PRACTICAL 1

SIMPLE SERVLET APPLICATIONS.

Q.1 A) CREATE A SIMPLE CALCULATOR APPLICATION USING SERVLET.

CODE:

INDEX.HTML

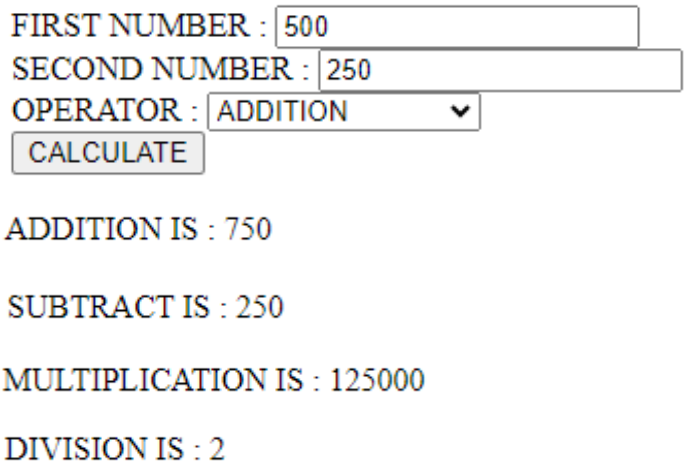
```
<html>
    <head>
        <title>CALCULATOR</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
       <form action="calculator"/>
       FIRST NUMBER : <input type="Textbox" name="fno"/>
       <br>
       SECOND NUMBER : <input type="Textbox" name="sno"/>
       <br>
       OPERATOR : <select name="opt">
           <option value="addition">ADDITION</option>
           <option value="subtractor">SUBTRACTION</option>
           <option value="multiplication">MULTIPLICATION</option>
           <option value="division">DIVISION</option>
       </select>
       <br>
       <input type="submit" value="CALCULATE"/>
       </form>
    </body>
</html>
```

CALCULATOR.JAVA

```
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;

protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            int n3;
            int n1=Integer.parseInt(request.getParameter("fno"));
            int n2=Integer.parseInt(request.getParameter("sno"));
            String op=request.getParameter("opt");
            if(op.equals("addition"))
            {
                n3=n1+n2;
                out.println("ADDITION IS : "+n3);

            }
            if(op.equals("subtractor"))
            {
                n3=n1-n2;
                out.println("SUBTRACT IS : "+n3);
            }
            if(op.equals("multiplication"))
            {
                n3=n1*n2;
                out.println("MULTIPLICATION IS : "+n3);

            }
            if(op.equals("division"))
            {
                n3=n1/n2;
                out.println("DIVISION IS : "+n3);

            }
```

```
        }
    }
```

OUTPUT:


FIRST NUMBER : 500
SECOND NUMBER : 250
OPERATOR : ADDITION
CALCULATE
ADDITION IS : 750
SUBTRACT IS : 250
MULTIPLICATION IS : 125000
DIVISION IS : 2

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:

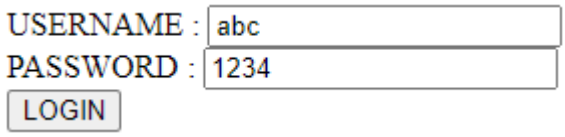
INDEX.HTML

```
<html>
    <head>
        <title>LOGIN PAGE </title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <form action="loginpage"/>
        USERNAME : <input type="Textbox" name="uname"/>
        <br>
        PASSWORD : <input type="Textbox" name="pass"/>
        <br>
        <input type="submit" value="LOGIN"/>
    </form>
    </body>
</html>
```

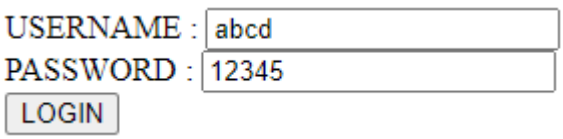
LOGINPAGE.JAVA

```
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;

protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            String u=request.getParameter("uname");
           String p=request.getParameter("pass");
           if(u.equals("abc")&& p.equals("1234"))
           {
               out.println("login successfully !!!");
           }
           else
           {
               out.println("login failed !!!");
           }
        }
    }
```

OUTPUT:



login successfully !!!



login failed !!!

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:

MYSQL COMMAND FROM MYSQL SOFTWARE: -

- Select **Services** → Expand **Databases** → Right click on **MySQL Server at localhost:3306[disconnected]** → Click on **Connect** → Enter Your **Password** → **OK.**
- Again, Right Click on **MySQL Server at localhost:3306** → Select **Create Database** → Enter Your Database Name and Select the **Checkbox** to grant permission.
- Right click on **Table** under your Database.
- Enter Your Table **Name** by Replacing **Untitled**. Click on **Add Column**, and Add Columns as below Mentioned:
  Name → **username**, Type → **Varchar**, Size → **50**, Select Checkbox of **Primary Key**
  Name → **password**, Type → **Varchar**, Size → **50**
  Name → **email**, Type → **Varchar**, Size → **50**
  Name → **country**, Type → **Varchar**, Size → **50**.
- Add **mysql-connector** in **Libraries** folder:
  Right click on **Libraries** folder Select Add **JAR/Folder.**
  mysql-connector-jar File Download Link
  (http://www.java2s.com/example/jar/m/download-mysqlconnectorjava8011jar-file.html#google_vignette)

INDEX.HTML

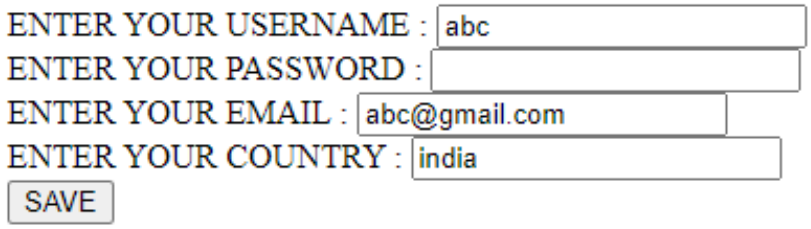
```
<html>
    <head>
        <title>REGISTRATION FORM</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
         <form action="registration">
            ENTER YOUR USERNAME : <input type="text" name="uname"/><br/>
            ENTER YOUR PASSWORD : <input type="password" name="pwd"/><br/>
            ENTER YOUR EMAIL : <input type="text" name="mail"/><br/>
            ENTER YOUR COUNTRY : <input type="text" name="cntry"/><br/>
                                <input type="submit" value="SAVE"/>
        </form>
    </body>
</html>
```

REGISTRATION.JAVA

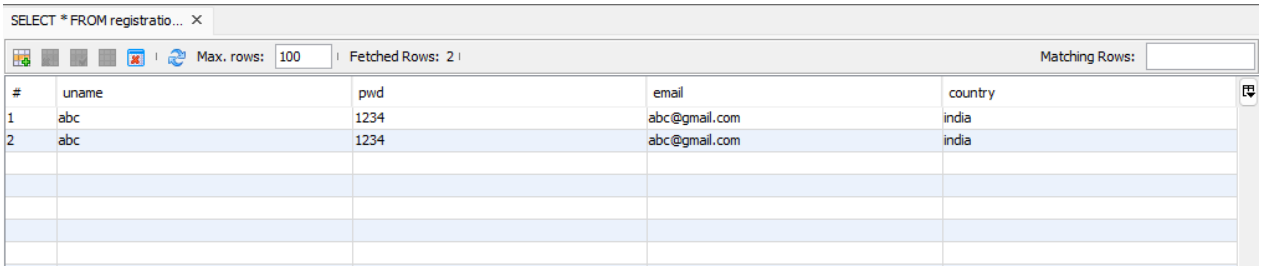
```
import java.sql.*;
import java.io.*;
import java.io.PrintWriter;
import javax.servlet.http.*;
import javax.servlet.*;

protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        String username=request.getParameter("uname");
        String password=request.getParameter("pwd");
        String email=request.getParameter("mail");
        String country=request.getParameter("cntry");
        Connection con=null;
        PreparedStatement pst=null;
        try
        {
            Class.forName("com.mysql.jdbc.Driver");
```

```
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/vinitx67?autoReco
nnect=true&useSSL=false","root","67Xvinit2021");
            out.println("CONNECTION DONE SUCCESSFULLY !!!  ");
            pst=con.prepareStatement("insert into registration_form
values(?,?,?,?)");
            pst.setString(1,username);
            pst.setString(2,password);
            pst.setString(3,email);
            pst.setString(4,country);
            pst.execute();
            out.println("DATA ADDED SUCCESSFULLY !!!");
        }
        catch(Exception e)
        {
            out.print(e);
        }
    }
```

OUTPUT:



CONNECTION DONE SUCCESSFULLY !!! DATA ADDED SUCCESSFULLY !!!



#	uname	pwd	email	country
1	abc	1234	abc@gmail.com	india
2	abc	1234	abc@gmail.com	india

PRACTICAL 2

SERVLET APPLICATIONS WITH COOKIES AND SESSIONS.

Q.2 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>
    <head>
        <title>LOGIN PAGE </title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <form action="requestdispatcher"/>
        USERNAME : <input type="Text" name="uname"/>
        <br>
        PASSWORD : <input type="Textbox" name="pass"/>
        <br>
        <input type="submit" value="LOGIN"/>
    </form>
    </body>
</html>
```

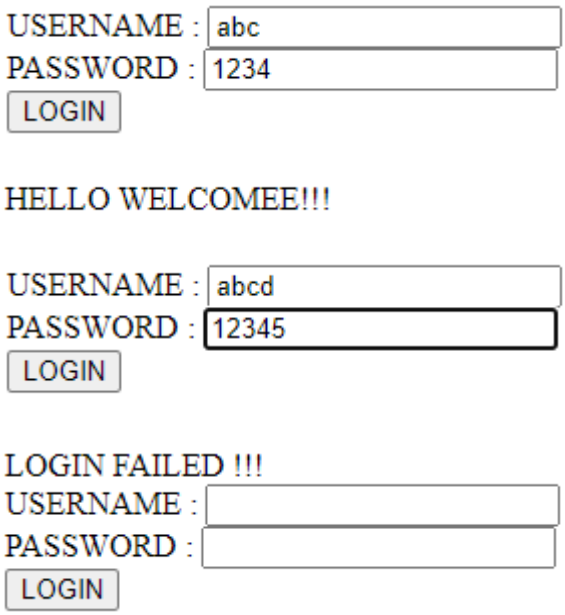
REQUESTDISPATCHER.JAVA

```
import javax.servlet.RequestDispatcher;

protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
           String u=request.getParameter("uname");
           String p=request.getParameter("pass");
           if(u.equals("abc")&& p.equals("1234"))
           {
               RequestDispatcher
rd=request.getRequestDispatcher("/welcomepage");
               rd.forward(request,response);

           }
           else
           {
               out.println("LOGIN FAILED !!!");
               RequestDispatcher
rd=request.getRequestDispatcher("/index.html");
               rd.include(request,response);
           }
        }
    }
```

WELCOMEPAGE.JAVA

```
protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            out.print("HELLO WELCOMEE!!!");
        }
    }
```

OUTPUT:


USERNAME : abc
PASSWORD : 1234
LOGIN
HELLO WELCOMEE!!!
USERNAME : abcd
PASSWORD : 12345
LOGIN
LOGIN FAILED !!!
USERNAME :
PASSWORD :
LOGIN

Q.2 B) CREATE A SERVLET THAT USES COOKIES TO STORE THE NUMBER OF TIMES A USER HAS VISITED SERVLET.

CODE:

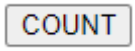
INDEX.HTML

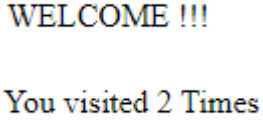
```
<html>
    <head>
        <title>COOKIES</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <form action="cookiescount" method="post">
            <input type="submit" value="COUNT"/>
        </form>

    </body>
</html>
```

COOKIESCOUNT.JAVA

```
import javax.servlet.http.Cookie;

static int count=1;
    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter(); {
            String c=String.valueOf(count);
            Cookie ck=new Cookie("noofvisit",c);
            response.addCookie(ck);
            int val=Integer.parseInt(ck.getValue());
            if(val==1)
            {
                out.println("WELCOME !!!");
            }
            else
            {
                out.println("You visited "+val+" Times");
            }
            count++;
        }
    }
```

OUTPUT:





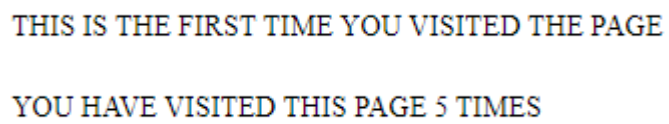
Q.2 C) CREATE A SERVLET DEMONSTRATING THE USE OF SESSION CREATION AND DESTRUCTION. ALSO CHECK WHETHER THE USER HAS VISITED THIS PAGE FIRST TIME OR HAS VISITED EARLIER ALSO USING SESSIONS.

CODE:

SESSIONN.JAVA

```
package session;

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

public class session extends HttpServlet {

    private int counter;
    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        HttpSession session=request.getSession(true);
        if(session.isNew())
        {
            out.print("THIS IS THE FIRST TIME YOU VISITED THE PAGE");
            ++counter;
        }
        else
        {
            synchronized(session.this)
            {
                if(counter==10)
                {
                    session.invalidate();
                    counter=0;
                    request.getSession(false);
                }
                else
                    out.print("YOU HAVE VISITED THIS PAGE "+(++counter)+ "
TIMES");
            }
        }
    }
```

OUTPUT:



PRACTICAL 3

SERVLET IO AND FILE APPLICATIONS.

Q.3 A) CREATE A SERVLET APPLICATION TO UPLOAD AND DOWNLOAD A FILE.

CODE:

FILE UPLOADING

INDEX.HTML

```
<html>
    <head>
        <title>FILE UPLOADING</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <form action="file_uploading" method="post" enctype="multipart/form-
data">
            SELECT FILE TO UPLOAD : <input type="file" name="file" id="file">
            DESTINATION : <input type="text" value="E:\FileUploadPath"
name="destination">
            <br>
            <input type="submit" value="UPLOAD FILE" name="upload"
id="upload">
        </form>
    </body>
</html>
```

FILE_UPLOADING.JAVA

```
package fileupload;

import java.io.*;
import javax.servlet.*;
import javax.servlet.annotation.MultipartConfig;
import javax.servlet.http.*;
@MultipartConfig

protected void doPost(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException
    {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        {
                String path=request.getParameter("destination");
                Part filePart=request.getPart("file");
                String sfilePart=request.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 SUCCESSFULLLY !!!");
                }
                catch(FileNotFoundException e)
                {
                    out.print(e);
                }
        }
    }
```

```
}
```

FILE DOWNLOADING

INDEX.HTML

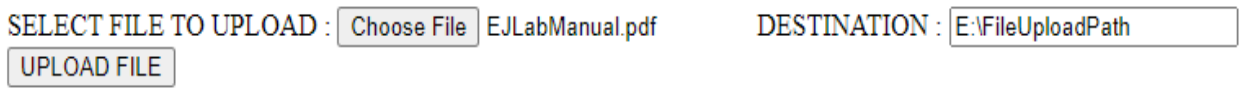
```
<html>
    <head>
        <title>FILE DOWNLOADING</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <h1>FILE DOWNLOADING APPLICATION</h1>
        Click To Download <a
href="File_Downloading?filename=EJLabManual.pdf">EJ LAB MANUAL</a>
        <br/><br/>
        Click To Download <a
href="File_Downloading?filename=AWPLabManual.pdf">AWP LAB MANUAL</a>
    </body>
</html>
```

FILE_DOWNLOADING.JAVA

```
package filedownload;

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

protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        String filename = request.getParameter("filename");
        int len=0;
        try(ServletOutputStream out = response.getOutputStream())
        {
            ServletContext context= getServletConfig().getServletContext();
            response.setContentType((context.getMimeType(filename) !=
null)?context.getMimeType(filename):"application/pdf");
            response.setHeader("Content-Disposition","attachment;
filename=\""+ filename+"\"");
            InputStream is=context.getResourceAsStream("/" + filename);
            byte[] b= new byte[1024];
            while((is != null ) && ((len= is.read(b)) != -1))
            {
                out.write(b,0,len);
            }
            out.flush();
            is.close();
            out.close();
        }
    }
}
```

OUTPUT:

FILE UPLOADING



filePart: File name=EJLabManual.pdf, StoreLocation=C:\Users\VINIT\AppData\Roaming\NetBeans\8.2\config\GF_4.1.1\domain1\generated\jsp\EJ_PRACTICAL_3A\upload_41c1ab5c_18b25234ef7__7ff7_00000009.tmp, size=2009505bytes, isFormField=false, FieldName=file

---

file name: EJLabManual.pdf
FILE UPLOADED SUCCESSFULLLY !!!

FILE DOWNLOADING

Q.3 B) DEVELOP SIMPLE SERVLET QUESTION ANSWER APPLICATION USING DATABASE.

CODE:

CREATE A TABLE IN MYSQL:

- Select **Services** → Expand **Databases** → Right click on **MySQL Server at localhost:3306[disconnected]** → Click on **Connect** → Enter Your **Password** → **OK.**
- Again, Right Click on **MySQL Server at localhost:3306** → Select **Create Database** → Enter Your Database Name and Select the **Checkbox** to grant permission.
- Right click on **Table** under your Database.
- Enter Your Table **Name** by Replacing **Untitled**. Click on **Add Column**, and Add Columns as below Mentioned:
  Name → **queno,** Type → **Integer,** Constraint → **Primary Key**
  Name → **question,** Type → **Varchar,** Size → **200**
  Name → **opt1,** Type → **Varchar,** Size → **50**
  Name → **opt2,** Type → **Varchar,** Size → **50**
  Name → **opt3,** Type → **Varchar,** Size → **50**
  Name → **opt4,** Type → **Varchar,** Size → **50**
  Name → **answer,** Type → **Varchar,** Size → **50**
- **Insert Minimum 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** in **Libraries** folder:
  Right click on **Libraries** folder Select Add **JAR/Folder.**
  mysql-connector-jar File Download Link
  (http://www.java2s.com/example/jar/m/download-mysqlconnectorjava8011jar-file.html#google_vignette)

INDEX.HTML

```
<html>
    <head>
        <title>QUESTION ANSWER APPLICATION</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <h1><a href="questionss" >CLICK TO START</a></h1>
    </body>
</html>
```

QUESTIONS.JAVA

```
package servletpack;

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

public class questionss 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("<form action='markss' method='post'>");
try {
    Class.forName("com.mysql.jdbc.Driver");
    Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/vinitx67?autoReconnec
t=true&useSSL=false", "root", "67Xvinit2021");
    Statement stmt = con.createStatement();
```

```
    ResultSet res = stmt.executeQuery("select * from dbqueans");
    out.println("<table border=1>");
    int qno = 0;
    while (res.next()) {
        qno++;
        out.println("<tr><td>" + res.getString(1) + "</td>");
        out.println("<td>" + res.getString(2) + "</td></tr>");
        out.println("<tr><td><input type='radio' name='q" + qno + "' value='"
+ res.getString(3) + "'></td><td>" + res.getString(3) + "</td></tr>");
        out.println("<tr><td><input type='radio' name='q" + qno + "' value='"
+ res.getString(4) + "'></td><td>" + res.getString(4) + "</td></tr>");
        out.println("<tr><td><input type='radio' name='q" + qno + "' value='"
+ res.getString(5) + "'></td><td>" + res.getString(5) + "</td></tr>");
        out.println("<tr><td><input type='radio' name='q" + qno + "' value='"
+ res.getString(6) + "'></td><td>" + res.getString(6) + "</td></tr>");
    }
}

        catch(Exception e)
        {
            out.println(e);
        }
        out.println("</table>");
        out.println("<input type=RESET >");
        out.println("<input type=submit value=SUBMIT >");
        out.println("</form>");
    }
}
```

MARKSS.JAVA

```
package servletpack;

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

public class markss extends HttpServlet
{
    public void doPost(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException
    {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/vinitx67?autoReconnec
t=true&useSSL=false", "root", "67Xvinit2021");
            Statement stmt = con.createStatement();
            ResultSet res = stmt.executeQuery("select ans from dbqueans");
            int count = 0, qno = 0;
            while (res.next())
            {
                qno++;
                String selectedAnswer = request.getParameter("q" + qno);
                if (selectedAnswer != null &&
selectedAnswer.equals(res.getString(1)))
                {
                    count++;
                    out.println("<h1>CORRECT </h1>");
                }
                else
                {
                    out.println("<h1>INCORRECT </h1>");
                }
            }
            out.println("<h1>YOUR MARKS IS " + count + " </h1>");
        }
        catch (Exception e)
        {
            out.println(e);
        }
```

```
    }
}
```

OUTPUT:

SELECT * FROM dbqueans LI... ×

Max. rows: 100 | Fetched Rows: 5 | Matching Rows:

#	qno	question	op1	op2	op3	op4	ans
1	1	Who invented Java Programming?	Guido van Rossum	James Gosling	Dennis Ritchie	Bjarne Stroustrup	James Gosling
2	2	Which component is used to compile, debug and execute the java programs?	JRE	JIT	JDK	JVM	JDK
3	3	Which one of the following is not a Java feature?	Object-oriented	Use of pointers	Portable	Dynamic and Extensible	Use of pointers
4	4	Which of these cannot be used for a variable name in Java?	identifier & keyword	identifier	keyword	none of the mentioned	keyword
5	5	What is the extension of java code files?	.js	.txt	.java	.html	.java

CLICK TO START

1	Who invented Java Programming?
○	Guido van Rossum
◉	James Gosling
○	Dennis Ritchie
○	Bjarne Stroustrup
2	Which component is used to compile, debug and execute the java programs?
○	JRE
○	JIT
◉	JDK
○	JVM
3	Which one of the following is not a Java feature?
○	Object-oriented
○	Use of pointers
◉	Portable
○	Dynamic and Extensible
4	Which of these cannot be used for a variable name in Java?
○	identifier & keyword
○	identifier
○	keyword
◉	none of the mentioned
5	What is the extension of java code files?
○	.js
○	.txt
◉	.java
○	.html

Reset SUBMIT

CORRECT

CORRECT

INCORRECT

INCORRECT

CORRECT

YOUR MARKS IS : 3

Q.3 C) CREATE SIMPLE SERVLET APPLICATION TO DEMONSTRATE NON-BLOCKING READ OPERATION.

CODE:

INDEX.HTML

```
Non Blo <html>
    <head>
        <title>NON-BLOCKING READ OPERATION</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <a href="NonBlockingServlet">Non Blocking Servlet</a>
    </body>
</html>
```

NONBLOCKINGSERVLET.JAVA

```
package blocking;

import java.io.*;
import java.net.*;
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.*;

public class NonBlockingServlet extends HttpServlet {
    protected void service(HttpServletRequest request, HttpServletResponse
response) throws
            ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try(PrintWriter out = response.getWriter()){
            out.println("<h1>File Reader</h1>");
            String filename="/WEB-INF/demo.txt";
            ServletContext c = getServletContext();
            InputStream in = c.getResourceAsStream(filename);
            String path = "http://" + request.getServerName() + ":" +
request.getServerPort() + request.getContextPath() +
"/ReadingNonBlockingServlet";
            URL url = new URL(path);
            HttpURLConnection conn = (HttpURLConnection) url.openConnection();
            conn.setChunkedStreamingMode(2);
            conn.setDoOutput(true);
            conn.connect();
            if(in!=null)
            {
                InputStreamReader inr = new InputStreamReader(in);
                BufferedReader br = new BufferedReader(inr);
                String text = "";
                System.out.println("Reading started...");
                BufferedWriter bw = new BufferedWriter(new
OutputStreamWriter(conn.getOutputStream()));
                while ((text = br.readLine()) != null){
                    out.print(text+"<br>");
                    try{
                        Thread.sleep(1000);
                        out.flush();
                    }
                    catch(InterruptedException ex) {}
                }
                out.print("Reading Completed.....");
                bw.flush();
                bw.close();
            }
        }
    }
}
```

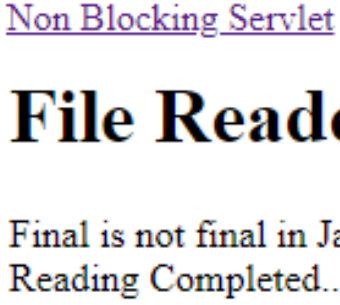
READINGNONBLOCKINGSERVLET.JAVA

```
package blocking;

import java.io.*;
import javax.servlet.*;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;
import javax.servlet.AsyncContext;

@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));
    }
}
```

READINGLISTNER.JAVA

```
package blocking;

import java.io.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.*;
import javax.servlet.ReadListener;
import javax.servlet.AsyncContext;
import javax.servlet.ServletInputStream;

public class ReadingListener implements ReadListener{
    private ServletInputStream input = null;
    private AsyncContext ac = null;
    ReadingListener(ServletInputStream in, AsyncContext c)
    {
        input = in;
        ac = c;
    }
    @Override
    public void onDataAvailable() throws IOException {
    }
    public void onAllDataRead() throws IOException
    {
        ac.complete();
    }
    public void onError(final Throwable t)
    {
        ac.complete();
        t.printStackTrace();
    }
}
```

OUTPUT:



PRACTICAL 4

IMPLEMENT THE FOLLOWING JSP APPLICATIONS.

Q.4 A) DEVELOP A SIMPLE JSP APPLICATION TO DISPLAY VALUES OBTAINED FROM THE USE OF INTRINSIC OBJECTS OF VARIOUS TYPES.

CODE:

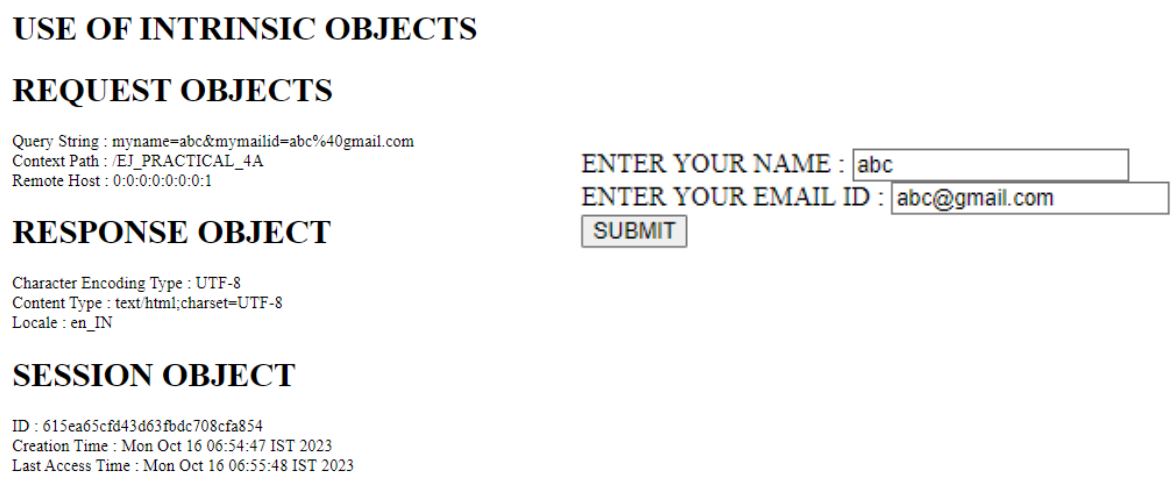
INDEX.HTML

```
<html>
    <head>
        <title>INTRINSIC OBJECTS</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <form action="intrinsic.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>
```

INTRINSIC.JSP

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP PAGE</title>
    </head>
    <body>
        <h1>USE OF INTRINSIC OBJECTS</h1>
        <h1>REQUEST OBJECTS</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())%><br>
        Last Access Time : <%=new
java.util.Date(session.getLastAccessedTime())%><br>
    </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.HTML

```
<html>
    <head>
        <title>PASS VALUES FROM ONE PAGE TO ANOTHER WITH VALIDATION</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <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

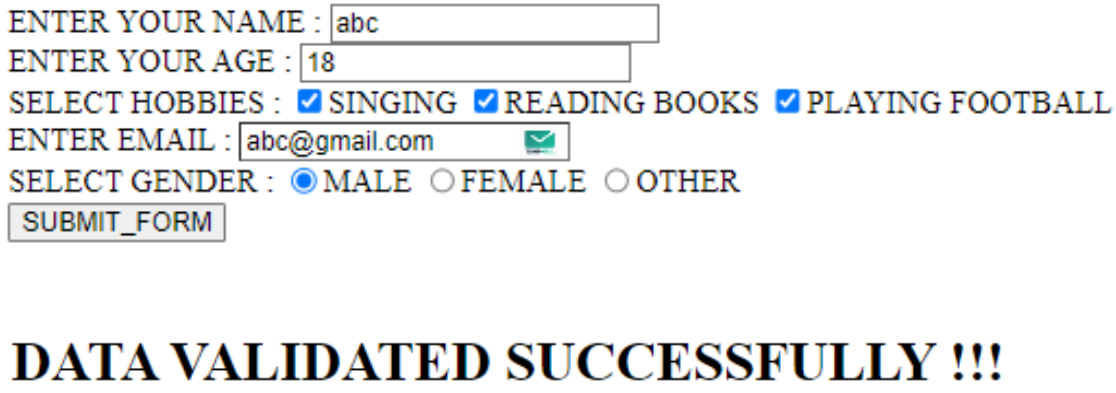
```
<%@page contentType="text/html" pageEncoding="UTF-8" import="mypack.*"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP PAGE</title>
    </head>
    <body>
    <body>
        <h1>VALIDATION PAGE</h1>
        <jsp:useBean 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"%>
<!DOCTYPE html>
<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:



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>REGISTRATION AND LOGIN JSP APPLICATION</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <form action="registration.jsp">
            <h1>NEW USER REGISTRATION PAGE</h1>
            ENTER USER NAME : <input type="text" name="txtName"><br>
            ENTER PASSWORD : <input type="password" name="txtPass1"><br>
            RE-ENTER PASSWORD : <input type="password" name="txtPass2"><br>
            ENTER EMAIL : <input type="text" name="txtEmail"><br>
            ENTER COUNTRY NAME : <select name="txtCon">
                <option>INDIA</option>
                <option>NORWAY</option>
                <option>SWITZERLAND</option>
                <option>ICELAND</option>
            </select><br>
            <input type="submit" value="REGISTER"><input type="reset"
value="RESET">
            <p>Already have an account? <a href="login.html">Login
here</a></p>
        </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.getParameter("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/vinitx67?autoReconnec
t=true&useSSL=false", "root", "67Xvinit2021");
                    PreparedStatement stmt=con.prepareStatement("insert into
userinfo 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>
    <head>
        <title>LOGIN PAGE</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <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">
            <p>Don't have an account? <a href="index.html">Create
Account</a></p>
        </form>
    </body>
</html>
```

LOGIN.JSP

```
<%@page contentType="text/html" import="java.sql.*"%>
<!DOCTYPE html>
<html>
    <body>
        <h1>LOGIN 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/vinitx67?autoReco
nnect=true&useSSL=false", "root", "67Xvinit2021");
                Statement stmt=con.createStatement();
                rs=stmt.executeQuery("select pass from userinfo where
uname='"+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:

NEW USER REGISTRATION PAGE

ENTER USER NAME : abcd
ENTER PASSWORD : ••••••••
RE-ENTER PASSWORD : ••••••••
ENTER EMAIL : abcd@gmail.com
ENTER COUNTRY NAME : INDIA
REGISTER RESET

Already have an account? Login here

REGISTRATION JSP PAGE

REGISTRATION SUCCESSFUL

SELECT * FROM userinfo LI... ×

Max. rows: 100 Fetched Rows: 2

#	uname	pass	email	cntry
1	abc	12345	abc@gmail.com	INDIA
2	abcd	12345678	abcd@gmail.com	INDIA

LOGIN PAGE

ENTER USER NAME : abcd
ENTER PASSWORD : ••••••••
LOGIN Reset

Don't have an account? Create Account

LOGIN JSP PAGE

LOGIN SUCCESSFULLL

PRACTICAL 5

IMPLEMENT THE FOLLOWING JSP JSTL AND EL APPLICATIONS.

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:

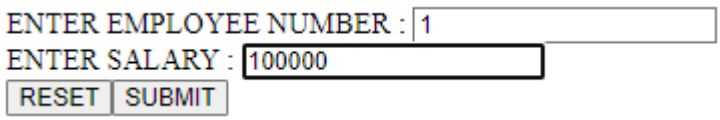
INDEX.HTML

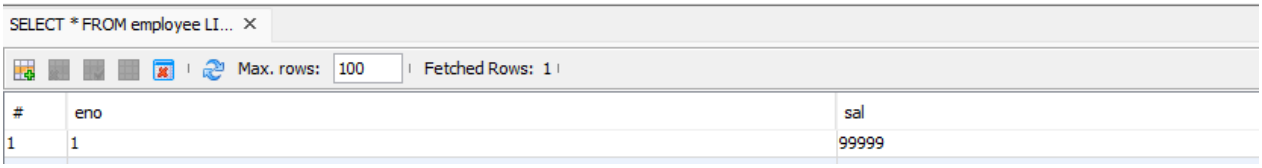
```
<html>
    <body>
        <form action="emp.jsp" >
            ENTER EMPLOYEE NUMBER : <input type="text" name="emp_no" ><br>
            ENTER SALARY : <input type="text" name="salary" ><br>
            <input type="reset" value="RESET" ><input type="submit"
value="SUBMIT">
        </form>
    </body>
</html>
```

EMP.JSP

```
<%@page contentType="text/html" import="java.sql.*" %>
<html>
    <body>
        <h1>UPDATING EMPLOYEE RECORD</h1>
        <%
            String eno = request.getParameter("emp_no");
            String sal = request.getParameter("salary");
            try {
                Class.forName("com.mysql.jdbc.Driver");
                Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/vinitx67?autoReconnec
t=true&useSSL=false","root","67Xvinit2021");
                PreparedStatement stmt = con.prepareStatement("select * from
employee where eno=?");
                stmt.setString(1, eno);
                ResultSet rs = stmt.executeQuery();
                if (rs.next()) {
                    out.println("<h1> EMPLOYEE " + rs.getString("eno") + "
EXIST </h1>");
                    PreparedStatement pst = con.prepareStatement("update
employee set sal=? where eno=?");
                    pst.setString(1, sal);
                    pst.setString(2, eno);
                    pst.executeUpdate();
                    out.println("<h1>EMPLOYEE RECORDS UPDATED !!!</h1>");
                }
                else
                {
                    out.println("<h1>EMPLOYEE RECORD DOES NOT EXIST
!!!</h1>");
                }
            }
            catch (Exception e)
            {
                out.println(e);
            }
        %>
    </body>
</html>
```

OUTPUT:




SELECT * FROM employee LI...

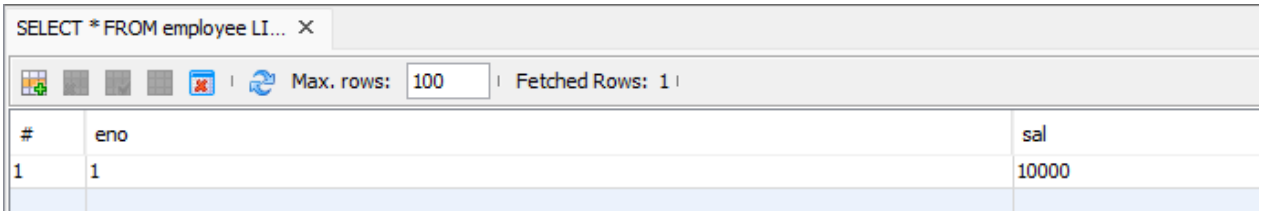
Max. rows: 100 Fetched Rows: 1

#	eno	sal
1	1	99999

UPDATING EMPLOYEE RECORD

EMPLOYEE 1 EXIST

EMPLOYEE RECORDS UPDATED !!!


SELECT * FROM employee LI...

Max. rows: 100 Fetched Rows: 1

#	eno	sal
1	1	10000

Q.5 B) CREATE A JSP PAGE TO DEMONSTRATE THE USE OF EXPRESSION LANGUAGE.

CODE:

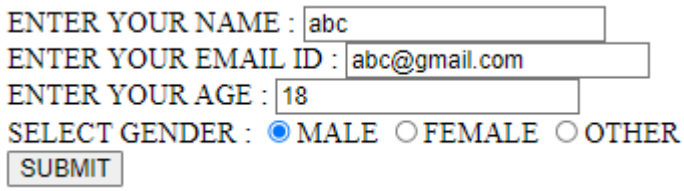
INDEX.HTML

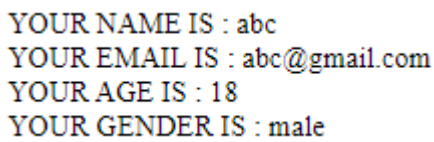
```
<html>
    <head>
        <title>JSP PAGE</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <form action="info.jsp">
            ENTER YOUR NAME : <input type= "text" name="uname"><br>
            ENTER YOUR EMAIL ID : <input type="text" name="em"><br>
            ENTER YOUR AGE : <input type="text" name="ag"><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= "submit" value="SUBMIT">
        </form>
    </body>
</html>
```

INFO.JSP

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP PAGE</title>
    </head>
    <body>
        YOUR NAME IS : ${param.uname}<br>
        YOUR EMAIL IS : ${param.em}<br>
        YOUR AGE IS : ${param.ag}<br>
       YOUR GENDER IS : ${param.gender}
    </body>
</html>
```

OUTPUT:





Q.5 C) CREATE A JSP APPLICATION TO DEMONSTRATE THE USE OF JSTL.

CODE:

INDEX.HTML

ADD JSTL JAR FILE (http://www.java2s.com/Code/Jar/j/Downloadjstl12jar.htm)

```
<html>
    <head>
        <title>JSP APPLICATION FOR USE OF JSTL</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <a href="satDemo.jsp"> SetDemo</a><br>
        <a href="Maxif.html"> Maxif</a><br>
        <a href="forEachDemo.jsp"> ForEachDemo</a><br>
        <a href=" outDemo.jsp "> OutDemo</a><br>
        <a href="URLDemo.jsp"> URLDemo</a><br>
        <a href="choose_when_otherwise.jsp"> choose_when_otherwise</a><br>
    </body>
</html>
```

SATDEMO.JSP

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>SET DEMO</title>
    </head>
    <body>
        <%@taglib prefix="c" uri="http://java.sun.com/jstl/core" %>
        <c:set var="pageTitle" scope="application"
               value="INDIAN PREMIERE LEAGUE: REGISTRATION" />
        ${pageTitle}
    </body>
</html>
```

MAXIF.HTML

```
<html>
    <head>
        <title>MAX IF</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <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>
    </body>
</html>
```

IFDEMO.JSP

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>IF DEMO</title>
    </head>
    <body>
        <%@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>
    </body>
</html>
```

FOREACHDEMO.JSP

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>FOR EACH DEMO</title>
    </head>
    <body>
        <%@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>
    </body>
</html>
```

OUTDEMO.JSP

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>OUT DEMO</title>
    </head>
    <body>
        <%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
        <c:set var="name" value="ABC"/>
        My name is: <c:out value= "${name}" />
    </body>
</html>
```

URLDEMO.JSP

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>URL DEMO</title>
    </head>
    <body>
        <%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
        <c:url value="/index.html"/>
    </body>
</html>
```

CHOOSE_WHEN_OTHERWISE.JSP

```
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>CHOOSE WHEN OTHERWISE</title>
    </head>
    <body>
        <%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
        <c:set var="income" value="${40000*4}"/>
        YOUR INCOME IS : <c:out value="${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>
    </body>
</html>
```

OUTPUT:

SetDemo
Maxif
ForEachDemo
OutDemo
URLDemo
choose_when_otherwise

INDIAN PREMIERE LEAGUE: REGISTRATION

x= 111
y= 11
Check Max

The Ans is:

111 is greater than 11

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

My name is: ABC

/EJ_PRACTICAL_5C/index.html

YOUR INCOME IS : 160000 INCOME IS VERY GOOD !!!

YOUR INCOME IS : 400 INCOME IS NOT GOOD !!!

YOUR INCOME IS : 1600 INCOME IS UNDETERMINED !!!

PRACTICAL 6

IMPLEMENT THE FOLLOWING EJB APPLICATIONS.

Q.6 A) CREATE A CURRENCY CONVERTER APPLICATION USING EJB.

CODE:

INDEX.HTML

```
<html>
    <head>
        <title>CURRENCY CONVERTER</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </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 1:

Create a **Session Bean** Named as **CCBean** 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 **CCBean.java** and **CCBeanLocal.java**

CCBEANLOCAL.JAVA

```
package mybeans;

import javax.ejb.Local;
@Local
public interface CCBeanLocal
{
    public double r2Dollar(double r);
    public double d2Rupees(double d);
}
```

CCBEAN.JAVA

```
package mybeans;

import javax.ejb.Stateless;

@Stateless
public class CCBean implements CCBeanLocal
{
    public double r2Dollar(double r)
    {
        return r/81.5;
    }
    public double d2Rupees(double d)
    {
        return d*81.5;
    }
}
```

STEP 2:

Create a **Servlet** file Named as **ccservlet.java.**

CCSERVLET.JAVA

```
package mypackage;

import java.io.*;
import java.io.PrintWriter;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.ejb.EJB;
import mybeans.CCBeanLocal;

public class CCServlet extends HttpServlet {
    @EJB CCBeanLocal obj;
    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            double amt = Double.parseDouble(request.getParameter("amt"));
            if(request.getParameter("type").equals("r2d"))
            {
                out.println("<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:

ENTER AMOUNT : 999
SELECT CONVERSION TYPE ○ Rupees TO Dollar ◉ Dollar TO Rupees
Reset CONVERT

999.0 Dollars = 81418.5 Rupees

ENTER AMOUNT : 999
SELECT CONVERSION TYPE ◉ Rupees TO Dollar ○ Dollar TO Rupees
Reset CONVERT

999.0 Rupees = 12.257668711656441 Dollars

Q.6 B) DEVELOP A SIMPLE ROOM RESERVATION SYSTEM APPLICATION USING EJB
CODE:

INDEX.HTML

```
<html>
    <head>
        <title>ROOM RESERVATION SYSTEM</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <form method="post" action="RoomClient">
            <br> NUMBER OF ROOMS <input type=text name="t1">
            <br> <input type="submit" name="btn" value="CheckIn">
            <br> <input type="submit" name="btn" value="CheckOut">
        </form>
    </body>
</html>
```

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:3306/vinitx67?autoReco
nnect=true&useSSL=false", "root", "67Xvinit2021");
            String sql1 = "select * from rooms";
            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 rooms 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:3306/vinitx67?autoReco
nnect=true&useSSL=false", "root", "67Xvinit2021");
            String sql1 = "select * from rooms";
            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 rooms 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;
        }
    }
}
```

ROOMCLIENT.JAVA

```
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:

NUMBER OF ROOMS 67
CheckIn
CheckOut

67 rooms check-in

Back

NUMBER OF ROOMS 50
CheckIn
CheckOut

Not possible to do Check IN / OUT

Back

NUMBER OF ROOMS 50
CheckIn
CheckOut

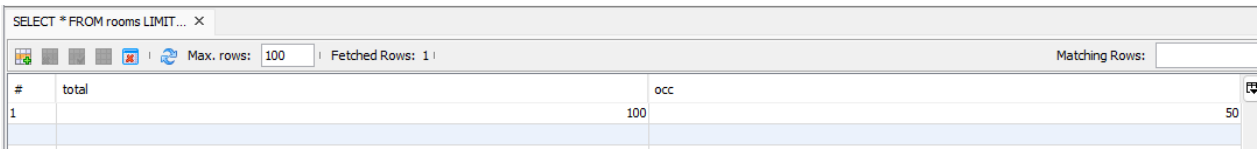
50 rooms check-out

Back

NUMBER OF ROOMS 101
CheckIn
CheckOut

Not possible to do Check IN / OUT

Back



Q.6 C) DEVELOP SIMPLE SHOPPING CART APPLICATION USING EJB [STATEFUL SESSION BEAN].

CODE:

INDEX.HTML

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 Web 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>
```

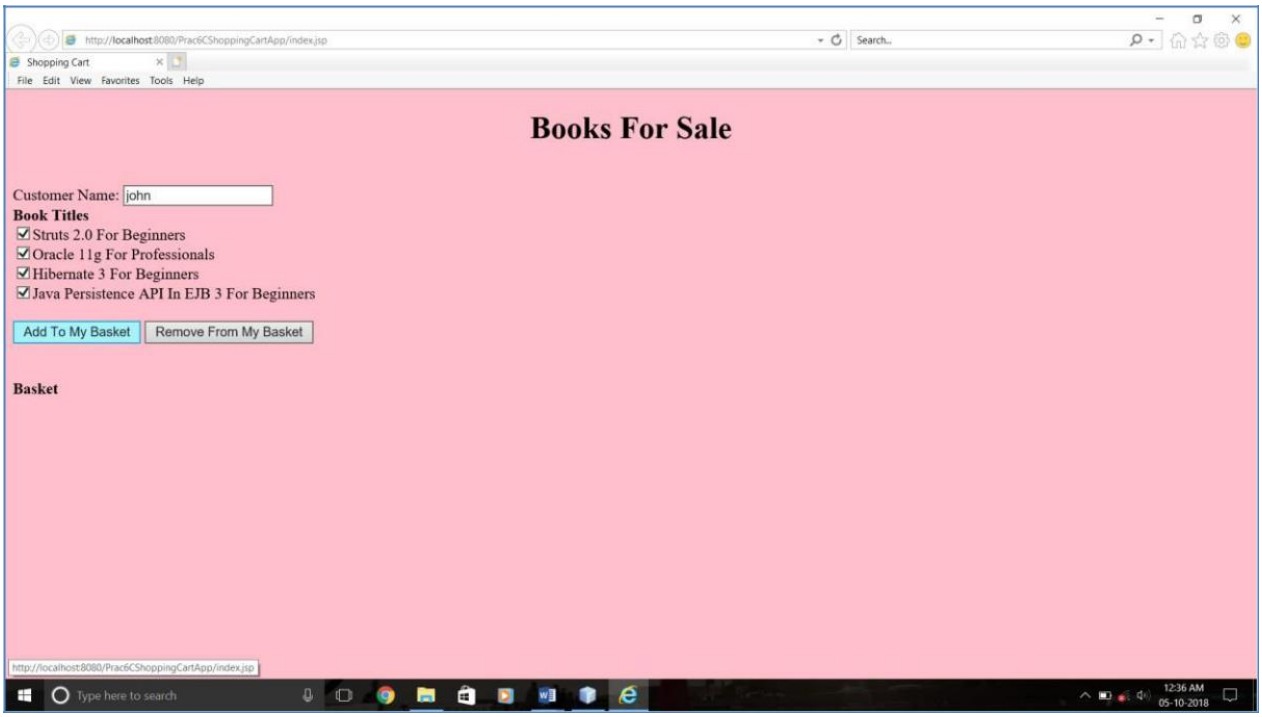
STEP 4: CREATE DATABASE AND DATABASE TABLE

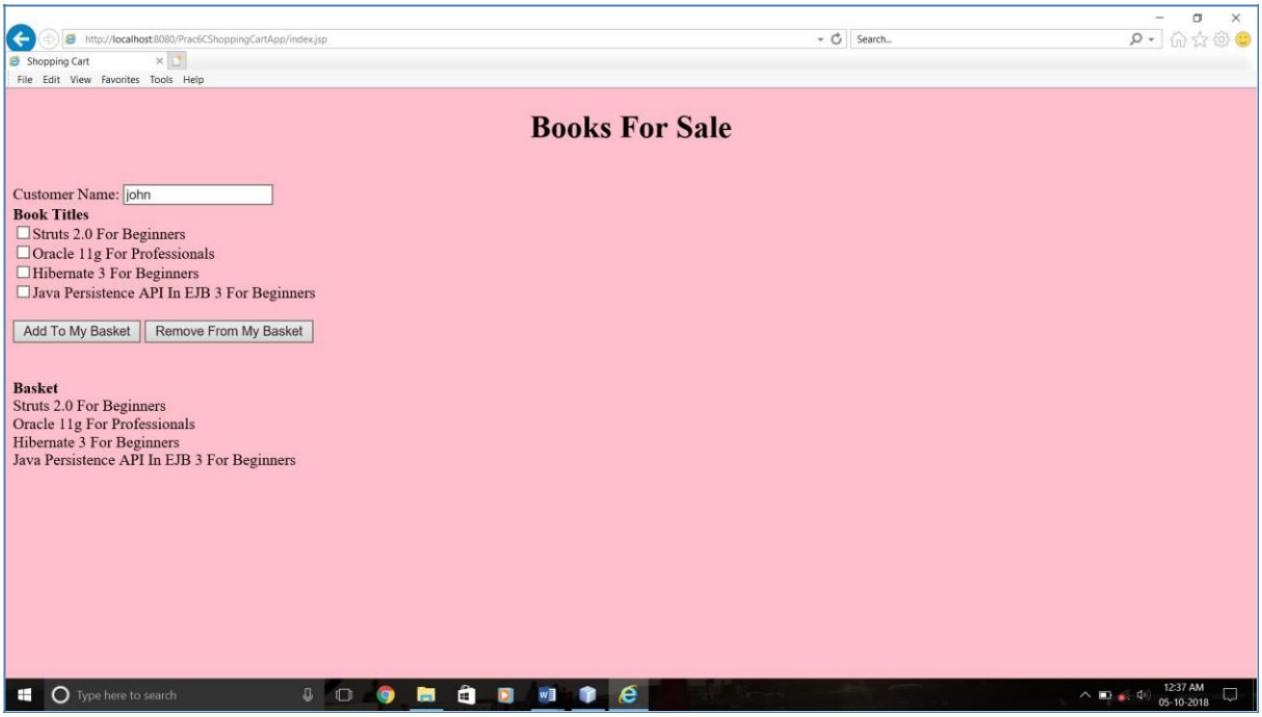
Services → Create Database → **cartdb** → Select cartdb → Right Click → Create Table → **Cart** → **UserName** varchar 35, **ItemName** varchar 50 → **Finish.**

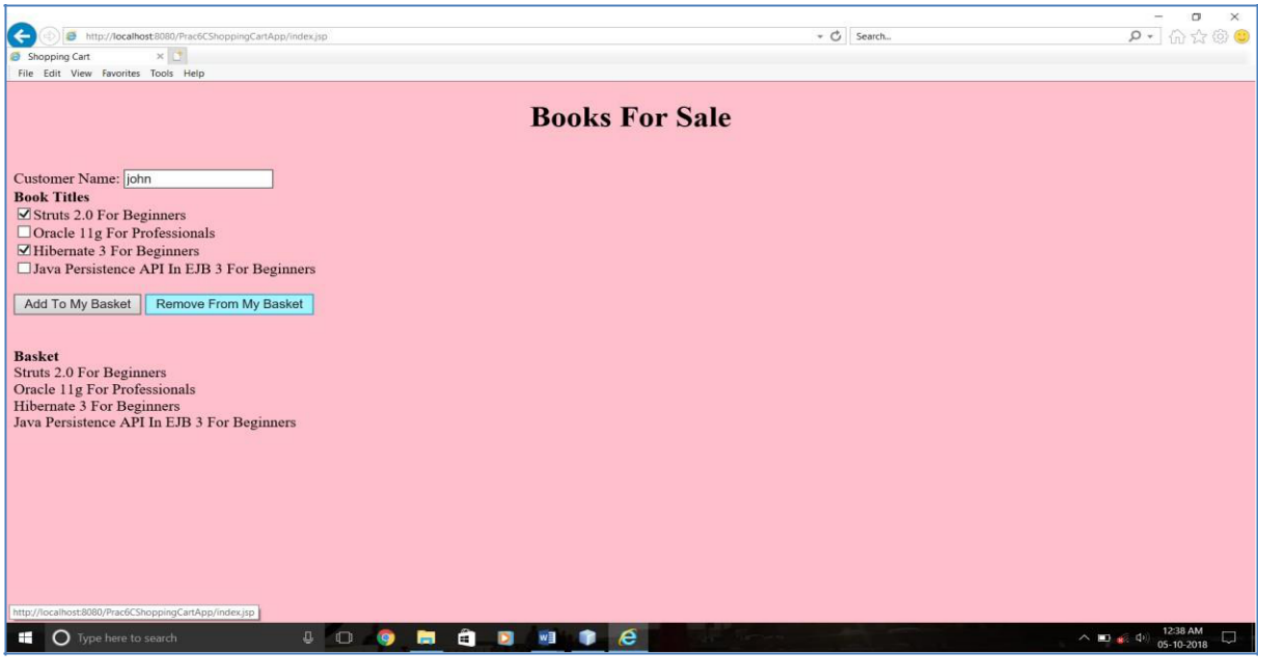
STEP 5: ADD MYSQL CONNECTOR TO THE LIBRARY UNDER PROJECT TAB.

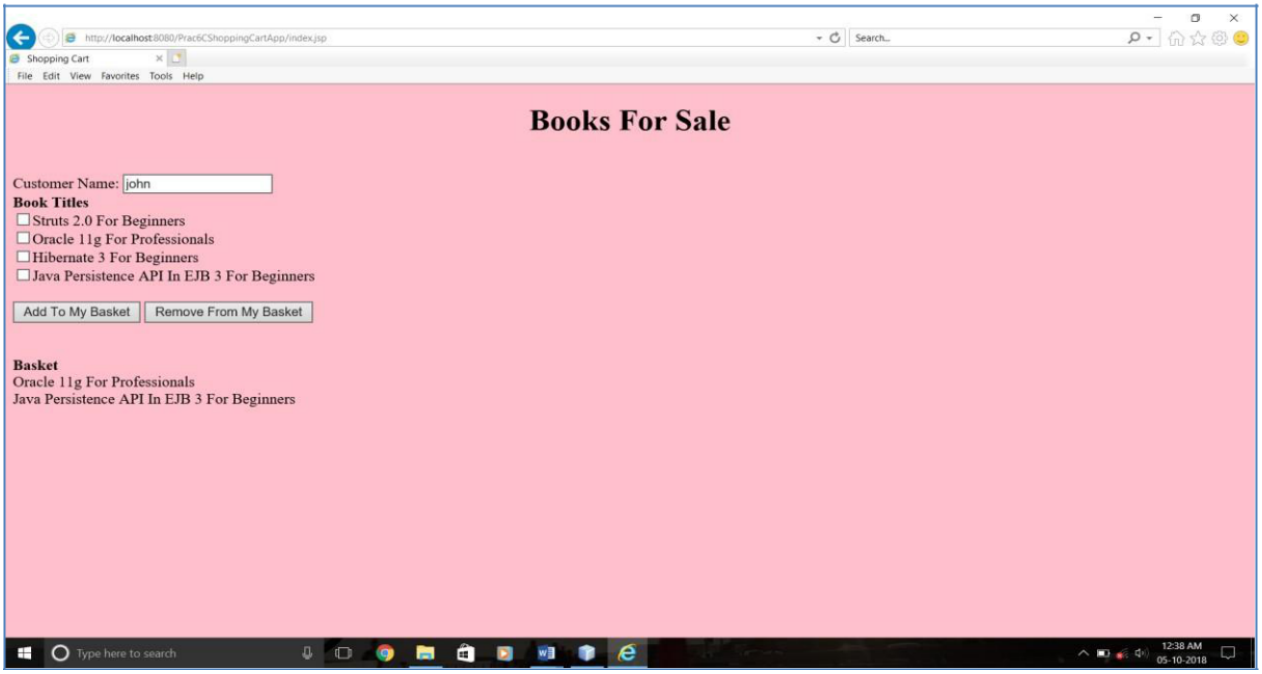
STEP 6: BUILD AND RUN THE APPLICATION.

OUTPUT:


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
Type here to search
12:36 AM
05-10-2018


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


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
12:38 AM
05-10-2018


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
Oracle 11g For Professionals
Java Persistence API In EJB 3 For Beginners
Type here to search
12:38 AM
05-10-2018

PRACTICAL 7

IMPLEMENT THE FOLLOWING EJB APPLICATIONS WITH DIFFERENT TYPES OF BEANS.

Q.7 A) DEVELOP SIMPLE EJB APPLICATION TO DEMONSTRATE SERVLET HIT COUNT USING SINGLETON SESSION BEANS.

CODE:

STEP 1:

Java Web → Web Application → **Pract7AServletHitsSingltonApp** → Finish.

INDEX.HTML

```
<html>
    <head>
        <title>SINGLETON SESSION BEAN</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <a href="ServletClient">CLICK HERE</a>
    </body>
</html>
```

STEP 2:

Create a **Session Bean** Named as **CountServletHitsBean** → Select **Singleton** → Package Name as **ejb** (**do not select Local or Remote**).

COUNTSERVLETHITSBEAN.JAVA

```
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

SERVLETCLIENT.JAVA

```
package servlet;

import ejb.CountServletHitsBean;
import javax.ejb.EJB;
import java.io.*;
import java.io.PrintWriter;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class ServletClient extends HttpServlet {
    @EJB CountServletHitsBean obj;
    protected void processRequest(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException
    {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter())
        {
            out.print("<b>NUMBER OF TIMES THIS SERVLET IS ACCESSED :  </b>
"+obj.getCount());
        }
    }
}
```

OUTPUT:

CLICK HERE

NUMBER OF TIMES THIS SERVLET IS ACCESSED : 6

Q.7 B) DEVELOP SIMPLE EJB APPLICATION TO DEMONSTRATE SERVLET HIT COUNT USING SINGLETON SESSION BEANS.

CODE:

INDEX.HTML

STEP 1:

Web → Web Application → **Pract7BVisitorStatisticsMDBApp** → Select Dedicated Folders For Storing Libraries → **Finish.**

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
connectionFactory; private static Queue
queue; Connection 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=(ConnectionFactory)ic.lookup("jms/Queu
      eFactory");
      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

STEP 3: Create a Session Bean Named as **VisitorStatBean** → Select **Stateless** → Package Name as **ejb**, (**do not select Local / Remote).**

VISITORSTATBEAN

```
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:

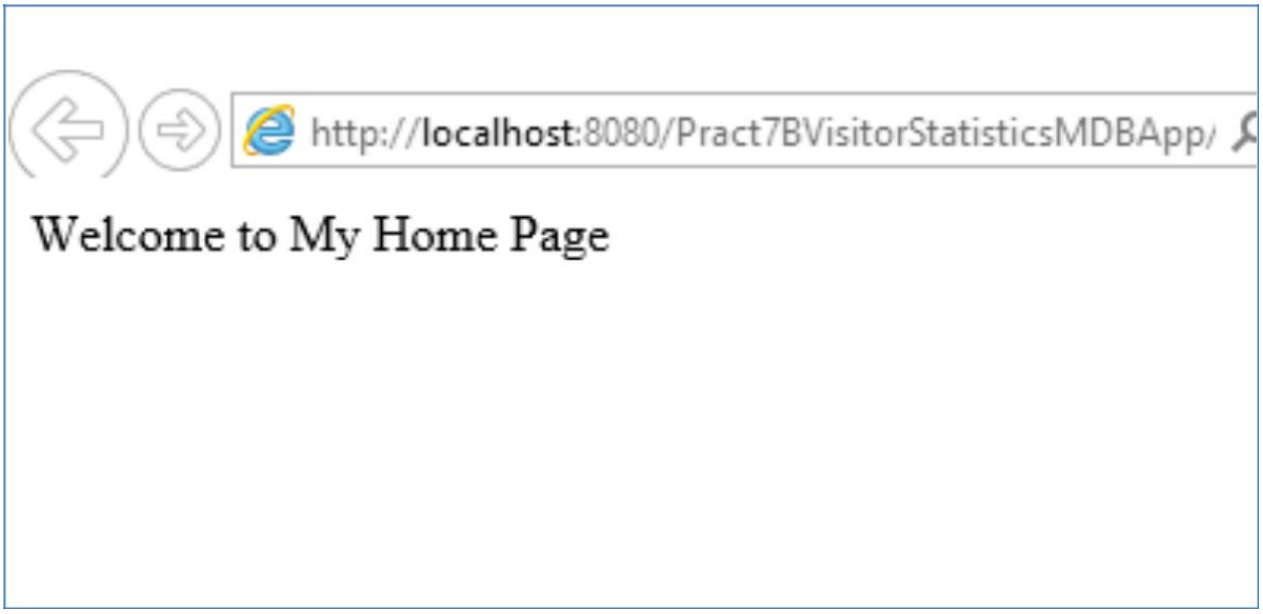
Before Deploying and Running the Application, **Glassfish Server Setting** is Required.
Browse the Path:
Localhost:4848 on any Browser.

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:


http://localhost:8080/Pract7BVisitorStatisticsMDBApp/
Welcome to My Home Page

Q.7 C) DEVELOP SIMPLE MARKS ENTRY APPLICATION TO DEMONSTRATE ACCESSING DATABASE USING EJB.

CODE:

INDEX.HTML

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);

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:

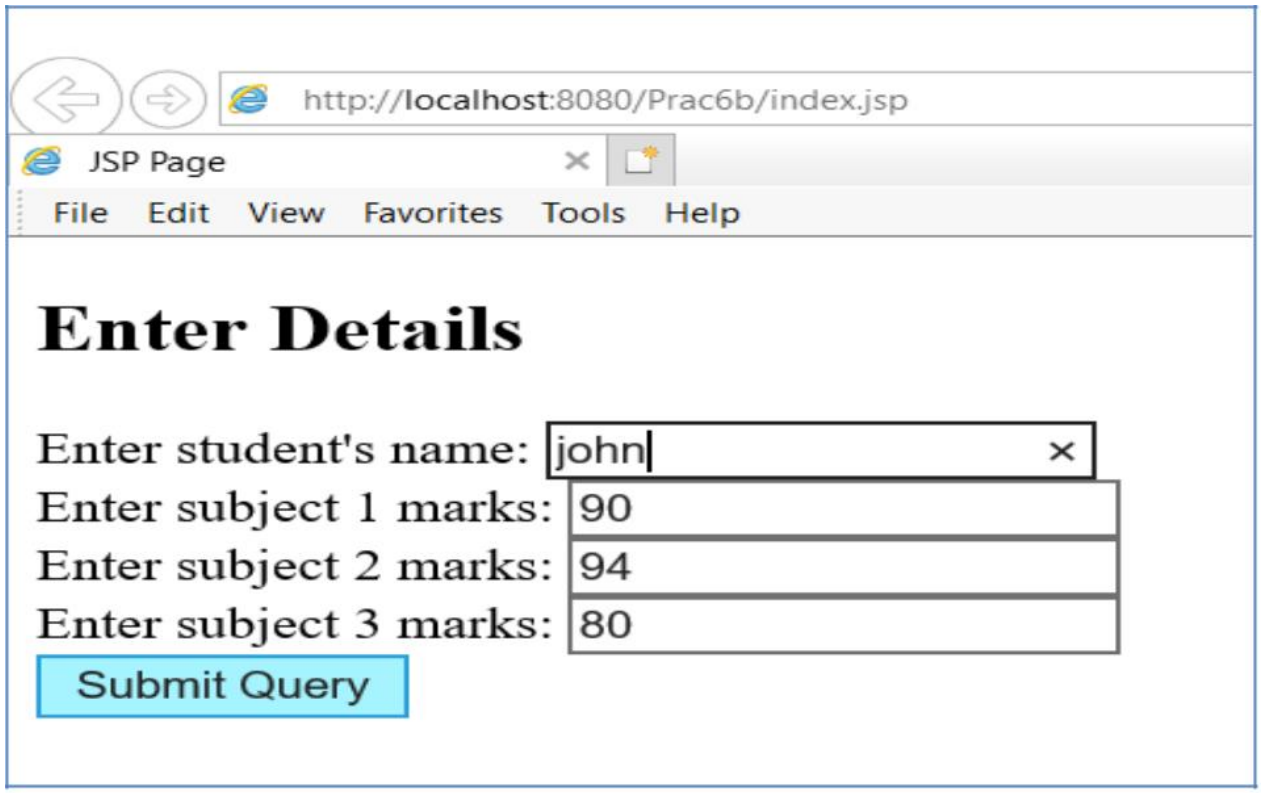
Create Stateful java Bean as Select Source Package → **Session Bean** → Class Name → **MarksEntryBean** → Package → **ejb** → bean type → **stateful** → **don't select Local / Remote** → finish.

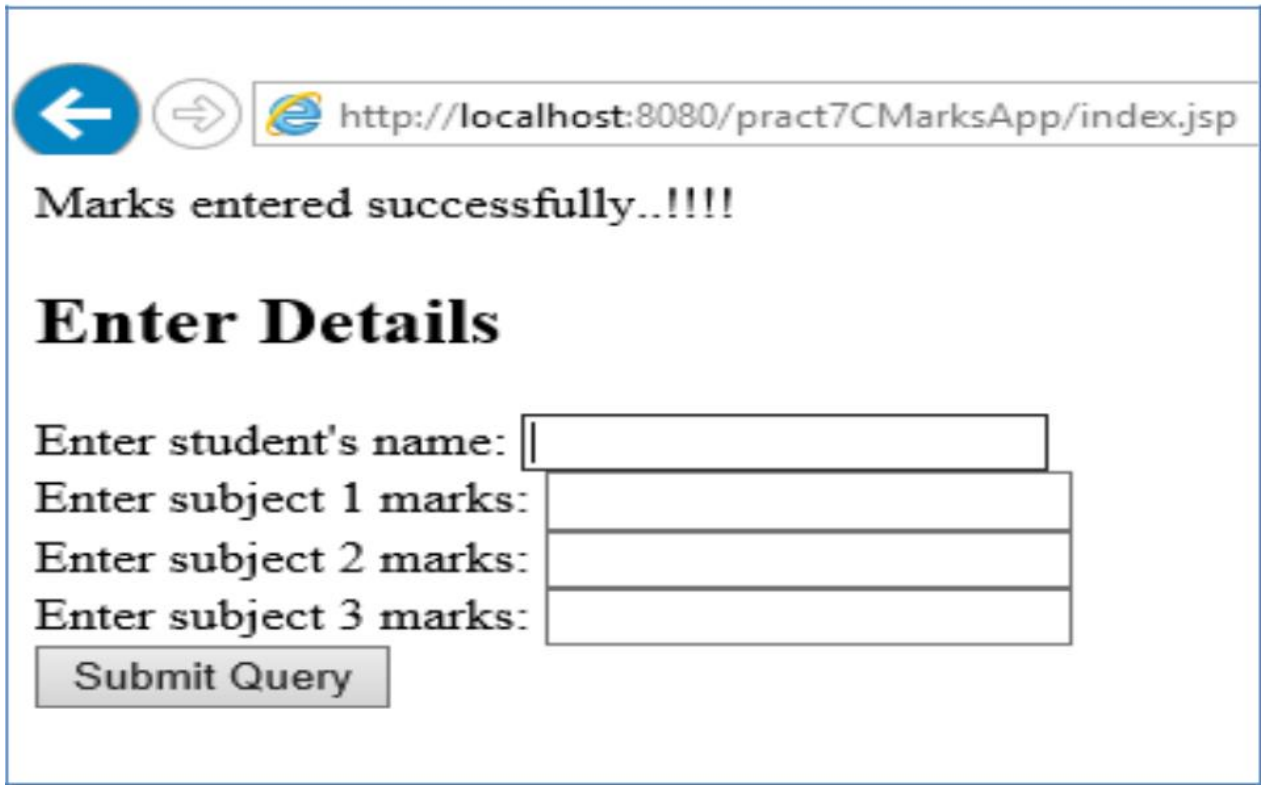
MARKSENTRYBEAN.JAVA

```
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:





PRACTICAL 9

IMPLEMENT THE FOLLOWING JPA APPLICATIONS WITH ORM AND HIBERNATE.

Q.9 A) DEVELOP A JPA APPLICATION TO DEMONSTRATE USE OF ORM ASSOCIATIONS.

CODE:

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">
</head>
<body>
User Details <hr><br><br>
<form action="userview.jsp" >
Name <input type="text" name="uname" maxlength="20"><br> User Type <input
type="text" name="utype" maxlength="20">
<br><input type="submit" value="submit">
</form>
</body>
</html>
```

USERVIEW.JSP

```
<%@page import="java.util.List"%>
<%@page import="java.util.Iterator"%>
<%@page import="hibernate.User"%>
<%!
SessionFactory sf;
org.hibernate.Session ss;
List<hibernate.User> User;
%>
<%
sf = new Configuration().configure().buildSessionFactory();
ss= sf.openSession();
Transaction tx=null;
User ur=new User();
try
{
tx=ss.beginTransaction();
String uname=request.getParameter("uname");
String utype=request.getParameter("utype");
```

```
ur.setUname(uname);
ur.setUtype(utype);
ss.save(ur);
tx.commit();
}
catch(Exception e){ out.println("Error"+e.getMessage()); }
try
{
ss.beginTransaction();
User=ss.createQuery("from User").list();
}
catch(Exception e){ }
%>
<html>
<head>
<title>Guest View</title>
</head>
<body>
Guest View
Click here to go <a href="index.html"> BACK </a> <br><br>
<%
Iterator it=User.iterator();
while(it.hasNext())
{
User eachrecord=(User)it.next();
out.print(eachrecord.getUid()+" ");
out.print(eachrecord.getUname()+"<br>");
out.print(eachrecord.getUtype()+"<br><hr>");
}
%>
</body>
</html>
```

HIBERNATE.REVENGE.XML

```
<hibernate-reverse-engineering>
<schema-selection match-catalog="userdb"/>
<table-filter match-name="user"/>
</hibernate-reverse-engineering>
```

HIBERNATE.CFG.XML

```
<hibernate-configuration>

<session-factory>

<property
name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>

<property
name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>

<property

name="hibernate.connection.url">jdbc:mysql://localhost:3306/userdb?zeroDateTim
eBehavior=conve

rtToNull</property>

<property name="hibernate.connection.username">root</property>

<property name="hibernate.connection.password">tiger</property>

</session-factory>

</hibernate-configuration>
```

USER.HBM.XML

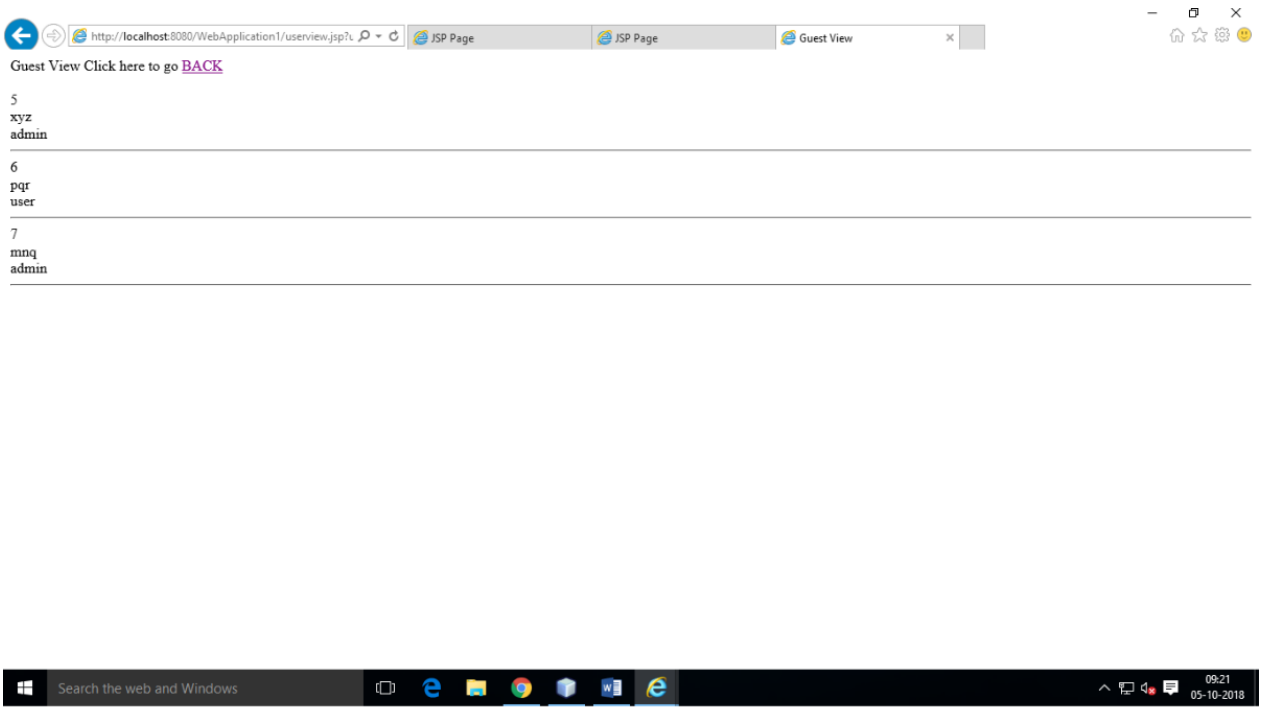
```
<hibernate-mapping>

<class optimistic-lock="version" catalog="userdb" table="user"
name="hibernate.User">

<id name="uid" type="java.lang.Integer">

<column name="uid"/>

<generator class="identity"/>

</id>

<property name="uname" type="string">

<column name="uname" length="20"/>

</property>

<property name="utype" type="string">

<column name="utype" length="100"/>

</property>

</class>

</hibernate-mapping>
```

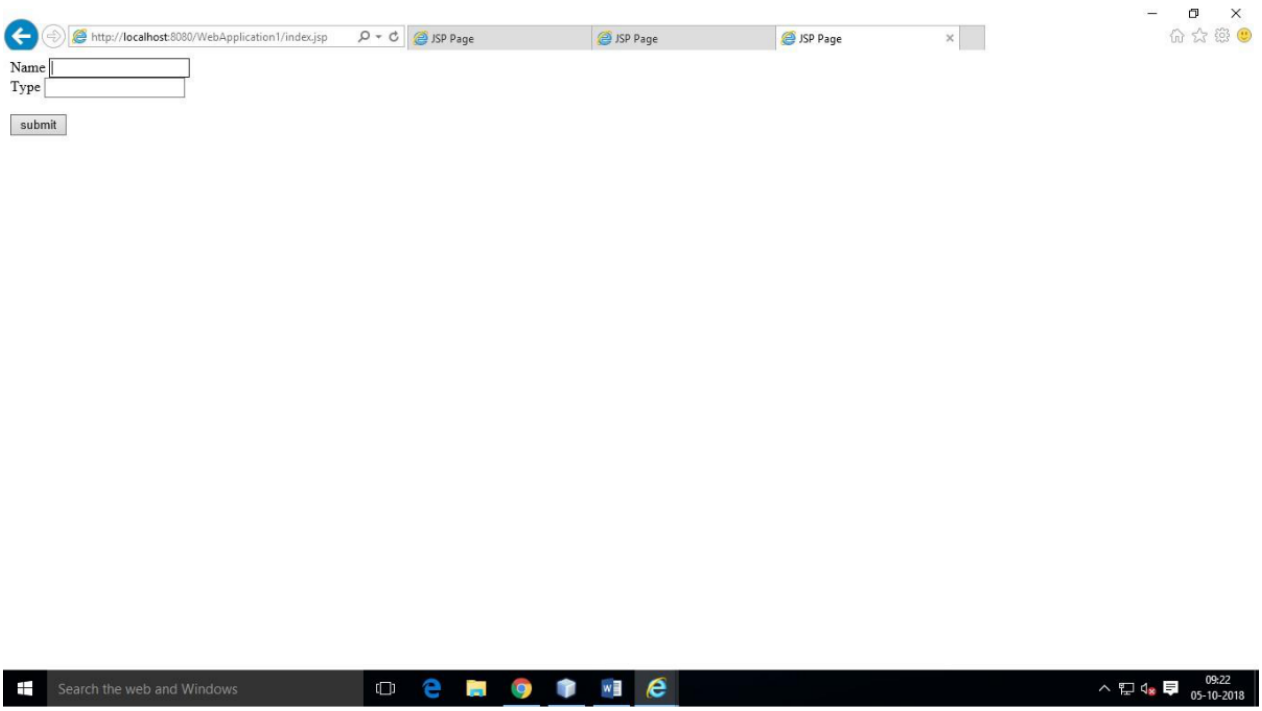
USER.JAVA

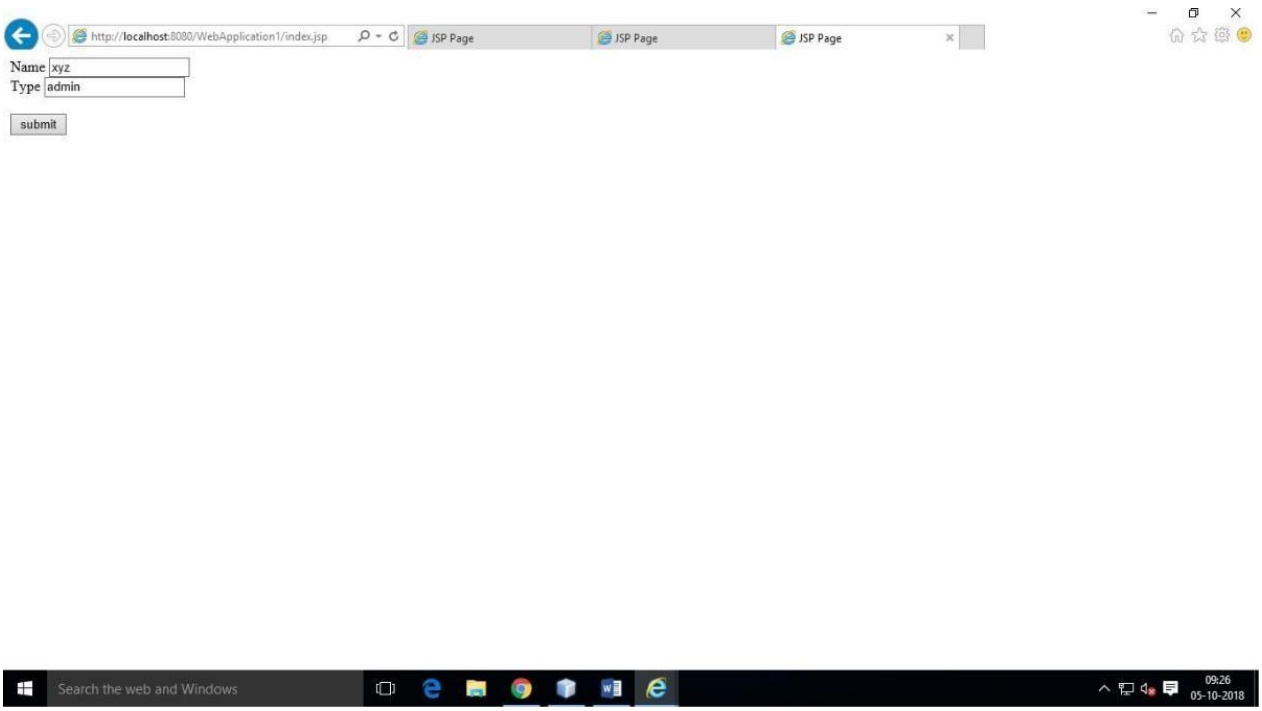
```
package hibernate;

public class User implements java.io.Serializable {
private Integer uid;
private String uname;
private String utype;
public User() {   }

public User(String uname, String utype) {

this.uname = uname;

this.utype = utype;

}

public Integer getUid() {
```

```
return this.uid;

}
public void setUid(Integer uid) {

this.uid = uid;

}
public String getUname()
{
return this.uname;
}
public void setUname(String uname)
{
this.uname = uname;
}
public String getUtype()
{
return this.utype;
}
public void setUtype(String utype)
{
this.utype = utype;
}
}
```

OUTPUT:


http://localhost:8080/WebApplication1/userview.jsp?
JSP Page
JSP Page
Guest View
Guest View Click here to go BACK
5
xyz
admin
6
pqr
user
7
mnq
admin
Search the web and Windows
09:21
05-10-2018


http://localhost:8080/WebApplication1/index.jsp
JSP Page
JSP Page
JSP Page
Name
Type
submit
Search the web and Windows
09:22
05-10-2018


http://localhost:8080/WebApplication1/index.jsp
JSP Page
JSP Page
JSP Page
Name xyz
Type admin
submit
Search the web and Windows
09:26
05-10-2018

Q.9 B) DEVELOP A HIBERNATE APPLICATION TO STORE FEEDBACK OF WEBSITE VISITOR IN MYSQL DATABASE.

CODE:

INDEX.HTML

HIBERNATE – FEEDBACK OF WEBSITE VISITOR (ON INDEX PAPER)

STEP 1: MYSQL COMMAND: -

Select Services → Right click on Database → Connect → Password → ok → Again Right click on Database → Create Database → db → ok. Expand db → Select and Right Click table → click on Execute command →

Create table guestbook (no int primary key auto_increment, name varchar(20), msg varchar(100), dt varchar(40));

STEP 2: CREATE A HIBERNATE PROJECT: -

File → New Project → Java Web → Web application → Next → give the project name → browse the location as required → select the checkbox → "dedicated folder for storing libraries" → Next Select glassfish server → next Select frame work → Hibernate → select the respective database connection → finish.

STEP 3: ADDING REVERSE ENGINEERING FILE :-

Right click on Project → new → other → select Hibernate → Hibernate Reverse Engineering wizard file type → next → file name (hibernate.reveng) , folder → click on browse and select src → java → next → select guestbook table name from the available tables option → click add (select the checkbox – include related files) → finish.

STEP 4: ADDING HIBERNATE MAPPING FILES AND POJOS FROM DATABASE FILE TYPE:-

Right click on Project → new → other → select Hibernate → Hibernate mapping files and POJOs from Database file type) → next → keep the default configuration file name file name (hibernate.cfg) and Hibernate Reverse Engineering File (hibernate.reveng) → type the package name (hibernate) → finish.

STEP 5: CREATING JSP FILE :-

Right click on project → new -> JSP → filename → guestbookview → select radiobutton → JSP file (Standard syntax) → Finish.

GUESTBOOK.JAVA

```
package hibernate;

public class Guestbook implements java.io.Serializable { private Integer no;

private String name;

private String msg;

private String dt;

public Guestbook() {

}

public Guestbook(String name, String msg, String dt) { this.name = name;

this.msg = msg;

this.dt = dt;

}

public Integer getNo() {

return this.no;
```

```
}

public void setNo(Integer no) {

this.no = no;

}
public String getName() {

return this.name;

}

public void setName(String name) {

this.name = name;

}

public String getMsg() {

return this.msg;

}

public void setMsg(String msg) {

this.msg = msg;

}

public String getDt() {

return this.dt;

}

public void setDt(String dt) {

this.dt = dt;

}

}
```

HIBERNATE.CFG.XML

```
<hibernate-configuration>
<session-factory>
<property
name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
<property
name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
<property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/db</property>
<property name="hibernate.connection.username">root</property>
<property name="hibernate.connection.password">tiger</property>
<mapping resource="hibernate/Guestbook.hbm.xml"/>
</session-factory>
</hibernate-configuration>
```

GUESTBOOK.HBM.XML

```
<hibernate-mapping>

<class name="hibernate.Guestbook" table="guestbook" catalog="db"> <id
name="no" type="java.lang.Integer"> <column name="no" />

<generator class="identity" />

</id>

<property name="name" type="string">

<column name="name" length="20" />
```

```
</property>

<property name="msg" type="string">

<column name="msg" length="100" />

</property>

<property name="dt" type="string">

<column name="dt" length="40" />

</property>

</class>

</hibernate-mapping>
```

INDEX.JSP

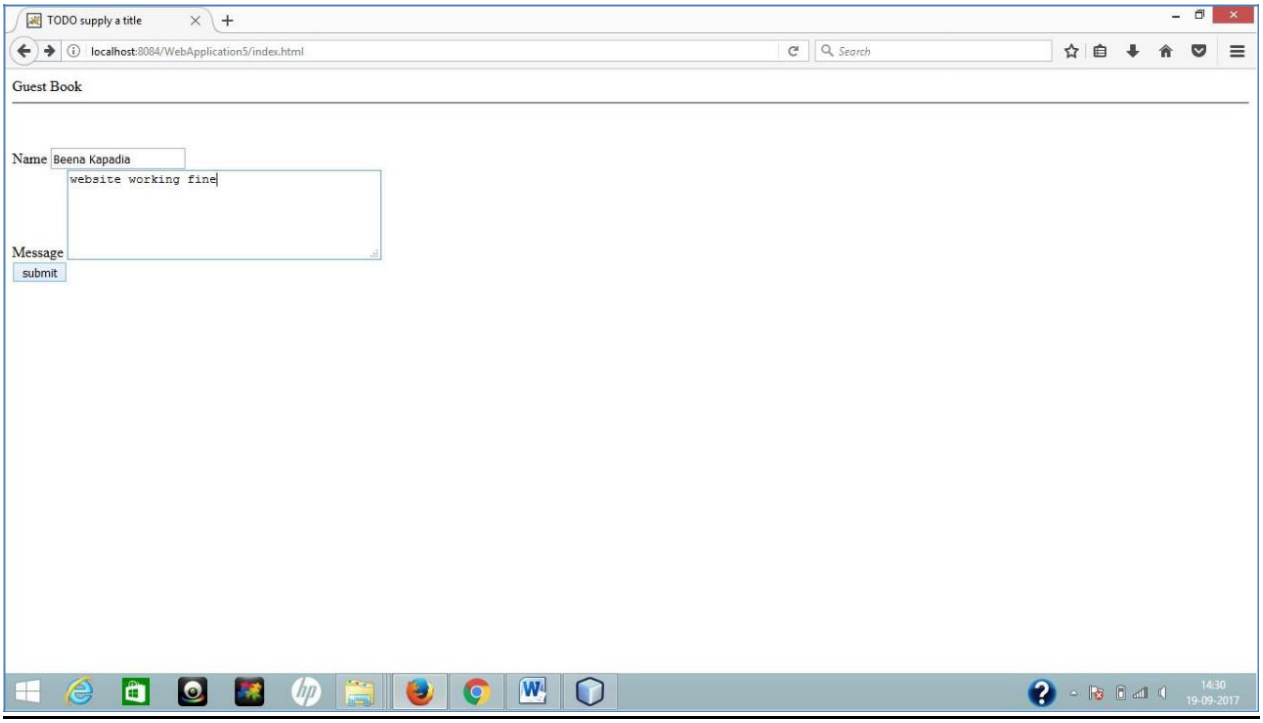
```
<html>

<head>

<title>Guest Book</title>

</head>

<body>
Guest Book <hr><br><br>

<form action="guestbookview.jsp" >

Name <input type="text" name="name" maxlength="20"><br>

Message <textarea rows="5" cols="40" maxlength="100" name="msg"></textarea>

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

</form>

</body>

</html>
```

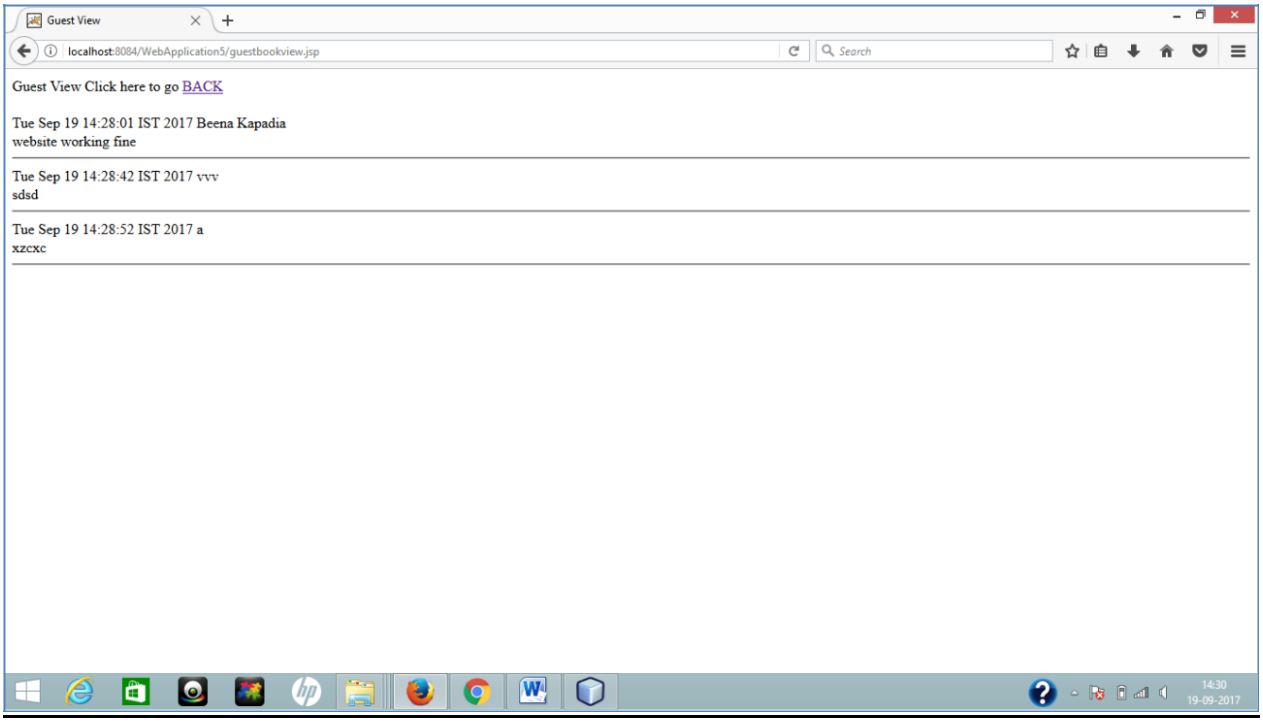
GUESTBOOKVIEW.JSP

```
<%@page import="org.hibernate.SessionFactory"%> <%@page
import="org.hibernate.Session"%> <%@page
import="org.hibernate.cfg.Configuration"%> <%@page
import="org.hibernate.Transaction"%> <%@page import="java.util.List"%> <%@page
import="java.util.Iterator"%>
<%@page import="hibernate.Guestbook"%>

<%!

SessionFactory sf;

org.hibernate.Session ss;

List<hibernate.Guestbook> gbook;

%>

<%

sf = new Configuration().configure().buildSessionFactory();

ss= sf.openSession();

Transaction tx=null;
Guestbook gb=new Guestbook();

try

{
```

```
tx=ss.beginTransaction();
String name=request.getParameter("name");
String msg=request.getParameter("msg");
String dt=new java.util.Date().toString();
gb.setName(name);
gb.setMsg(msg);
gb.setDt(dt);
ss.save(gb);
tx.commit();
}
catch(Exception e){ out.println("Error"+e.getMessage()); } try
{ ss.beginTransaction();
gbook=ss.createQuery("from Guestbook").list();
}
catch(Exception e){ }
%>
<html>
<head>
<title>Guest View</title>
</head>
<body>
Guest View
Click here to go <a href="index.jsp"> BACK </a> <br><br>
<%    Iterator it=gbook.iterator();
while(it.hasNext())
{
Guestbook eachrecord=(Guestbook)it.next();
out.print(eachrecord.getDt()+" ");
out.print(eachrecord.getName()+"<br>");
out.print(eachrecord.getMsg()+"<br><hr>");
}
%>
</body>
</html>
```

OUTPUT:


TODO supply a title
localhost:8084/WebApplication5/index.html
Search
Guest Book
Name Beena Kapadia
website working fine
Message
submit


Guest View
localhost:8084/WebApplication5/guestbookview.jsp
Search
Guest View Click here to go BACK
Tue Sep 19 14:28:01 IST 2017 Beena Kapadia
website working fine
Tue Sep 19 14:28:42 IST 2017 vvv
sdsd
Tue Sep 19 14:28:52 IST 2017 a
xzcxc

Q.9 C) DEVELOP A HIBERNATE APPLICATION TO STORE AND RETRIEVE EMPLOYEE DETAILS IN MYSQL DATABASE.

CODE:

INDEX.HTML

```
<html>

<head>

<title>Employee Details</title>

</head>

<body>

Employee Details <hr><br><br>

<form action="empview.jsp" >

Name <input type="text" name="name" maxlength="20"><br>

Salary <input type="text" name="salary" maxlength="20"><br>

Designation <input type="text" name="designation" maxlength="20">

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

</form>

</body>

</html>
```

EMPVIEW.JSP

```
<%@page import="org.hibernate.SessionFactory"%> <%@page
import="org.hibernate.Session"%> <%@page
import="org.hibernate.cfg.Configuration"%> <%@page
import="org.hibernate.Transaction"%> <%@page import="java.util.List"%> <%@page
import="java.util.Iterator"%>
<%@page import="hibernate.Emp"%>

<%!

SessionFactory sf;

org.hibernate.Session ss;

List<hibernate.Emp> Emplist;

%>

<%

sf = new Configuration().configure().buildSessionFactory();
ss= sf.openSession();

Transaction tx=null;

Emp em=new Emp();

try

{

tx=ss.beginTransaction();

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

String Salary=request.getParameter("salary");

String Designation=request.getParameter("designation");

System.out.print("Name ..............."+Name+" "+Salary+" "+Designation);
```

```
em.setName(Name);
em.setSalary(Salary);
em.setDesignation(Designation);
System.out.print("set done ..... ");
ss.save(em);
System.out.print("save done ..... ");
tx.commit();
System.out.print("commit done ..... ");
}
catch(Exception e){ out.println("Error"+e.getMessage()); }

try
{
ss.beginTransaction();
Emplist=ss.createQuery("from Emp").list();
}
catch(Exception e){ }
%>
<html>
<head>
<title>Employee View</title>
</head>
<body>
Employee View
Click here to go <a href="index.html"> BACK </a> <br><br>
<%
Iterator it=Emplist.iterator();
while(it.hasNext())
{
Emp eachrecord=(Emp)it.next(); out.print(eachrecord.getName()+"<br>");
out.print(eachrecord.getSalary()+"<br><hr>");
out.print(eachrecord.getDesignation()+"<br><hr>"); }
%>
```

HIBERNATE.REVENGE.XML

```
<hibernate-reverse-engineering>
<schema-selection match-catalog="employeedb"/>
<table-filter match-name="emp"/>
</hibernate-reverse-engineering>
```

HIBERNATE1.CFG.XML

```
<hibernate-configuration>

<session-factory>

<property
name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>

<property
name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>

<property

name="hibernate.connection.url">jdbc:mysql://localhost:3306/employeedb?zeroDat
eTimeBehavior=c

onvertToNull</property>

<property name="hibernate.connection.username">root</property>

<property name="hibernate.connection.password">tiger</property>

</session-factory>

</hibernate-configuration>
```

EMP.HBM.XML

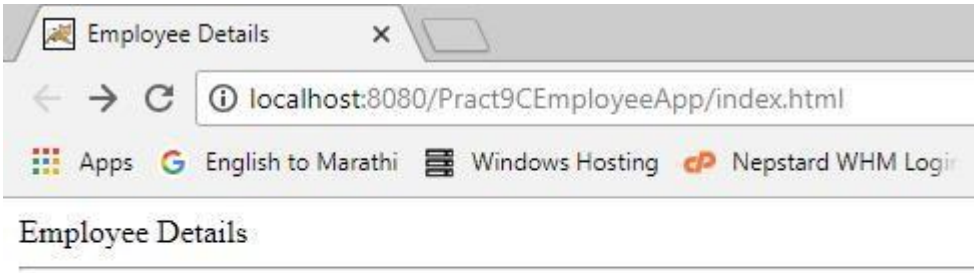
```
<hibernate-mapping>

<class optimistic-lock="version" catalog="employeedb" table="emp"
ame="hibernate.Emp">

<id name="id" type="java.lang.Integer">

<column name="id"/>

<generator class="identity"/>

</id>

<property name="name" type="string">

<column name="name" length="20"/>

</property>

<property name="salary" type="string">

<column name="salary" length="20"/>

</property>

<property name="designation" type="string">

<column name="designation" length="20"/>

</property>

</class>

</hibernate-mapping>
```

EMP.JAVA

```
package hibernate;

public class Emp implements java.io.Serializable {

private Integer id;

private String name;

private String salary;

private String designation;

public Emp() { }
```

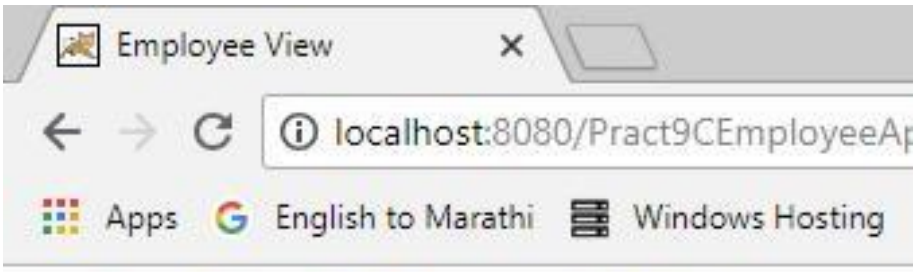
```
public Emp(String name, String salary, String designation) {
this.name = name;
this.salary = salary;
this.designation = designation;
}
public Integer getId() {
return this.id;
}
public void setId(Integer id) {
this.id = id;
}
public String getName() {
return this.name;
}
public void setName(String name) {
this.name = name;
}
public String getSalary() {
return this.salary;
}
public void setSalary(String salary) {
this.salary = salary;
}
public String getDesignation() {
return this.designation;
}
public void setDesignation(String designation) {
this.designation = designation;
}
}
```

OUTPUT:



Employee Details

Name Sarthak
Salary 100000
Designation CEO
submit



Employee View Click here to go BACK

Sarthak
100000

CEO