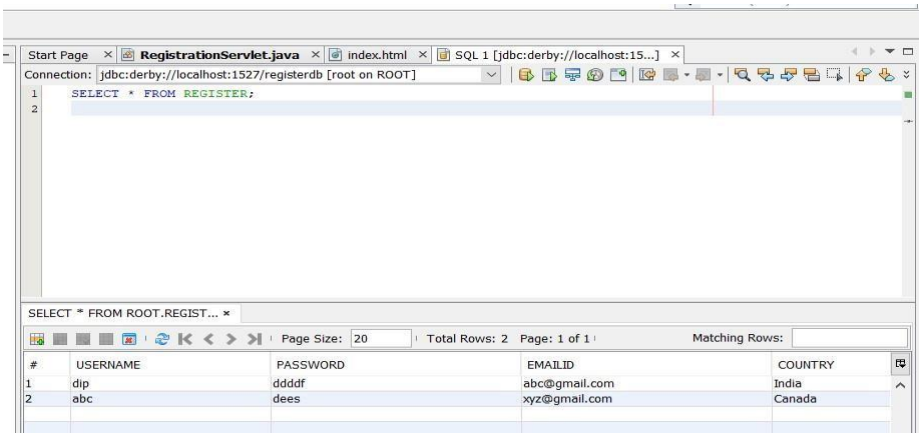
try

{ Class.forName("com.mysql.jdbc.Driver");

```
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/registerdb","root","tige
r");          out.println("connection  done  successfully...");
ps=con.prepareStatement("insert into user values (?,?,?,?)");
ps.setString(1,username);          ps.setString(2,password);          ps.setString(3,emailid);
ps.setString(4,country);          ps.execute();

          out.print("Data insserted successfully!!!!");
       }
      catch(Exception e) { out.println(e); }
out.println("<b>"+"<b>");

    }
}
```

Output –





Practical No. 02

a) Using Request Dispatcher Interface create a Servlet which will validate the password entered by the user, if the user has entered "Servlet" as password, then he will be forwarded to Welcome Servlet else the user will stay on the index.html page and an error message will be displayed.

CODE:

Index.html

```
<html>
<body>
<form method="post" action="ValidateServlet">
        User Name: <input type="text" name ="un"><br>
        Password: <input type="password" name ="pw"><br>
        <input type="submit" value="Login">
    </form>
   </body>
</html>
```

ValidateServlet.java

```
import java.io.*; import javax.servlet.*;
import javax.servlet.http.*; public class
ValidateServlet extends HttpServlet
{ public void doPost(HttpServletRequest req, HttpServletResponse res)throws IOException, ServletException
    {       res.setContentType("text/html");
          PrintWriter out=res.getWriter();
            String username=req.getParameter("un");
          String password=req.getParameter("pw");
          if(password.equals("Servlet"))
           {
```

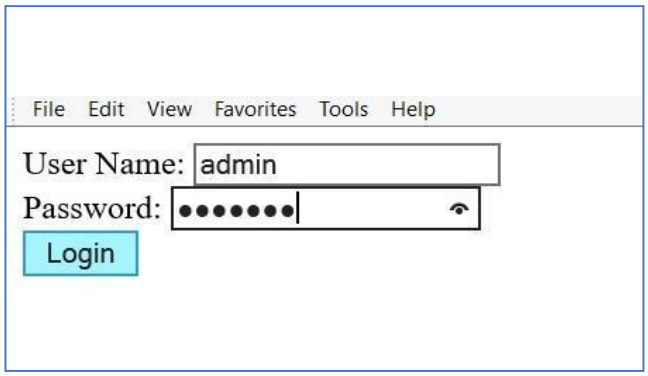
```
            req.setAttribute("s1username",username);
req.setAttribute("s1password",password);
            RequestDispatcher rd= req.getRequestDispatcher("/WelcomeServlet");
             rd.forward(req, res);
          }
else
        {
             out.print("Incorrect password");
              RequestDispatcher rd= req.getRequestDispatcher("/index.html");
             rd.include(req, res);
         }
     }
}
```

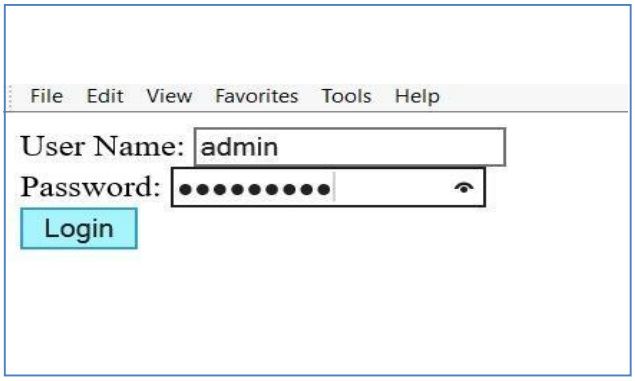
WelcomeServlet.java

```
import java.io.*; import javax.servlet.*;
import javax.servlet.http.*; public class
WelcomeServlet extends HttpServlet
{
   public void doPost(HttpServletRequest req, HttpServletResponse res) throws ServletException,
IOException
 {
res.setContentType("text/html");
try (PrintWriter out =
res.getWriter()) {
```

```
        String s2username =
(String)req.getAttribute("s1username"); String
s2password = (String)req.getAttribute("s2password");
out.println("Welcome "+s2username);
   }
}
}
```

Output:





Q.2 b) Create a servlet that uses Cookies to store the number of times a user has visited servlet.

CODE:

CookieServlet.java

/*

*** To change this license header, choose License Headers in Project Properties.**

*** To change this template file, choose Tools |**

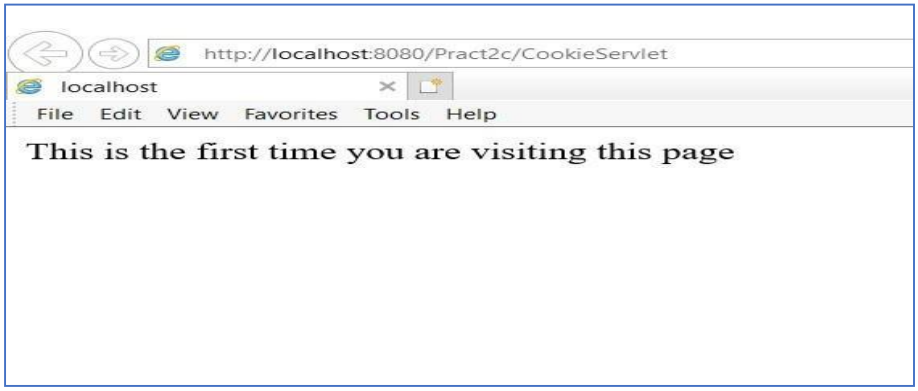
Templates * and open the template in the editor.

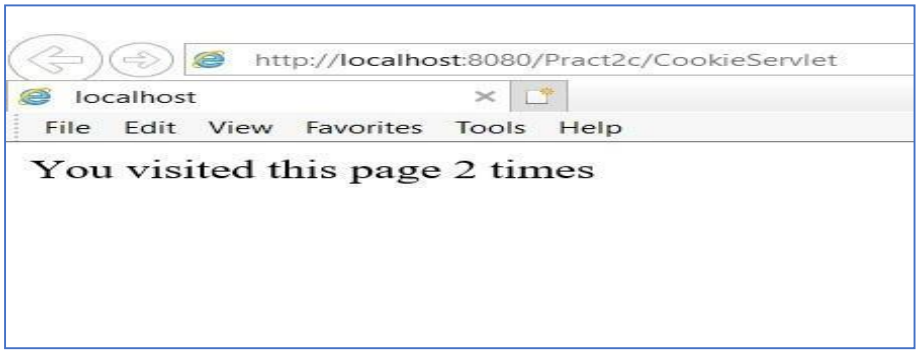
***/**

package pract2;

```
import java.io.*; import javax.servlet.*;
import javax.servlet.http.*; public class
CookieServlet extends HttpServlet
{
   private int i=1;
   public void doGet(HttpServletRequest request, HttpServletResponse
response) throws IOException, ServletException
   {    response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String k=String.valueOf(i);
Cookie c = new Cookie("visit",k);
response.addCookie(c);              int
j=Integer.parseInt(c.getValue());
        if(j==1)
        {
          out.println("This is the first time you are visiting this page");
        }
else
        {      synchronized(CookieServlet.this)
               { out.println("You visited this page "+i+" times");

}       }
i++;
   }
}
```

OUTPUT:





Q.2 c) Create a servlet demonstrating the use of session creation and destruction. Also check whether the user has visited this page first time or has visited earlier also using sessions.

CODE:

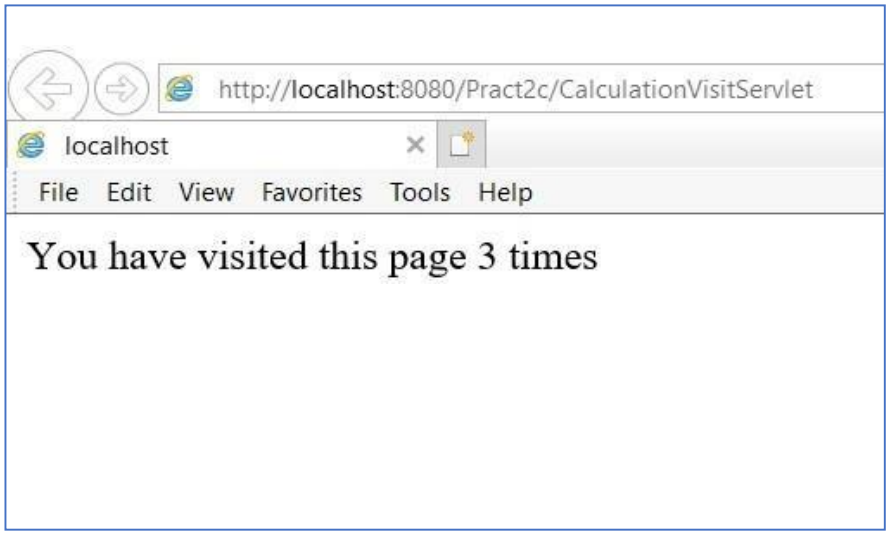
CalculationVisitServlet.java

```
package sessionapp;

/*
* To change this license header, choose License Headers in Project Properties.
* To change this template file, choose Tools | Templates
* and open the template in the editor.
 */
package pract2;
/*
```

```
* To change this license header, choose License Headers in Project Properties.
* To change this template file, choose Tools | Templates
* and open the template in the editor.
 */

import java.io.*; import javax.servlet.*;
import javax.servlet.http.*; public class
HttpSessionServlet extends HttpServlet
{    private    int
counter;
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException
  {
     response.setContentType("text/html");
     PrintWriter out = response.getWriter();
HttpSession
session=request.getSession(true);
if(session.isNew())
    {
       out.print("This is the first time you are visiting this page");
         ++counter;
    }
else
{
synch
ronize
d(Htt
```

```
pSessi
onSer
vlet.th
is)

        {
if(counter==10)
                {   session.invalidate();
counter=0;
request.getSession(false);
                 }
else
                out.print("You have visited this page "+(++counter)+ " times");
        }
     }
   }
}
```

OUTPUT:



Q.3.a) Create a servlet application to upload and download a file

Code –

Uploading a file

Index.html

<form action = “FileUploadServlet” method=”post” enctype=”multipart/form-data”>

Select File to Upload:<inputtype=”file” name=”file” id=”file”>

Destination<input type = “text” value=”/tmp” name=”destination”>

```
<br>
<input type=”submit” value=”Upload file” name=”upload” id=”upload”>
</form>
```

FileUploadServlet.java

```
package fileservletapp;
import  java.io.*;
import javax.servlet.*;
import javax.servlet.annotation.MultipartConfig;
import javax.servlet.http.*;
@MultipartConfig
public class FileUploadServlet extends HttpServlet {
public void doPost(HttpServletRequest req,HttpServletResponse res) throws ServletException,
IOException
{
res.setContentType("text/html");
PrintWriter out = res.getWriter();
String path = req.getParameter(“destination”);
  Part filePart=req.getPart("file");
    String sfilePart=req.getPart("file").toString();
out.print("<br> filePart: "+sfilePart);
    String filename=filePart.getSubmittedFileName().toString();
out.print("<br><br><hr> file name: "+filename);
    OutputStream os=null;
InputStream is=null;
 try   {
```

```
    os=new                                  FileOutputStream(new
File(path+File.separator+filename));
is=filePart.getInputStream();       int read=0;       byte[] b=new
byte[1024];     while ((read = is.read(b)) != -1) {       os.write(b,
0, read);
    }
    out.println("<br>file uploaded sucessfully...!!!");
  }
  catch(FileNotFoundException e){out.print(e);}
} }
```

Downloading a file

Index.html

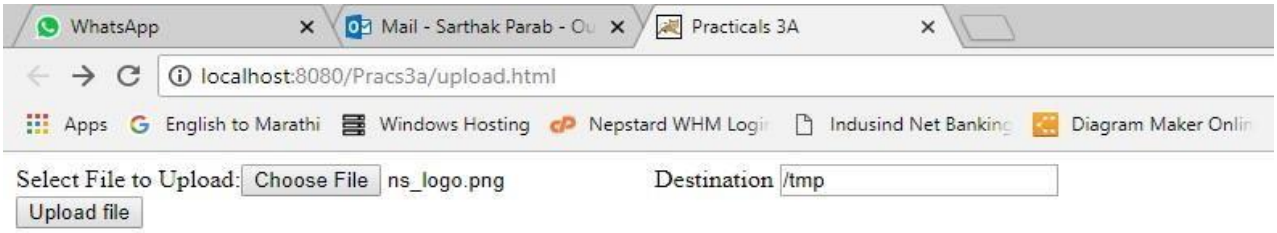
```
<body>
    <h1>File Download Application</h1>
    Click <a href="DownloadServlet?filename=SampleChapter.pdf">Sample Chapter</a>
    <br/><br/>
    Click <a href="DownloadServlet?filename=TOC.pdf">Table Of Contents</a>
</body>
```

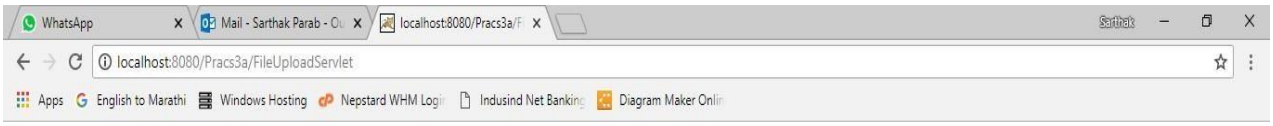
DownloadServlet.java package

```
filedownloadapp; import java.io.*; import
javax.servlet.*; import javax.servlet.http.*;
public class DownloadServlet extends
HttpServlet
{    public void doGet(HttpServletRequest request, HttpServletResponse response)
```

```
                throws ServletException, IOException
{           response.setContentType("APPLICATION/OCTET-STREAM");
           String filename = request.getParameter("filename");
           ServletContext context = getServletContext();
                InputStream is = context.getResourceAsStream("/" + filename);
ServletOutputStream os = response.getOutputStream();
response.setHeader("Content-Disposition","attachment; filename=\"" + filename + "\"");
// if comment this statement then it will ask you about the editor with which you want to
open the file
           int i;
           byte b[]=new byte[1024];
                while ((i=is.read(b)) != -1)
{                   os.write(b);
           }

is.close();
os.close();
   }
}
```

OUTPUT:





filePart: File name=ns_logo.png, StoreLocation=C:\Users\Sarthak\AppData\Roaming\NetBeans\8.0.2\config\GF_4.1\domain1\generated\jsp\Pracs3a\upload__665e6b4f_1665e440109__7ffb_00000000.tmp, size=159983bytes, isFormField=false, FieldName=file

File Download Application

Click Sample Chapter

Click Table Of Contents

Q.3.b) Develop Simple Servlet Question Answer Application using Database

Create a table in mysql

- Click on 'Services' tab
- Create a database
- Database name: queansdb Table name: queans
- Fields:
    - o queno integer primary
      key o question varchar
      200 o opt1 varchar 100
      o opt2 varchar 100 o
      opt3 varchar 100 o opt4
      varchar 100 o anskey
      varchar 1

Insert min 2 records

Right click on table-> click on 'view data' -> right click on empty dataset -> insert a record

> click on 'Add Row' -> OK

add mysql connector to Libray

- click on projects tab
- right click on libraries
- click on add jar
- browse the connector 'mysql-connector-java-5.1.23-bin' in folder: C:\Program Files\NetBeans 8.0\ide\modules\ext

click on OK

CODE:

QueAnsDBServlet.java

```
package dbapp; import javax.servlet.*;

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

import java.sql.*; public class

QueAnsDBServlet extends HttpServlet
```

```
{
  public void doGet(HttpServletRequest request, HttpServletResponse response)  throws
ServletException, IOException
 {
  response.setContentType("text/html");
PrintWriter out = response.getWriter();
  try
  {
   out.print("<html><body><br>");
out.println("<form method='post'
action='Marks'>");
    Class.forName("com.mysql.jdbc.Driver");
    Connection
con=DriverManager.getConnection("jdbc:mysql://localhost/queansdb","root","tiger");
    Statement st = con.createStatement();
    String    sql="select    *    from
queans";              ResultSet  rs  =
st.executeQuery(sql);  int i=0;
    out.print("<center>Online Exam</center>");
while(rs.next())
     {
i++;
       out.print("<br><br><hr>"+rs.getInt(1)+" ");
out.print(rs.getString(2));         out.print("<br><input type=radio
name="+i+" value=1>"+rs.getString(3));         out.print("<br><input
```

```
type=radio name="+i+" value=2>"+rs.getString(4));

out.print("<br><input type=radio name="+i+"

value=3>"+rs.getString(5));          out.print("<br><input type=radio

name="+i+" value=4>"+rs.getString(6));

        String ans="ans"+i;

        out.println("<br><input type=hidden name="+ans+" value="+rs.getString(7)+">");

      }

        out.println("<br><input type=hidden name=total value="+i+">");

out.println("<input type=submit value=submit>");



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

out.print("</body></html>");

  }

catch(Exception e)

  {

    out.println("ERROR "+e.getMessage());

   }

 }

}
```

Marks.java package dbapp;

import javax.servlet.*; import

javax.servlet.http.*; import

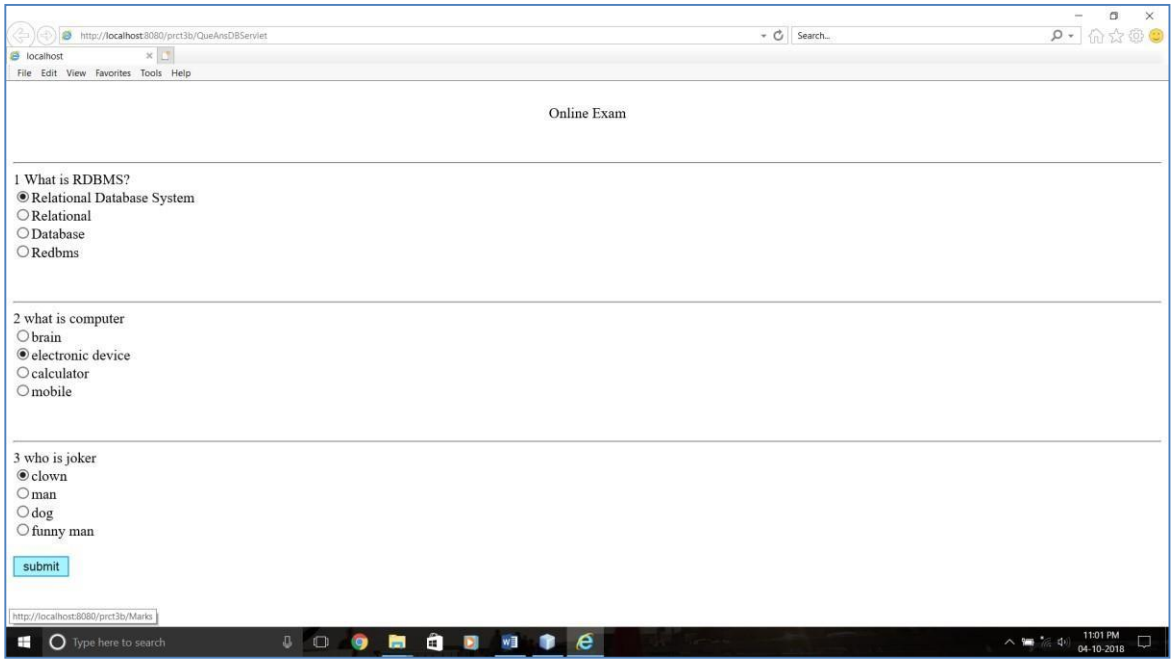
java.io.*; public class Marks

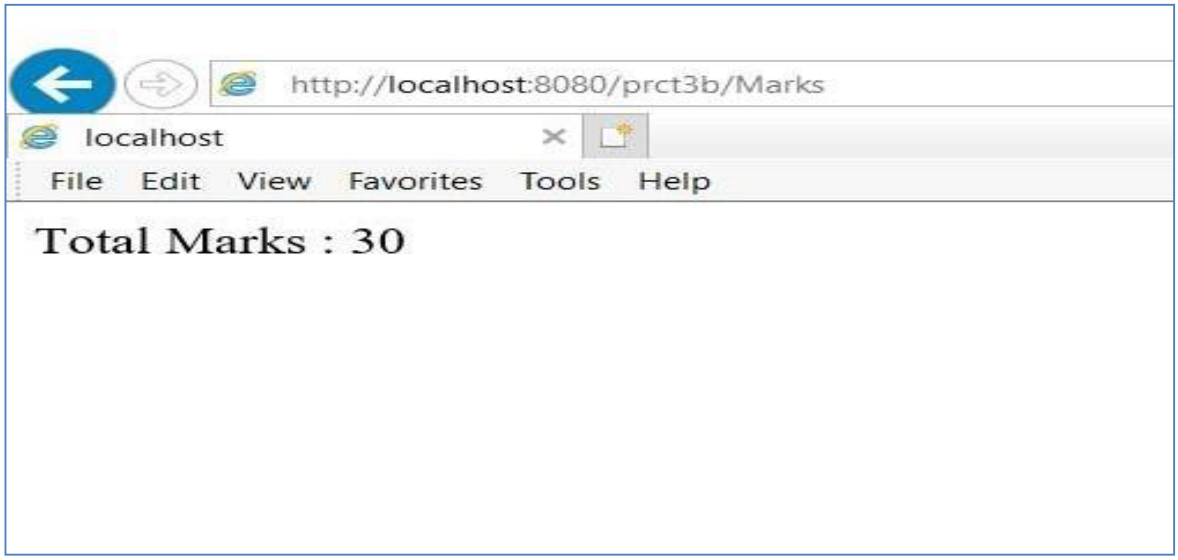
extends HttpServlet

```
{       public void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException
       {      response.setContentType("text/html");
              PrintWriter out = response.getWriter();
      try
      {   out.print("<html><body>");            int
total=Integer.parseInt(request.getParameter("total"));
int marks=0;              for(int i=1; i<=total; i++)
          {
             String                    sel=request.getParameter(new
Integer(i).toString());                                    String
ans=request.getParameter("ans"+i);                              if
(sel.equals(ans)) marks++;
           }
       out.println("Total Marks : "+marks);
out.print("</body></html>");
    }
    catch(Exception e)
    {
           out.println("ERROR "+e.getMessage());
    }
   }
}
```

Righclick on QueAnsDbServlet and Run

OUTPUT:





Q.3 c) Create simple Servlet application to demonstrate Non-Blocking Read Operation.

CODE:

Index.html

```
<html>
  <body>
    <a href="NonBlockingServlet">Non Blocking Servlet</a>
  </body>
</html>
```

ReadingListener.java package nonblkapp;

```
import java.io.*; import java.util.logging.Level;
import java.util.logging.Logger; import
javax.servlet.*; public class ReadingListener
implements ReadListener
{ ServletInputStream input = null;
  AsyncContext ac = null;
  ReadingListener(ServletInputStream in, AsyncContext c) {
 input     =
in;       ac =
c;    }
  @Override     public void
onDataAvailable()  {
  }
  public void onAllDataRead()
  {    ac.complete();
  }
public void onError(Throwable t)
{ ac.complete();
tprintStackTrace();
}
}
```

ReadingNonBlockingServlet.jav

a package nonblkapp; import

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

import

javax.servlet.annotation.WebServlet;

import javax.servlet.http.*;

@WebServlet (name = "ReadingNonBlockingServlet",

urlPatterns =

{"/ReadingNonBlockingServlet"},asyncSupported = true)

public class ReadingNonBlockingServlet extends HttpServlet

{

@Override protected void service(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException

{ response.setContentType("text/html");

AsyncContext ac = request.startAsync();

ServletInputStream

in=request.getInputStream();

in.setReadListener(new ReadingListener(in,ac));

}

}

NonBlockingServlet

.java package

nonblkapp; import

java.io.*;

```
import java.net.HttpURLConnection;

import java.net.URL; import

java.util.logging.Level; import

java.util.logging.Logger; import

javax.servlet.*; import

javax.servlet.annotation.WebServlet;

import javax.servlet.http.*;

@WebServlet(name = "NonBlockingServlet", urlPatterns =

{"/NonBlockingServlet"}) public class NonBlockingServlet extends HttpServlet {

   @Override

   protected void service(HttpServletRequest request, HttpServletResponse

response) throws ServletException, IOException {

response.setContentType("text/html");

      PrintWriter out = response.getWriter();

      String filename = "booklist.txt";

      ServletContext c = getServletContext();

      InputStream is = c.getResourceAsStream("/"+filename);

      InputStreamReader isr = new InputStreamReader(is);

      BufferedReader br = new BufferedReader(isr);

      String path = "http://" + request.getServerName() + ":" +

request.getServerPort() + request.getContextPath() +

"/ReadingNonBlockingServlet"; out.println("<h1>File Reader</h1>");

      //out.flush();

      URL url = new URL(path);
```

```
HttpURLConnection hc = (HttpURLConnection)
url.openConnection();  hc.setChunkedStreamingMode(2); //2bytes
at a time  hc.setDoOutput(true); // true if URL connection done

hc.connect();
String text = "";
System.out.println("Reading started...");
BufferedWriter bw = new BufferedWriter(new
OutputStreamWriter(hc.getOutputStream()));  while ((text = br.readLine()) != null)
{  bw.write(text);
bw.flush();
out.println(text+"<br>");
out.flush();  try
{
Thread.sleep(1000);
}
catch (Exception ex)
{
out.print(ex);
}
}
bw.write("Reading
completed...");  bw.flush();
bw.close();
}
```

}

OUTPUT:

PRACTICAL 4

Q.4 a) Develop a simple JSP application to display values obtained from the use of intrinsic objects of various types.

CODE:

index.jsp

```
<html>
<body>
<form action="implicitObjectEx.jsp">
Enter your name:<input type="text" name="myname"><br>
Enter your email id:<input type="text" name="mymailid"><br>
<input type="submit" value="submit">
</form>
</body>  </html>
```

implicitObjectE

x.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<html>
<head>
<title>JSP Page</title>
</head>
<body>
<h1>Use of Intrinsic Objects in JSP</h1>
<h1>Request Object</h1>
Query String<%=request.getQueryString() %><br>
Context Path<%=request.getContextPath() %><br>
Remote Host<%=request.getRemoteHost() %><br>
<h1>Response Object</h1>
Character Encoding Type<%=response.getCharacterEncoding() %><br>
Content Type <%=response.getContentType() %><br>
Locale <%=response.getLocale() %><br>
<h1>Session Object</h1>
ID<%=session.getId() %><br>
```

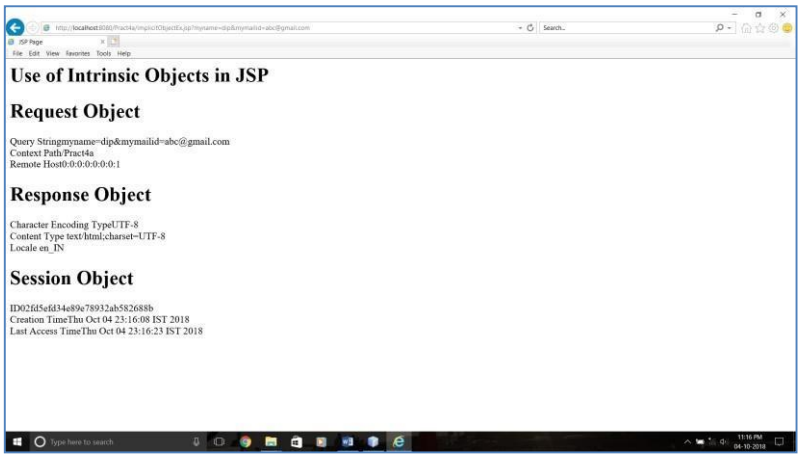
Creation Time<%=new java.util.Date(session.getCreationTime()) %>

Last Access Time<%=new java.util.Date(session.getLastAccessedTime()) %>

</body>

</html>

OUTPUT:



Q.4.b) Develop a simple JSP application to pass values from one page to another with validations. (Name-txt, age-txt, hobbies-checkbox, email-txt, gender-radio button)

CODE-

Index.jsp

```
<html>
<body>
<form action=”Validate.jsp”>
Enter your Name<input type=”text” name=”name”><br>
Enter your Age<input type=”text” name=”age”><br>
Select Hobbies<input type=”checkbox” name=”hob” value=”Singing”>Singing
<input type=”checkbox” name=”hob” value=”Reading”>Reading Books
<input type=”checkbox” name=”hob” value=”Football”>Playing Football<br>
Enter Email<input type=”text” name=”email”><br>
Select Gender<input type=”radio” name=”gender” value=”male”>Male
<input type=”radio” name=”gender” value=”female”>Female
<input type=”radio” name=”gender” value=”other”>Other<br>
<input type=”hidden” name=”error” value=””>
<input type=”submit” value=”Submit Form”>
</form>
</body>
</html>
```

CheckerBean.java

```
package mypack; import
java beans*; import
java.io.Serializable; import
java.util.regex.Matcher; import
java.util.regex.Pattern;
public class CheckerBean
{
```

```
String name.hob.email gender.error;
int age; public CheckerBean()
{ name=" "; hob=" "; email=" ";
gender=" "; error=" "; age=0; }
public void setName (String
n)
{
        {
        name=n;
        } public String getName()
        {
        return name;
        }
        public void setAge(int a)
        { age=a; } public int getAge()
        { return age;
        } public void setHob(String h)
        { hob=h;
        } public String getHob()
        { return hob;
        } public void setEmail(String e)
        { email=e;
        } public String getEmail()
        { return email;
        }
        public void setGender(String g)
        { gender=g;
        }
        public String getGender()
        { return gender;
        }
        public String getError()
```

```
{ return
error;
} public boolean
validate()
{
boolean res=true;
if(name.trim().equals(""))
{ error+="<br>Enter First
Name"; res=false; }
if(age<0||age>99)
{
error+="<br>Age Invalid";
res=false;
}
String emailregex="^[_A-Za-z0-9-]+(\\.[_A-Za-z0-9-]+)*@[A-Za-z0-9-]+(\\.[A-Za-z0-9
-]+)*(\\.[A-Za-z]{2,})$";
Boolean
b=email.matches(emailregex); if(!b) {
error+="<br>email Invalid";
res=false; } return res;
}
}
```

Validate.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8" import="mypack.*"%>
<html>
<head>
<title>JSP Page</title>
</head>
<body>
<h1>Validation Page</h1>
```

```
<jsp:useBean id="obj" scope="request"
class="mypack.CheckerBean" >
<jsp:setProperty name="obj" property="*"/>

        </jsp:useBean>
        <%if(obj.validate())
        {%>
        <jsp:forward page="successful.jsp"/>
        <% } else {%>
        <jsp:include page="index.html"/>
        <%}%>
        <%=obj.getError() %>
        </body>
        </html>
```

successful.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPEhtml>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>JSP Page</title>
</head>
<body>
<h1>DATA VALIDATED SUCCESSFULLY</h1>
</body>
</html>
```

OUTPUT –

DATA VALIDATED SUCCESSFULLY

Q.4.c) Create a registration and login JSP application to register and authenticate the user based on username and password using JDBC.

CODE:

Index.html

<html>

<head>

<title>New User Registration Page</title>

</head>

<body>

<form action=”Registration.jsp”>

<h1>New User Registration Page</h1>

Enter User Name<input type=”text” name=”txtName”>

Enter Password<input type=”password” name=txtPass1”>

Re-Enter Password<input type=”password” name=”txtPass2”>

```
Enter Email<input type=”text” name=”textEmail”><br>
Enter Country Name<select name=txtCon”>
<option>India</option>
<option>France</option>
<option>England</option>
<option>Argentina</option>
</select><br>
<input type=”submit” value=”REGISTER”><input type=”reset”>
</form>
</body>
</html>
```

Registration.jsp

```
<%@page contentType=”text/html” import=”java.sql.*”%>
<html>
<body>
<h1> Registration JSP Page</h1>
<%
String uname=request.getParameter(“txtName”);
String pass1=request.getParameter(“txtPass1”);
String pass2=request.getParameter(“txtPass2”);
String email=request.getParamter(“txtEmail”);
String ctry=request.getParameter(“txtCon”);
if(pass1.equals(pass2))
{try
{
Class forName(“com.mysql.jdbc.Driver”);
Connection
con=DriverManager.getConnection(“jdbc:mysql://localhost:3306/logindb","root","tiger"
);PreparedStatement stmt=con.prepareStatement("insert into userpass values(?,?,?,?)");
stmt.setString(1,uname); stmt.setString(2,pass1); stmt.setString(3,email);
stmt.setString(4,ctry); int row=stmt.executeUpdate(); if(row==1) {
out.println("Registration Successful");}
else {
```

```
out.println("Registration FAILED!!!!");
%>
<jsp:include page="index.html"></jsp:include>
<%
}
}catch(Exception e){out.println(e);}
} else { out.println("<h1>Password
Mismatch</h1>");
%>
<jsp:include page="index.html"></jsp:include>
<% }
%>
</body>
</html>
```

Login.html

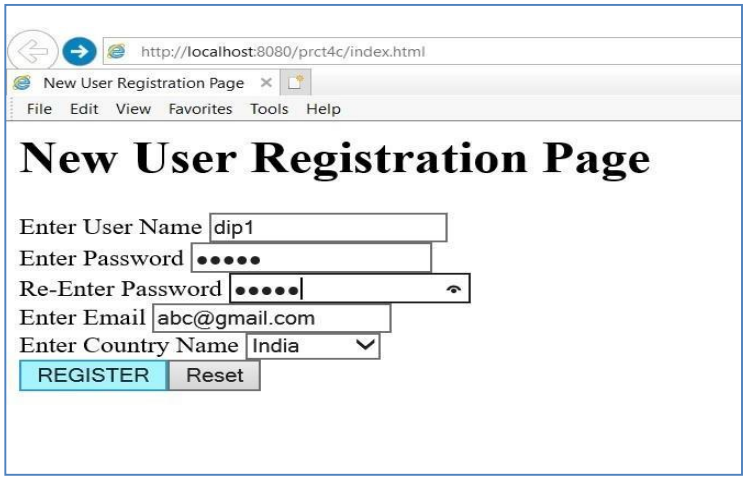
```
<html>
<body>
<h1>Login Page</h1>
<form action="Login.jsp">
Enter User Name<input type="text" name="txtName"><br>
Enter Password<input type="password" name="txtPass"><br>
<input type="submit" value="~~~LOGIN~~"><input type="reset">
</form>
</body>
</html>
```

Login.jsp

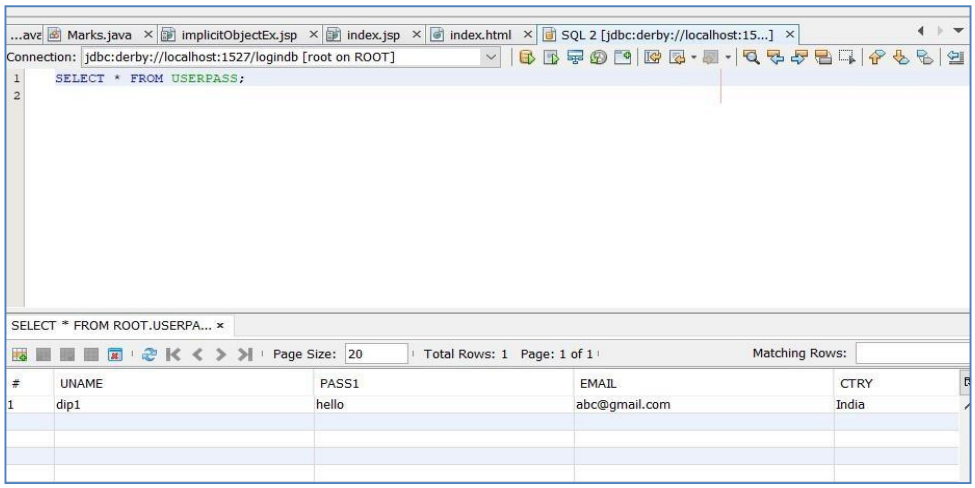
```
<%@page contentType="text/html" import="java.sql.*"%>
<html><body>
<h1>Registration JSP Page</h1>
<%
String uname=request.getParameter("txtName");
```

```
String pass=request.getParameter("txtPass");
ResultSet rs=null; try{
Class.forName("com.mysql.jdbc.Driver"); Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/logindb"","root","tiger"
); Statement stmt=con.createStatement(); rs=stmt.executeQuery("select password from
userpass where username='"+uname+"'"); rs.next(); if(pass.equals(rs.getString(1)))
{
out.println("<h1>~~~LOGIN SUCCESSFULLL~~~</h1>");
} else { out.println("<h1>password does not
match!!!!!</h1>");
%>
<jsp:include page="index.html"></jsp:include>
<%
}
}catch(Exception e){ out.println("<h1>User does
not exist!!!!!</h1>");
%>
<jsp:include page="index.html"></jsp:include>
<%
}
%>
</body>
</html>
```

OUTPUT:


http://localhost:8080/prct4c/index.html
New User Registration Page
File Edit View Favorites Tools Help
New User Registration Page
Enter User Name dip1
Enter Password
Re-Enter Password
Enter Email abc@gmail.com
Enter Country Name India
REGISTER
Reset


http://localhost:8080/prct4c/Registration.jsp?txtName=dip1&txtP
localhost
File Edit View Favorites Tools Help
Registration JSP Page
Registration Successful


Marks.java
implicitObjectEx.jsp
index.jsp
index.html
SQL 2 [jdbc:derby://localhost:15...]
Connection: jdbc:derby://localhost:1527/logindb [root on ROOT]
SELECT * FROM USERPASS;
SELECT * FROM ROOT.USERPA...
Page Size: 20
Total Rows: 1
Page: 1 of 1
Matching Rows:
UNAME PASS1 EMAIL CTRY
1 dip1 hello abc@gmail.com India

PRACTICAL 5

Q.5 a) Create an html page with fields, eno, name, age, desg, salary. Now on submit this data to a JSP page which will update the employee table of database with matching eno.

CODE:

Index.html

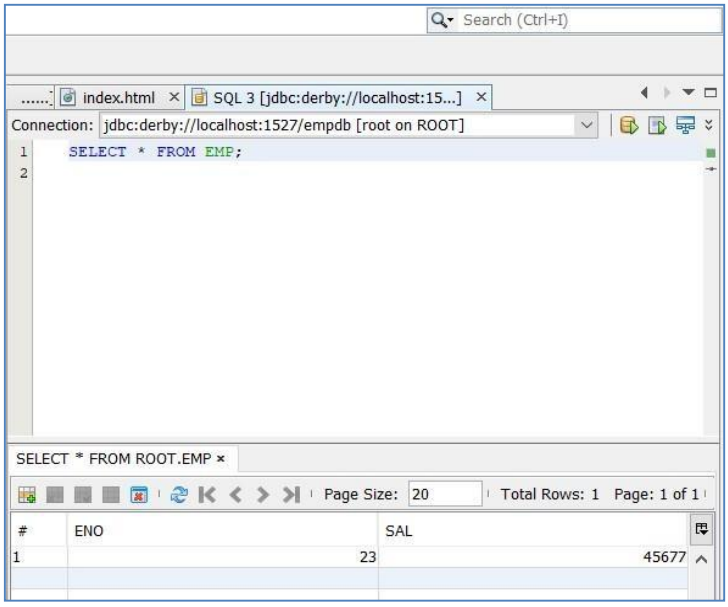
```
<html>
<body>
<form action="UpdateEmp.jsp" >
Enter Employee Number<input type="text" name="txtEno" ><br>
Enter Salary to update<input type="text" name="txtSal" ><br>
<input type="reset" ><input type="submit">
</form>
</body>
</html>
```

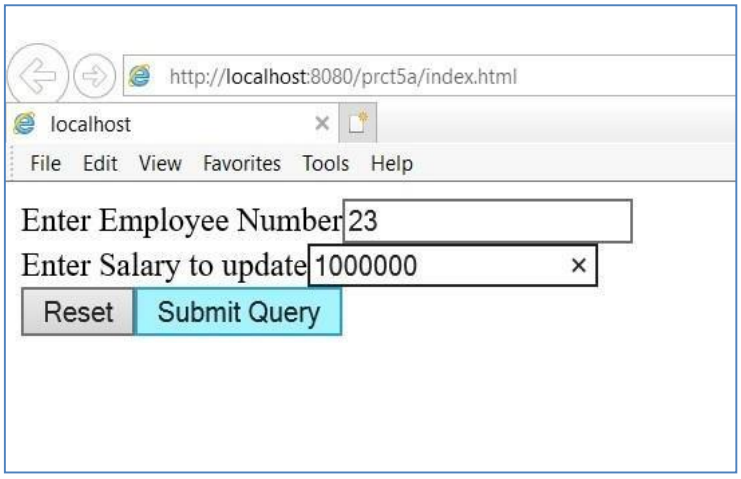
UpdateEmp.jsp

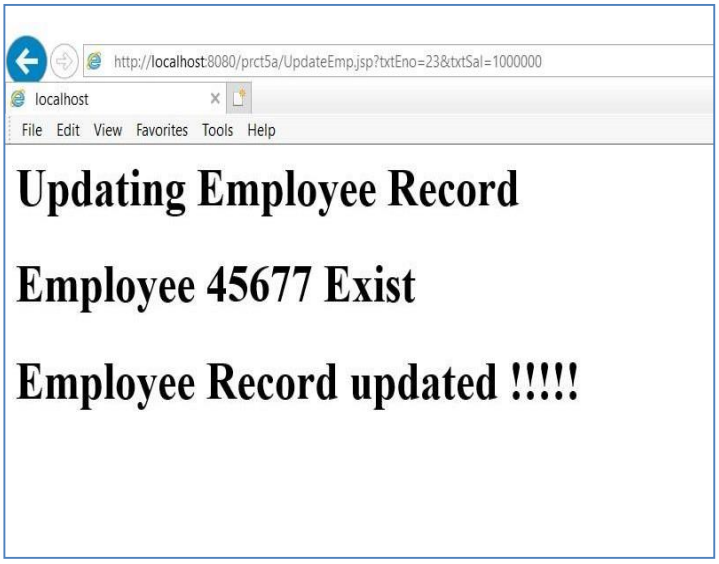
```
<%@page contentType=”text/html” import=”java.sql.*”%>
<html>
<body>
<h1>Updating Employee Record</h1>
<%
String eno=request.getParameter(“txtEno”);
String sal=request.getParameter(“txtSal”);
try{
Class.forName(“com.mysql.jdbc.Driver”);
Connection con=
DriverManager.getConnection(“jdbc:mysql://localhost:3306/empdb”,”root”,”tiger”);
PreparedStatement stmt=con.prepareStatement(“select*from emp where
empno=?”); stmt.setString(1, eno);
ResultSet rs=stmt.executeQuery(); if(rs.next()) {
out.printIn(“<h1>Employee”+rs.getString(2)+”Exist</h1>”);
PreparedStatement pst= con.prepareStatement(“update emp set salary=? where empno=?”);
```

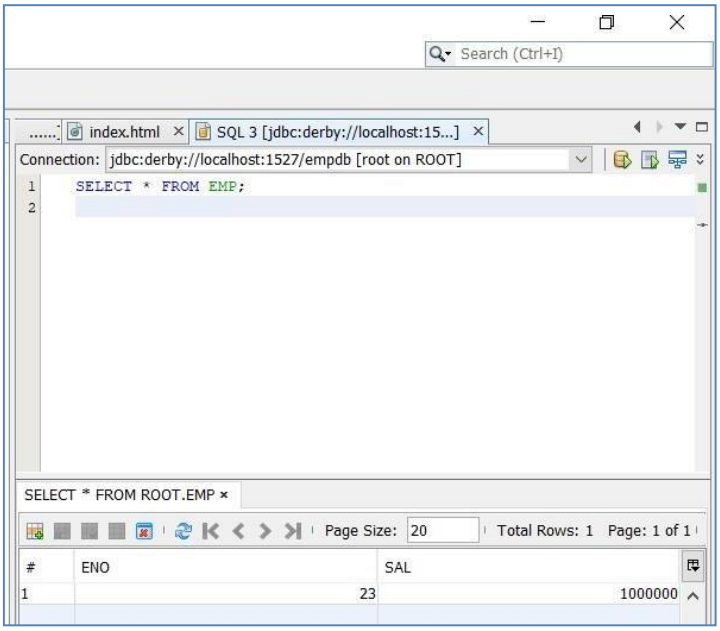
```
pst.setString(1, sal); pst.setString(2, eno);
pst.executeUpdate( ); outprintIn("<h1>Employee Record
updated !!!!!</h1>");
}
else {out.printIn("<h1>Employee Record not exist
!!!!! </h1>"); }
} catch (Exception e) {out.printIn(e); }
%>
</body>
</html>
```

OUTPUT:









Q.5 b) Create a JSP page to demonstrate the use of Expression language.

CODE:

a. Index.jsp

```
<body>
<h3>welcome to index page</h3>
<%
session.setAttribute("user","Admin");
```

```
%>
<%
Cookie ck=new Cookie("name","mycookie"); response.addCookie(ck);
%>
<form action="ExpressionLanguage.jsp">
Enter Name:<input type="text" name="name" /><br/><br/>
<input type="submit" value="Submit"/>
</form> </body>
```

b. ExpressionLanguage.jsp

```
<body>
Welcome, ${ param.name }
Session Value is ${ sessionScope.user } Cookie
name is , ${cookie.name.value}
</body>
```

c. ELArithemeticOperator.jsp

```
<body>
<%-- arithmetic op --%>
5*5+4: ${5*5+4} <br>
1.4E4+1.4: ${1.4E4+1.4}<br> 10
mod 4: ${10 mod 4}<br>
15 div 3: ${15 div 3}<br>
</body>
```

d. ELLogicalOperator.jsp

```
<body>
<%-- LogicalOperator --%> <h2>Logical Operator</h2> true and true: ${true and true}<br> true && false: ${true && false}<br> true or true: ${true or true}<br> true || false: ${true || false}<br> not true: ${not true}<br> !false: ${!false}
</body>
```

e. ELRelationalOperator.jsp

```
<body>
<%-- RelationalOperator --%>
<h2>Relational Operator</h2>
10.0==10: ${10.0==10} <br>
10.0 eq 10: ${10.0 eq 10} <br>
((20*10)!= 200): ${((20*10)!= 200)} <br>
3 ne 3: ${3 ne 3} <br>
3.2>=2: ${3.2>=2} <br>
3.2 ge 2: ${3.2 ge 2} <br>
2<3: ${2<3} <br>
4 lt 6: ${4 lt 6} <br>
2 <= 4: ${2 <= 4} <br>
4 le 2: ${4 le 2}
</body>
```

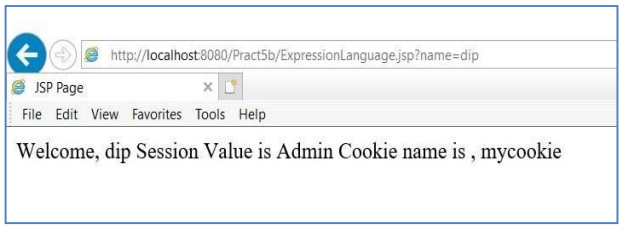
f. ELconditional op

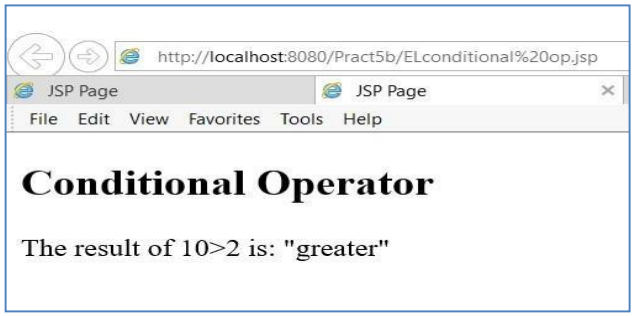
```
<body>
<h2>Conditional Operator</h2>

The result of 10>2 is: "${(10>1)?'greater':'lesser'}"
</body>
```

g. Empty Operator

```
<H1>Empty Operator Example</H1>
The Value for the Empty operator is:: ${empty "txxt"}
```

OUTPUT:





Q.5 c) Create a JSP application to demonstrate the use of JSTL.

CODE:

index.html

```
<html><body>

<a href=”setDemo.jsp”> SetDemo</a>

<a href =”Maxif.html”> MaxIF</a>

<a href=”forEachDemo.jsp”> ForEachDemo</a>

<a href=”outDemo.jsp”>outDemo</a>

<a href=”outDemo.jsp”> URLDemo</a>

<a href=”URLDemo.jsp”> URLDemo</a>

<a href=”choose_when_otherwise.jsp”>choose_when_otherwise</a>

</body></html>
```

setDemo.jsp

```
<%@taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
<c:set var="pageTitle" scope="application" value="Dukes Soccer League:

Registration" />

 ${pageTitle}
```

Maxif.html

```
<form  action  ="IFDemo.jsp">          x=<input type="text"  name="x"

/><br>    y=<input type="text" name="y"  /><br>

  <input type="submit" value="Check Max" />

  </form>
```

IFDemo.jsp

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<c:set var="x" value="${param.x}"/> <c:set var="y"

value="${param.y}"/>

<c:if test="${x>y}">

   <font color="blue"><h2>The Ans is:</h2></font>

   <c:out value="${x} is greater than ${y}"/>

</c:if>
```

ForeachDemo.jsp

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<c:forEach begin="1" end="10" var="i">

 The Square of <c:out value=" ${i}=${i*i}"/><br>

   </c:forEach>
```

outDemo.jsp

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<c:set var="name" value="John"/>

My name is: <c:out value= "${name}" />
```

URLDemo.jsp

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<c:url value="/index.html"/>
```

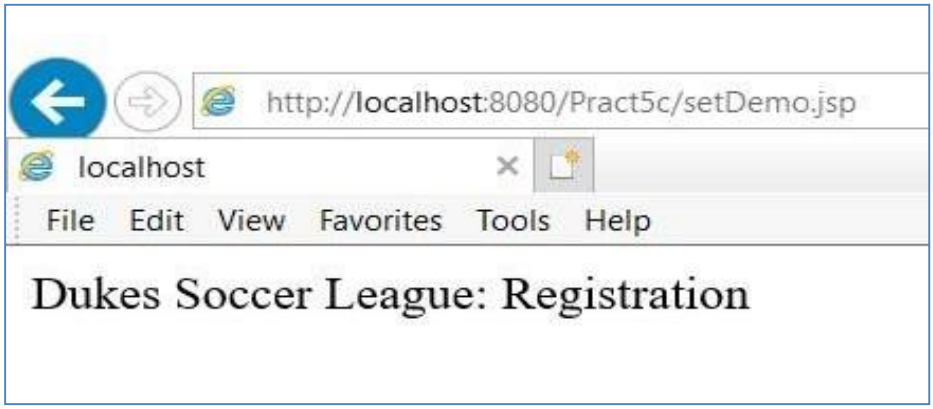
choose_when_otherwise.jsp

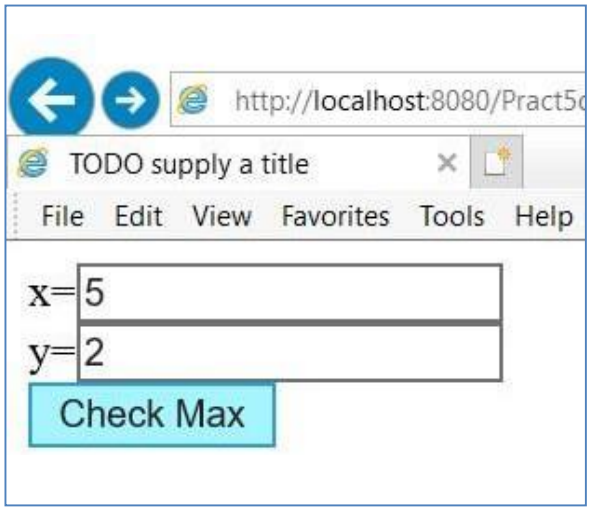
```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<c:set var="income" value="${4000*4}"/>

Your Income is: <c:out value="${income}"/>

<c:choose>
```

```
<c:when test="${income <=1000}">
Income is not good
</c:when>
<c:when test="${income > 10000}">
Income is Very Good
</c:when>
<c:otherwise>
Income is undetermined
</c:otherwise>
</c:choose>
```

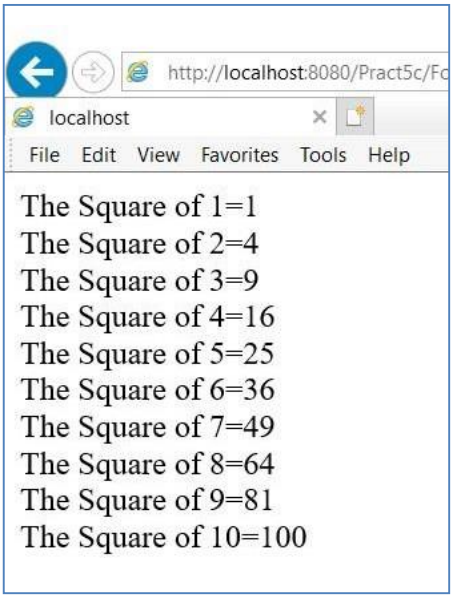
OUTPUT:


http://localhost:8080/Pract5c/index.html
localhost
File Edit View Favorites Tools Help
SetDemo MaxIF ForEachDemo OutDemo URLDemo choose_when_otherwise

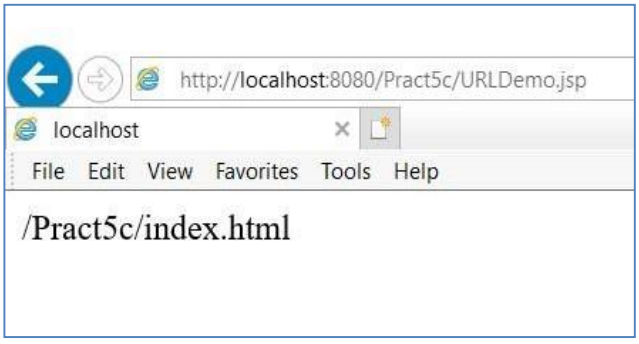

http://localhost:8080/Pract5c/setDemo.jsp
localhost
File Edit View Favorites Tools Help
Dukes Soccer League: Registration

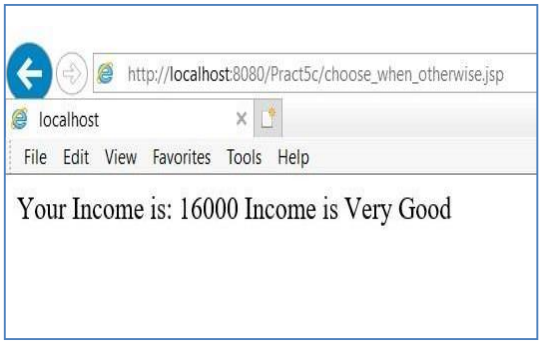

http://localhost:8080/Pract5c
TODO supply a title
File Edit View Favorites Tools Help
x=5
y=2
Check Max


http://localhost:8080/Pract5c/IFD
localhost
File Edit View Favorites Tools Help
The Ans is:
5 is greater than 2


http://localhost:8080/Pract5c/Fo
localhost
File Edit View Favorites Tools Help
The Square of 1=1
The Square of 2=4
The Square of 3=9
The Square of 4=16
The Square of 5=25
The Square of 6=36
The Square of 7=49
The Square of 8=64
The Square of 9=81
The Square of 10=100


http://localhost:8080/Pract5c/out
localhost
File Edit View Favorites Tools Help
My name is: John


http://localhost:8080/Pract5c/URLDemo.jsp
localhost
File Edit View Favorites Tools Help
/Pract5c/index.html


http://localhost:8080/Pract5c/choose_when_otherwise.jsp
localhost
File Edit View Favorites Tools Help
Your Income is: 16000 Income is Very Good

PRACTICAL 6

Q.6 a) Create a Currency Converter application using EJB.

CODE:

Index.html

```
<html><head><title>Currency Converter</title></head>
  <body>
    <form action="CCServlet" >
      Enter Amount <input type="text" name="amt"><br>
      Select Conversion Type
<input type="radio" name="type" value="r2d" checked>Rupees to Dollar
<input type="radio" name="type" value="d2r" >Dollar to Rupees<br>
      <input type="reset" ><input type="submit" value="CONVERT" >
    </form>
  </body> </html>
```

Step 2 : Create a session bean named as *CCBean* in the package named *mybeans*. Select the option Stateless and click on Local Interface.

Here you will find two files created in the mybeans package named as *CCBean.java* *and CCBeanLocal.java* CCBeanLocal.java package mybeans; import javax.ejb.Stateless;

@Stateless public interface

CCBeanLocal { //default constructor

public double r2Dollar(double r);

public double d2Rupees(double d); }

CCBean.java

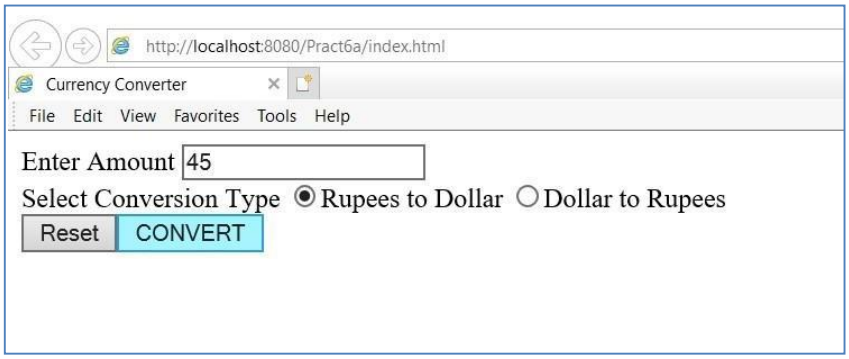
```
package mybeans; import
javax.ejb.Stateless;
@Stateless public class CCBean implements
CCBean1Local
{ public double r2Dollar(double
r)
{    return
r/65.65;
} public double d2Rupees(double
d)
{    return d*65.65;
}
}
```

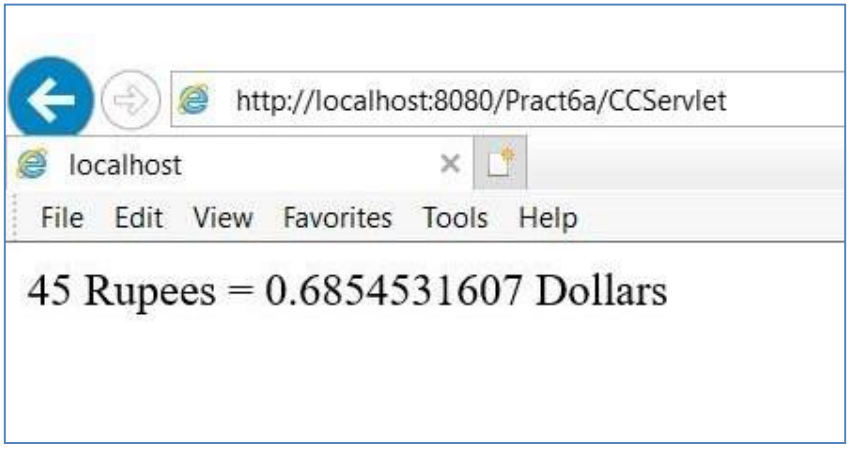
Step 3: Create a Servlet file name CCServlet in the package mypack.

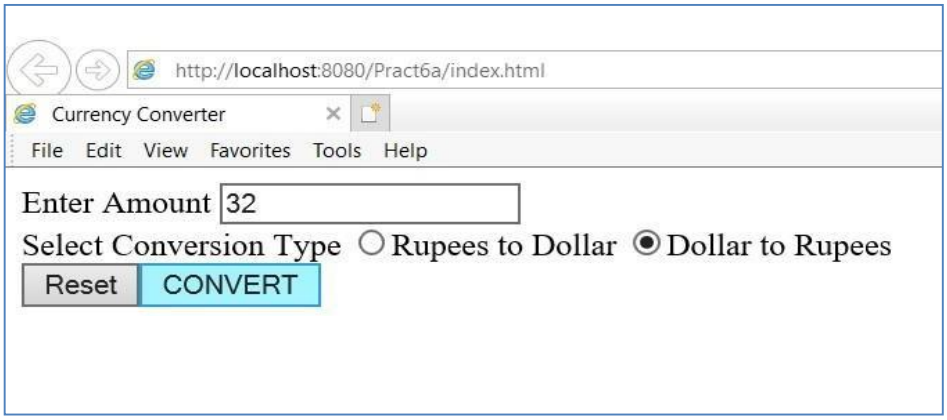
```
package mypack; import java.io.*; import
javax.servlet.*; import javax.servlet.http.*;
import javax.ejb.EJB; import
mybeans.CCBeanLocal; public class
CCServlet extends HttpServlet {
  @EJB CCBeanLocal obj;
public void doGet(HttpServletRequest request, HttpServletResponse response)throws
ServletException, IOException
    { response.setContentType("text/html;charset=UTF-8"); PrintWriter out =
    response.getWriter();
```

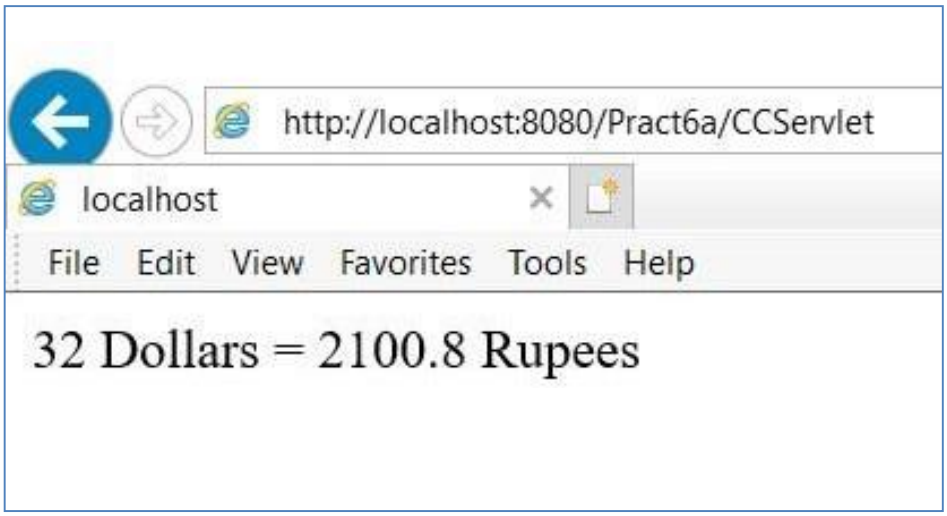
```
        double amt = Double.parseDouble(request.getParameter("amt"));
       if(request.getParameter("type").equals("r2d"))
{ out.printIn("<h1>"+amt+"Rupees ="+obj.r2Dollar(amt)+"
Dollars</h1>");
}
if(request.getParameter("type").equals("d2r"))
{ out.println("<h1>"+amt+ " Dollars = "+obj.d2Rupees(amt)+"
Rupees</h1>");
}
}
}
```

OUTPUT:









Q.6 b) Develop a Simple Room Reservation System Application Using EJB.

CODE:

Index.html

```
<html>
  <head>
    <title>Room Reservation</title>
  </head>
<body>
<form method=”post” action =”RoomClient”>
<br> No. of Rooms<input type=text name=”t1”>
<br> <input type=”submit” name=”btn” value=”CheckIN”>
```


 <input type=”submit” name=”btn” value=”CheckOUT”>

</form>

</body>

</html>

Step2: Create a session bean named as *RoomBean* in the package named *ejb*. Select the option Stateless and click on Local Interface.

Here you will find two files created in the ejb package named as *RoomBean.java and RoomBeanLocal.java*

RoomBeanLocal.java

```
package ejb; import

javax.ejb.Local;

@Local public interface

RoomBeanLocal {

      public int checkin(int no);

public int checkout(int no);

}
```

RoomBean.java

```
package ejb; import javax.ejb.Stateless; import

java.sql.*; @Stateless public class RoomBean

implements RoomBeanLocal { public int checkin(int

no) {

   try
   {
     Class.forName("com.mysql.jdbc.Driver");
     Connection
con=DriverManager.getConnection("jdbc:mysql://localhost/roomdb","root","tiger");
     String sql1 = "select * from room";
```

```
    Statement st=con.createStatement();
ResultSet rs=st.executeQuery(sql1);
    rs.next();      int
total=rs.getInt(1);      int
occ=rs.getInt(2);      int
free=total-occ;
    System.out.println(total);
System.out.println(free);      if
(free>=no)
    {
      String sql2="update room set occ=?";
      PreparedStatement ps=con.prepareStatement(sql2);
      ps.setInt(1, occ+no);
     int res=ps.executeUpdate();
return res;
}      else return 0;
  }
  catch(Exception e)
  {       return
0;
  }
  }
  public int checkout(int no) {
  try
```

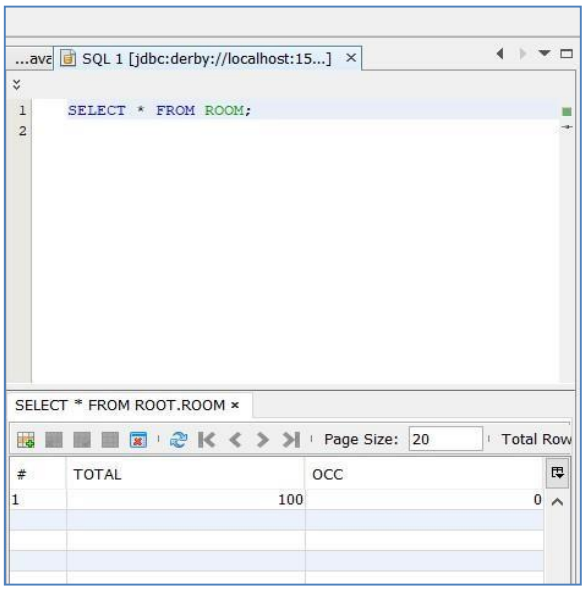
```
  { Class.forName("com.mysql.jdbc.Driver");
    Connection
con=DriverManager.getConnection("jdbc:mysql://localhost/roomdb","root","tiger");
    String sql1 = "select * from room";
    Statement st=con.createStatement();
ResultSet rs=st.executeQuery(sql1);
    rs.next();              int
total=rs.getInt(1);       int
occ=rs.getInt(2);          if
(occ>=no)
    {
     String sql2="update room set occ=?";
     PreparedStatement ps=con.prepareStatement(sql2);
     ps.setInt(1, occ-no);
     int res=ps.executeUpdate();
return res;
    }     else
return 0;
  }
  catch(Exception e)
  {       return
0;
  }
  }
}
```

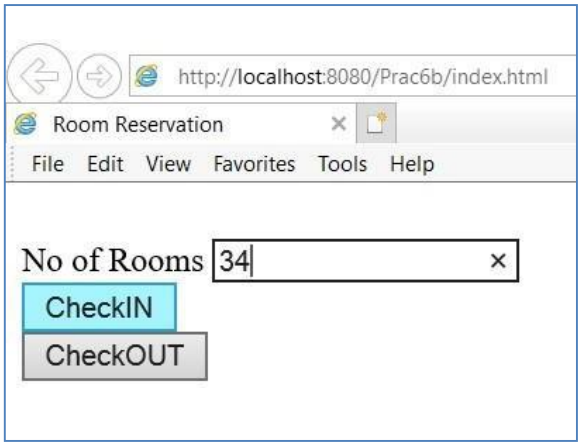
Step 3: Create a Servlet file named as RoomClient. Do not click on web.xml (Deployment Descriptor)

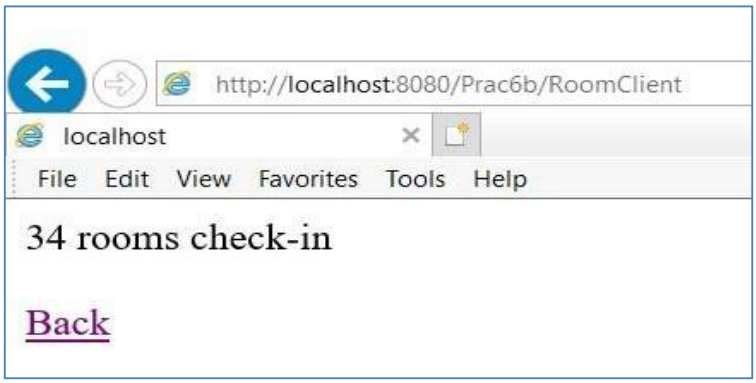
```
package servlet; import
ejb.RoomBeanLocal; import
java.io.*; import javax.ejb.EJB;
import javax.servlet.*; import
javax.servlet.http.*; import
javax.servlet.annotation.*;
@WebServlet(name = "roomclient", urlPatterns = {"/roomclient"})
public class roomclient extends HttpServlet {
   @EJB RoomBeanLocal obj;     protected void doPost(HttpServletRequest
request, HttpServletResponse response)           throws ServletException,
IOException {       response.setContentType("text/html");       PrintWriter out =
response.getWriter();
      try     {                                                    int
no=Integer.parseInt(request.getParameter("t1"));
String b=request.getParameter("btn");

            int res=0;
        if(b.equals("CheckIN"))
        {
          res=obj.checkin(no);            if
(res==1)            out.println(no + " rooms
check-in");
        }
        if(b.equals("CheckOUT"))
        {
```

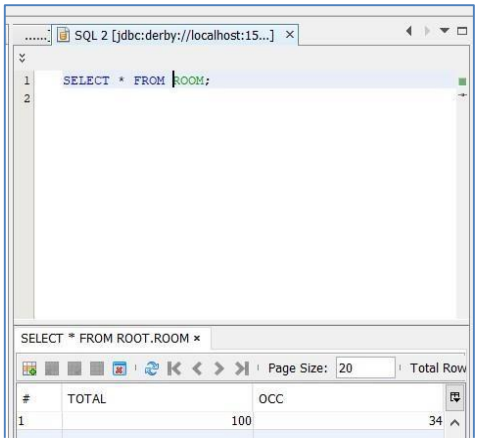
```
            res=obj.checkout(no);              if
(res==1)              out.println(no + " rooms
check-out");
         }                  if(res==0)              out.println("Not possible to
do Check IN / OUT");              out.println("<br><br><a
href=index.html> Back </a>");
       }            finally {
out.close();
       }
    }
 }
```

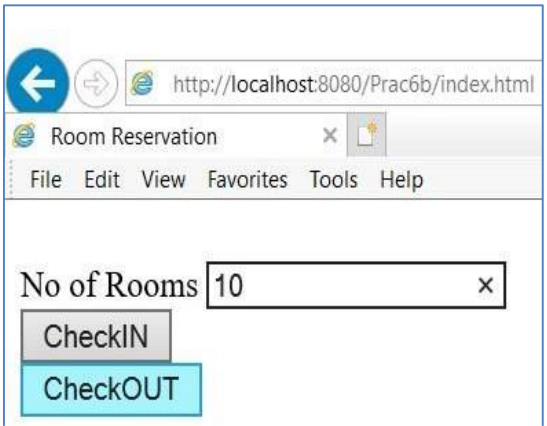
OUTPUT:


SQL 1 [jdbc:derby://localhost:15...]
SELECT * FROM ROOM;
SELECT * FROM ROOT.ROOM ×
Page Size: 20
Total Row
TOTAL OCC
1 100 0


http://localhost:8080/Prac6b/index.html
Room Reservation
File Edit View Favorites Tools Help
No of Rooms 34
CheckIN
CheckOUT


http://localhost:8080/Prac6b/RoomClient
localhost
File Edit View Favorites Tools Help
34 rooms check-in
Back


SQL 2 [jdbc:derby://localhost:15...]
SELECT * FROM ROOM;
SELECT * FROM ROOT.ROOM ×
Page Size: 20
Total Row
TOTAL OCC
1 100 34


http://localhost:8080/Prac6b/index.html
Room Reservation
File Edit View Favorites Tools Help
No of Rooms 10
CheckIN
CheckOUT

Q.6 c) Develop simple shopping cart application using EJB [Stateful Session Bean].

CODE:

Step 1 creating application

File -> new project-> java web->web application -> Prac6CShoppingCartApp -> select Use dedicated folder for storing libraries -> finish

Step 2: Creating a stateful session bean

Source package -> new -> other -> enterprise java beans -> session bean -> next -> new session bean -> ejb name: ->ShoppingCart -> package: -> ejb -> session type option -> Stateful -> finish. ShoppingCart.java

```
package ejb; import

java.sql.*; import

java.util.*; import

javax.ejb.*; @Stateful

public class ShoppingCart

{   List<String> contents;    String

customerName;    private Connection

conn = null;    private ResultSet rs;

private Statement stmt = null;

private String query = null;    public

void initialize(String person)

  { if (person != null) {

customerName = person;

        try {

           Class.forName("com.mysql.jdbc.Driver").newInstance();
```

```
            conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/cartdb", "root",
"tiger");

        } catch(ClassNotFoundException | IllegalAccessException | InstantiationException |
SQLException e) {

  System.err.println("Sorry failed to connect to the Database." + e.getMessage());

        }

      }

      contents = new ArrayList<>();

   }

   public void addBook(String title) {

      try {              stmt = conn.createStatement();              query = "INSERT INTO cart

VALUES('" + customerName + "','" + title + "')";           stmt.executeUpdate(query);

      } catch(SQLException e) {

         System.err.println("Sorry failed to insert values from the database table. " +
e.getMessage());

      }

   }

   public void removeBook(String title) {

      try {           stmt =

conn.createStatement();

         query = "DELETE FROM cart WHERE UserName='" + customerName + "' AND

ItemName='" + title + "'";           stmt.executeUpdate(query);

      } catch(SQLException e) {

         System.err.println("Sorry failed to delete values from the database table. " +
e.getMessage());

      }

   }

   public List<String> getContents() {
```

```
    try {           stmt = conn.createStatement();          query = "SELECT * FROM cart
WHERE UserName='" + customerName + "'";          rs = stmt.executeQuery(query);
while(rs.next()) {          contents.add(rs.getString("ItemName"));
      }
    } catch(SQLException e) {
      System.err.println("Sorry failed to select values from the database table. " + e.getMessage());
    }      return
contents;
  }
  @Remove()    public
void remove() {
contents = null;
  }
}
```

Step 3: creating a web client using index.jsp

Right click on wewb pages -> new -> JSP -> filename -> index -> finish.

```
<%@page import="java.util.Iterator, java.util.List, javax.naming.InitialContext,
ejb.ShoppingCart"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<%!
  private static ShoppingCart cart;
  public void jspInit() {
    try {
      InitialContext ic = new InitialContext();          cart = (ShoppingCart)
ic.lookup("java:global/Prac6CShoppingCartApp/ShoppingCart");
```

```
      } catch (Exception ex) {
         System.out.println("Could not create cart bean." + ex.getMessage());
      }
   }
%>
<%
   if(request.getParameter("txtCustomerName") != null) {
cart.initialize(request.getParameter("txtCustomerName"));
   } else {
cart.initialize("Guest");
   }
   if (request.getParameter("btnRmvBook") != null) {
String books[] = request.getParameterValues("chkBook");
if (books != null) {           for (int i=0; i<books.length; i++) {
cart.removeBook(books[i]);
         }
      }
   }
   if (request.getParameter("btnAddBook") != null) {
String books[] = request.getParameterValues("chkBook");
if (books != null) {

         for (int i=0; i<books.length; i++) {
cart.addBook(books[i]);
         }
      }
```

```
  }
%>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Shopping Cart</title>
  </head>
  <body style="background-color: pink;">
    <h1 style="text-align: center;">Books For Sale</h1><br>
    <form method="post">
      Customer Name: <input type="text" name="txtCustomerName" value=<%=
request.getParameter("txtCustomerName")%> /><br>

            <b>Book Titles</b><br>
              <input type="checkbox" name="chkBook" value="Struts 2.0 For Beginners">Struts
2.0 For Beginners<br>

              <input type="checkbox" name="chkBook" value="Oracle 11g For
Professionals">Oracle 11g For Professionals<br>
              <input type="checkbox" name="chkBook" value="Hibernate 3 For
Beginners">Hibernate 3 For Beginners<br>
              <input type="checkbox" name="chkBook" value="Java Persistence API In EJB 3 For
Beginners">Java Persistence API In EJB 3 For Beginners<br>

              <br>
              <input type='submit' value='Add To My Basket' name='btnAddBook'>
              <input type='submit' value='Remove From My Basket'
name='btnRmvBook'><br><br><br>

        <%
if(cart!=null)
        {
```

```
            out.print("<b>Basket</b><br>");
            List<String> bookList = cart.getContents();
Iterator iterator = bookList.iterator();                    while
(iterator.hasNext())
            {
               String title = (String) iterator.next();
        %>
        <%= title %><br>
        <%
            }
        }
        %>
    </form>
  </body>
</html>
```

Step 4:

Create database and database table

Services -> create database -> cartdb ->select cartdb - > right click -> create table -> cart -> UserName varchar 35

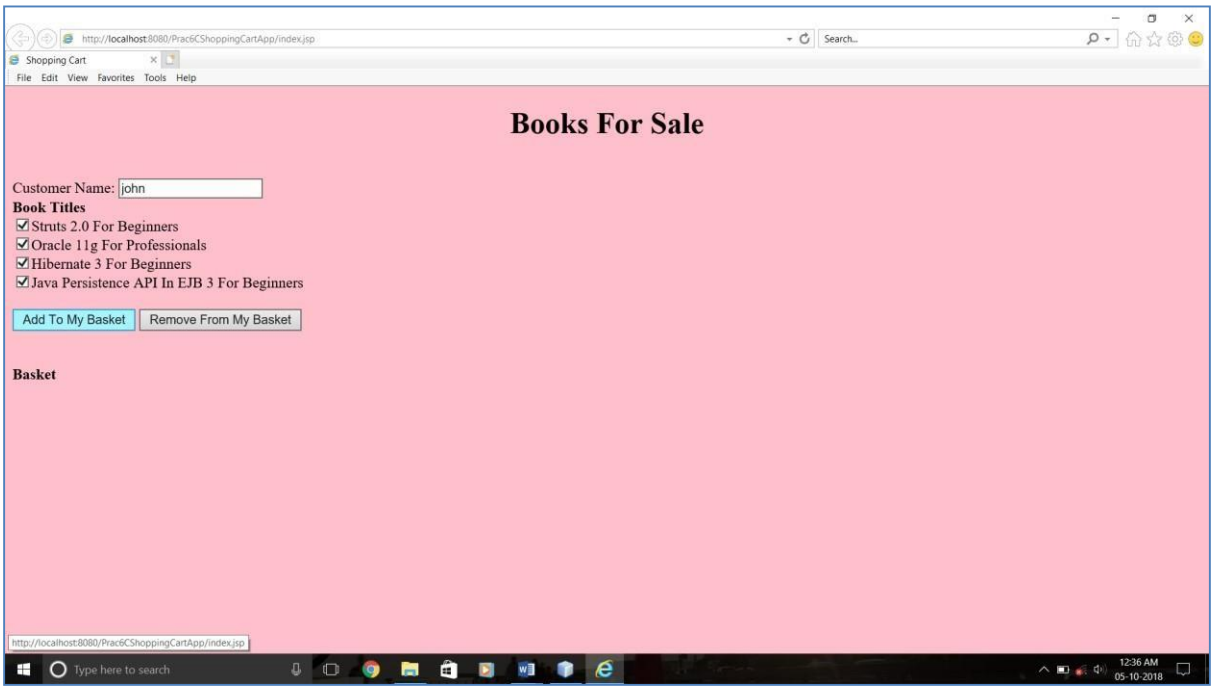
ItemName varchar 50 Finish.

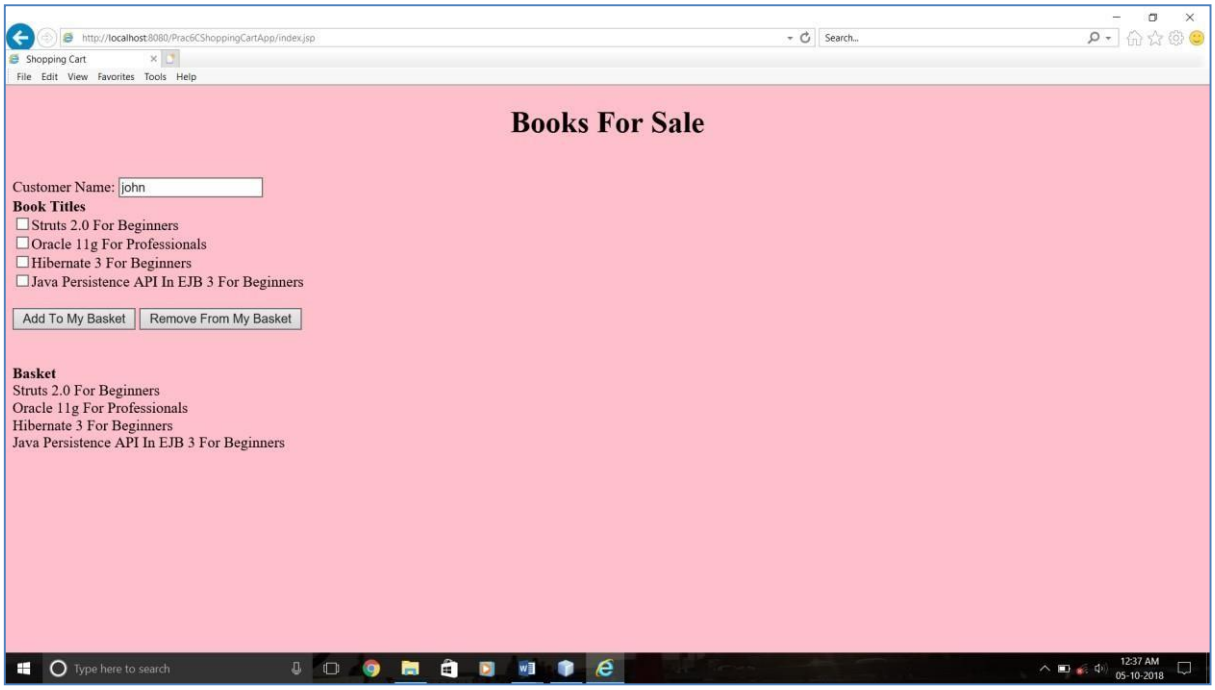
Step 5.

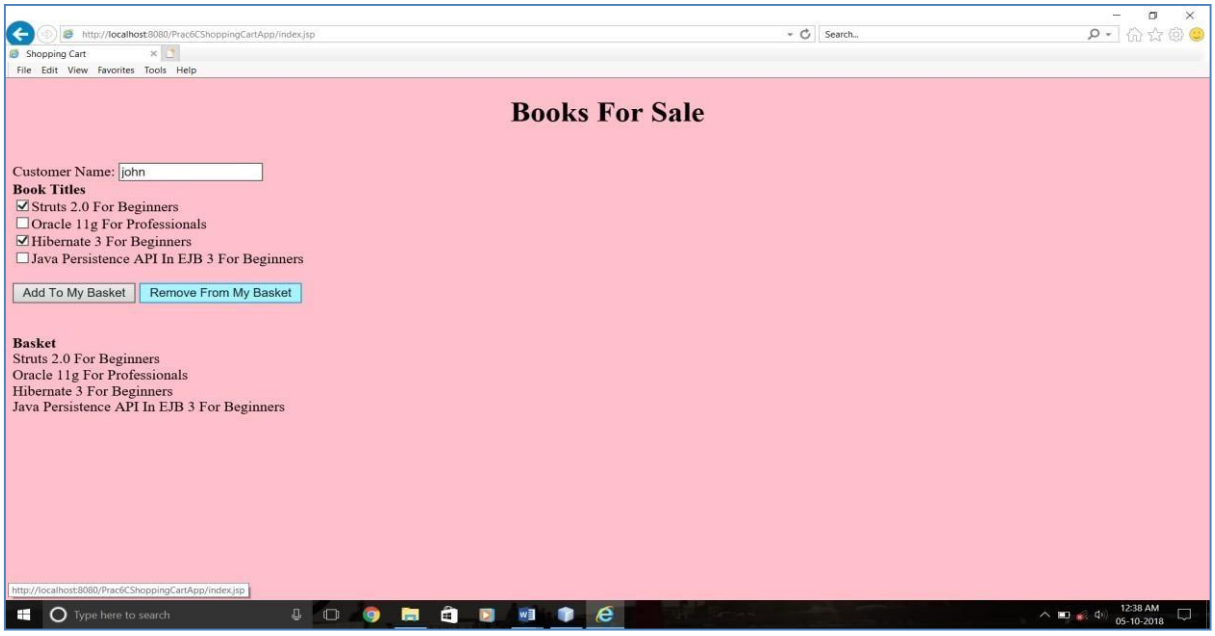
Add mysql connector to the library under project tab.

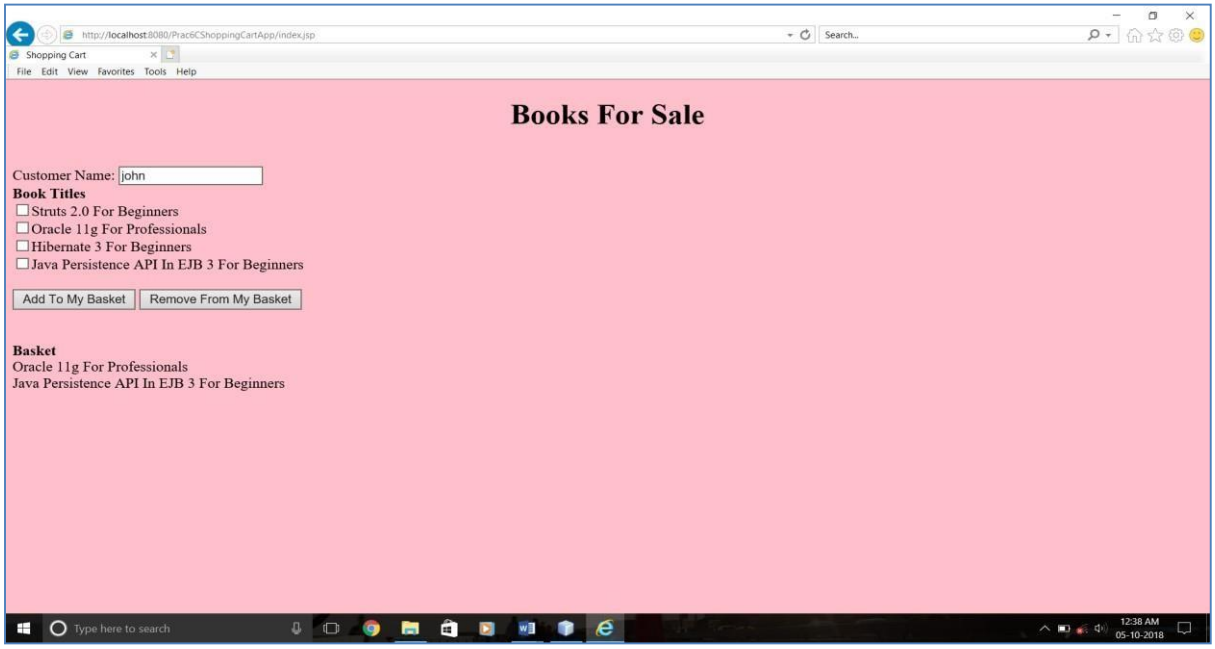
Step 6: build and run the application.

OUTPUT:


http://localhost:8080/Prac6CShoppingCartApp/index.jsp
Search...
Shopping Cart
File Edit View Favorites Tools Help
Books For Sale
Customer Name: john
Book Titles
Struts 2.0 For Beginners
Oracle 11g For Professionals
Hibernate 3 For Beginners
Java Persistence API In EJB 3 For Beginners
Add To My Basket
Remove From My Basket
Basket
http://localhost:8080/Prac6CShoppingCartApp/index.jsp
Type here to search
12:36 AM
05-10-2018


http://localhost:8080/Prac6CShoppingCartApp/index.jsp
Search...
Shopping Cart
File Edit View Favorites Tools Help
Books For Sale
Customer Name: john
Book Titles
Struts 2.0 For Beginners
Oracle 11g For Professionals
Hibernate 3 For Beginners
Java Persistence API In EJB 3 For Beginners
Add To My Basket
Remove From My Basket
Basket
Struts 2.0 For Beginners
Oracle 11g For Professionals
Hibernate 3 For Beginners
Java Persistence API In EJB 3 For Beginners
Type here to search
12:37 AM
05-10-2018





PRACTICAL 7

Q.7 a) Develop simple EJB application to demonstrate Servlet Hit count using Singleton Session Beans.

CODE:

Java Web-> web application -> Pract7AServletHitsSingltonApp -> finish.

Step 1: Index.html

```
<html>
<head>

<title>TODO supply a title</title>

<meta charset=”UTF-8”>

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

    <meta http-equiv="Refresh" content="0; URL=ServletClient">
  </head>
  <body>
    <div>TODO write content</div>
  </body>
</html>
```

Step2: Create a Session Bean named as CountServletHitsBean Select Singleton package name as ejb (do not select Local or Remote)

```
package ejb; import
javax.ejb.Singleton;
@Singleton
public class CountServletHitsBean {
private int hitCount;
  public synchronized int getCount()
  {
  return hitCount++;
  }
}
```

Step 3: Create a Servlet File name ServletClient in the package name as servlet. Do not select the Deployment Discriptor file.

```
package servlet;
import ejb.CountServletHitsBean;
import.java.io.*; import
```

```
javax.ejb.EJB;
import javax.servlet.ServletException; import
javax.servlet.annotation.WebServlet; import
javax.servlet http*;
@WebServlet(name=”ServletClient”, ur1patterns urlPatterns =
{"/ServletClient"}) public class ServletClient extends HttpServlet {
@EJB CountServletHitsBean obj;
@Override
protected void service (HttpServletRequest req, HttpServletResponse res) throws
ServletException, IOException
{
   res.setContentType("text/html");                PrintWriter out=res.getWriter();
out.print("<b>Number of times this Servlet is accessed </b>: "+obj.getCount());
}
}
```

OUTPUT:

Q.7 b) Develop simple visitor Statistics application using Message Driven Bean [Stateless Session Bean].

CODE:

Web-> web application -> Pract7BVisitorStatisticsMDBApp -> select dedicated folders for storing libraries -> finish.

Step 1: index.jsp

```
<%@page import="javax.jms.JMSException"%>
<%@page import="javax.naming.InitialContext"%>
<%@page import="javax.jms.Connection"%>
<%@page import="javax.jms.TextMessage"%>
<%@page import="javax.jms.MessageProducer"%>
<%@page import="javax.jms.Session"%>
<%@page import="javax.jms.Queue"%>
<%@page import="javax.jms.ConnectionFactory"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html> <%!
private static
ConnectionFactory
connection factory;
private static Queue
queue; Connection=null;
Session mySession=null;
%>
MessageProducer messageProducer=null;
TextMessage message=null;
%>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
</head>
<body>
Welcome to My Home Page
<%
try{
InitialContext ic=new InitialContext();
```

```
queue = (Queue)ic.lookup("jms/Queue");
connectionFactory= (Connection Factory) ic.lookup("jms/QueueFactory");
connection = connectionFactory.createConnection( );
mySession= connection.createSession(false, Session.AUTO_ACKNOWLEDGE);
messageProducer= mySession.createProducer(queue);
message=mySession.createTextMessage();
message.setText(request.getRemoteAddr());          messageProducer.send(message);
      }
      catch(JMSException e)
     {
      System.out.println("Exception Occoured "+e.toString());
      }
      %>
   </body>
</html>
```

Step2: Create a Database name visitorstat Create table name userstat column names
Firstvisitdt – timestamp
Hostname – varchar 30 Primary Key
Visits – int

Step3: Create a Session Bean named as VisitorStatBean Select Stateless package name as ejb, do not select Local / Remote

```
package ejb; import java.sql.*; import
javax.annotation.PostConstruct; import
javax.annotation.PreDestroy; import
javax.ejb.Stateless;
@Stateless public class
VisitorStatBean { private
Connection conn=null; private
ResultSet rs; private Statement
st=null; private String query
=null; @PostConstruct public
void connect()
```

```
{    try {
       Class.forName("com.mysql.jdbc.Driver").newInstance();
       conn=DriverManager.getConnection("jdbc:mysql://localhost/visitorstat", "root", "tiger");
    }
    catch (Exception e) {
       System.err.println(e.getMessage());
    }
}

@PreDestroy public void
disconnect()
{    try {
       conn.close();
    } catch (Exception e) {
        System.err.println(e.getMessage());
    }
} public void addVisitor(String
host)
{ try { st= conn.createStatement(); query="insert into userstat
(hostname,visits) values ('"+host+"','1')";
st.executeUpdate(query);
    }
    catch (SQLException e)
    {       try {          st=conn.createStatement(); query="update userstat
set visits=visits+1 where hostname='"+host+"' ";
st.executeUpdate(query);


       }
       catch (SQLException ex) {
              System.err.println("Cannot Update"+e.getMessage());
       }

    }
}
```

}

Step 4: Right click on Source Packages Select New Other Enterprise Java Bean MessageDrivenBean EJB Name: BasicMessageBean Package: ejb Select Project Destination Click on Add Button Destination Name: jms/Queue Destination Type select the option Queue click on OK Click on Next Activation Configuration Properties should be as it is. Click on Finish

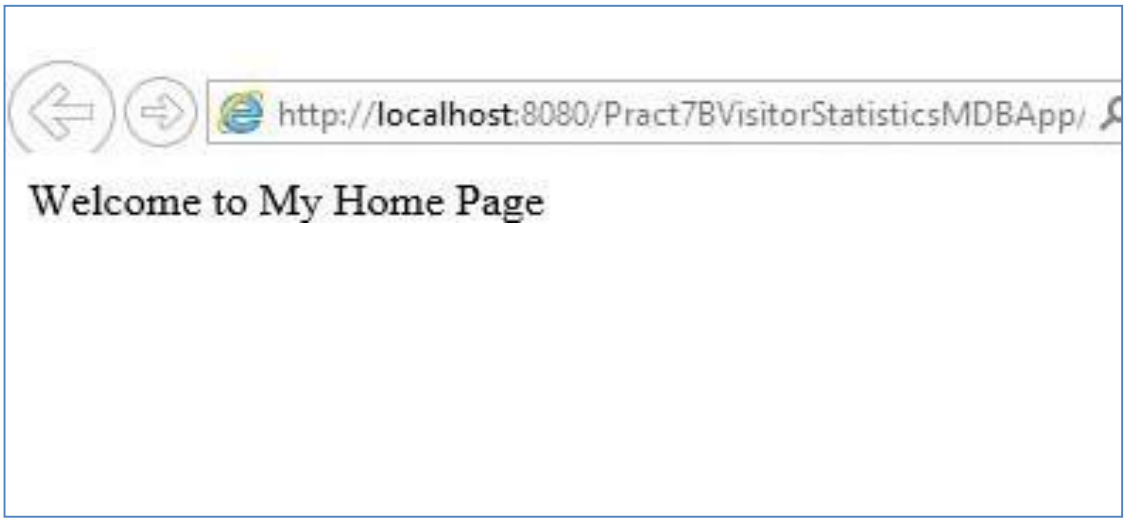
```
package ejb;
import javax.annotation.Resource; import
javax.ejb.ActivationConfigProperty; import
javax.ejb.EJB; import
javax.ejb.MessageDriven; import
javax.ejb.MessageDrivenContext; import
javax.jms.JMSException; import
javax.jms.Message; import
javax.jms.MessageListener; import
javax.jms.TextMessage;

@MessageDriven(activationConfig = {
   @ActivationConfigProperty(propertyName = "destinationLookup", propertyValue =
"jms/Queue"),
   @ActivationConfigProperty(propertyName = "destinationType", propertyValue =
"javax.jms.Queue")
}) public class BasicMessageBean implements
MessageListener {
@EJB VisitorStatBean vs;

@Resource private
MessageDrivenContext mdc;
public BasicMessageBean() {
   }
```

```
    @Override
    public void onMessage(Message message) {
try {
          if(message instanceof TextMessage){
             TextMessage msg= (TextMessage) message;
vs.addVisitor(msg.getText());
          }
       }
       catch (JMSException e) {
mdc.setRollbackOnly();
       }
    }


}
```

Step 5:
Before deploying and running the application, Glassfish Server setting is required.
Browse the path:
Localhost:4848 on any browser.
Find Resources -> connectors -> Connector Resources double click on Connector Resources -> click on 'New' Button -> write JNDI name as -> jms/QueryFactory.

Find Admin Object Resources and double click on that -> click on 'New' Button -> write JNDI name as -> jms/Queue.

Now run index.jsp file.
OUTPUT:



Q.7 c) Develop simple Marks Entry Application to demonstrate accessing Database using EJB.

CODE:
Step 1:

Create web application as pract7CMarksApp.

Step 2:

Create database marksdb Step

3:

Create tables marks in marksdb database as:

create table marks (id int primary key auto_increment, sname varchar(35), marks1 int, marks2 int, marks3 int); step 4: **index.jsp**

```
<%@page import="ejb.MarksEntryBean"%>
<%@page import="javax.naming.InitialContext"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<%!
private static MarksEntryBean obj; public
void jspInit()
{
  try
   {
    InitialContext ic=new InitialContext();
obj=(MarksEntryBean)ic.lookup("java:global/Pract7CMarksApp/MarksEntryBean");
  }
  catch(Exception e)
  {
     System.out.println(e);
```

```
   }
}
%>
<%
    if(request.getParameter("InsertMarks")!=null)
    {
       String sname;          int marks1, marks2, marks3;
 sname = request.getParameter("sname");

marks1=Integer.parseInt(request.getParameter("m1"));

marks2=Integer.parseInt(request.getParameter("m2"));

marks3=Integer.parseInt(request.getParameter("m3"));

obj.addMarks(sname,marks1,marks2,marks3);

out.print("Marks entered successfully..!!!!");

    }
%>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>JSP Page</title>
   </head>
   <body>
      <h2>Enter Details</h2>
      <form name="result" method="post">
         Enter student's name: <input type='text' name="sname" /><br>
         Enter subject 1 marks: <input type='text' name="m1" /><br>
         Enter subject 2 marks: <input type='text' name="m2" /><br>
         Enter subject 3 marks: <input type='text' name="m3" /><br>
```

```
        <input type='submit' name="InsertMarks" /><br>
      </form>
   </body>
</html>
```

Step 4:

create stateful java bean as select source package -> session bean -> class name -> MarksEntryBean -> package -> ejb -> bean type-> stateful -> don't select Local / Remote -> finish.

```
package ejb; import java.sql.*; import javax.ejb.Stateful;

@Stateful public class

MarksEntryBean { String

sname; int m1,m2,m3;

Connection con=null;

Statement st=null; String query=""; public void

addMarks(String sname,int m1,int m2,int m3)

{
   try
   {
      Class.forName("com.mysql.jdbc.Driver");

con=DriverManager.getConnection("jdbc:mysql://localhost:3306/marksdb", "root","tiger");

st=con.createStatement();

      query="insert into marks (sname,marks1,marks2,marks3) values
('"+sname+"','"+m1+"','"+m2+"','"+m3+"')";

st.executeUpdate(query);

      System.out.print("Marks entered sucessfully!!");
   }
```

```
    catch(Exception e){System.out.println(e);}
}
}
```

OUTPUT: