Practical no-1:
Implement the following Simple Servlet applications.

Q.1 a) Create a simple calculator application using servlet

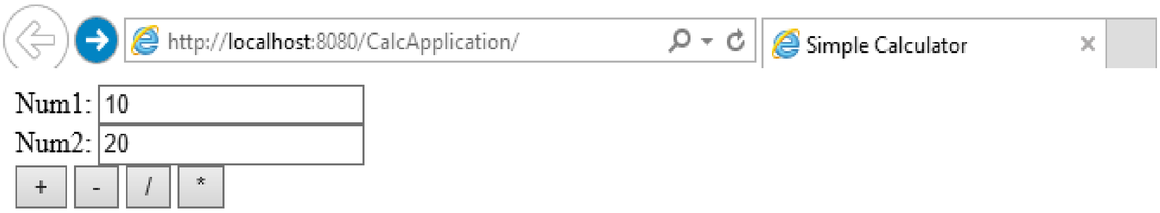
index.html:-

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html><head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Simple Calculator </title>
    </head>
    <body>
        <form method="Post" action=" CalcServlet">
            Num1: <input type="text" name="t1" > <br>
            Num2: <input type="text" name="t2" > <br>
            <input type="submit" value="+" name="btn">
            <input type="submit" value="-" name="btn">
            <input type="submit" value="/" name="btn">
            <input type="submit" value="*" name="btn">
        </form>
    </body>
</html>
```

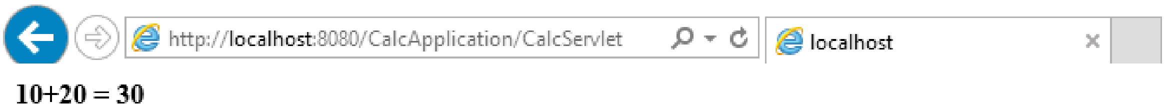
CalcServlet.java:-

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class CalcServlet extends HttpServlet
{
    public void doPost(HttpServletRequest req,HttpServletResponse res) throws
ServletException,IOException
    {
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();
        int a = Integer.parseInt(req.getParameter("t1"));
        int b = Integer.parseInt(req.getParameter("t2"));
        String op = req.getParameter("btn");
        int c = 0;
        if(op.equals("+"))
        {
            c = a+b;
        }
        else if(op.equals("-"))
        {
            c=a-b;
        }
        else if(op.equals("*"))
        {
            c=a*b;
        }else if(op.equals("/"))
        {
            c=a/b;
        }
        out.print("<b>"+a+op+b+" = "+c+"</b>");
    }
}
```

Output:



Num1: 10
Num2: 20



10+20 = 30

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

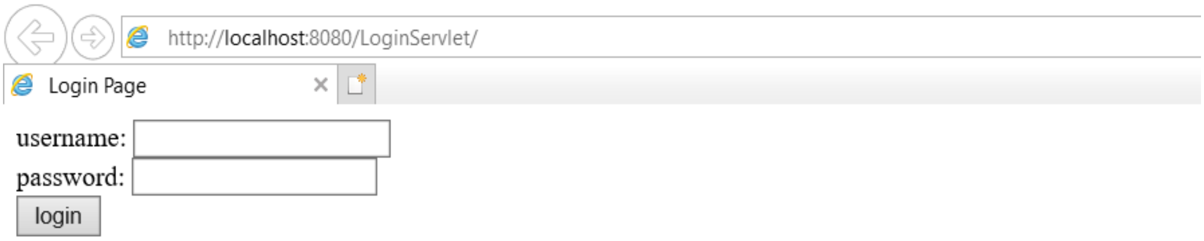
<u>index.jsp:-</u>

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>Login Page</title>
   </head>
   <body>
      <form method="post" action="LoServlet">
         username: <input type="text" name="un"><br>
         password: <input type="password" name="pw"><br>
         <input type="submit" value="login">
      </form>
   </body>
</html>
```

<u>LoServlet:-</u>

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class LoServlet extends HttpServlet
{
   protected void doPost(HttpServletRequest req, HttpServletResponse res)
         throws ServletException, IOException {
      res.setContentType("text/html");
      PrintWriter out = res.getWriter();
      String username = req.getParameter("un");
      String password = req.getParameter("pw");
      if((username.equals("admin"))&&(password.equals("admin123")))
      {
         out.print("Hello Ab");
      }
      else
      {
         out.print("login failed...!!");
      }
   }
}
```

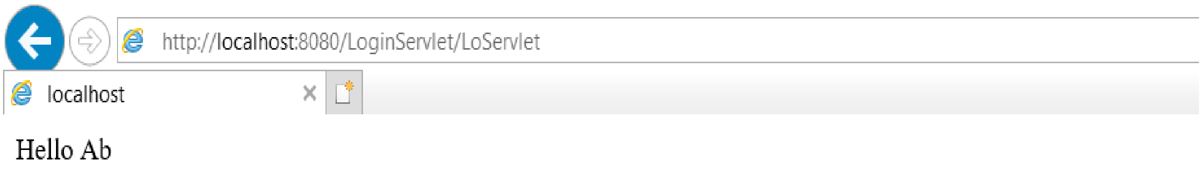
Output:


http://localhost:8080/LoginServlet/

Login Page

username:

password:


http://localhost:8080/LoginServlet/LoServlet

localhost

Hello Ab

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.

MySql Command from mysql software:-

1. Select **services ->** expand **databases** -> right click on **MySQL server at localhost:3306[disconnected] ->** click on **connect ->** enter password **(tiger)** -> OK
2. Again right click on **MySQL server at localhost:3306 ->** select **Create database ->** enter database name and select the check box to grant permission.
3. Right click on **Table** under your daatbase
4. Enter table name user by replacing untitled. Click on **Add column,** name -> username, type-> varchar, size-> 20, select checkbox of primary key, again click on **Add column** password varchar size 20, again click on **Add column** emailid varchar(20), again click **Add column** country varchar 10;
5. add mysql-connector to library folder of the current application

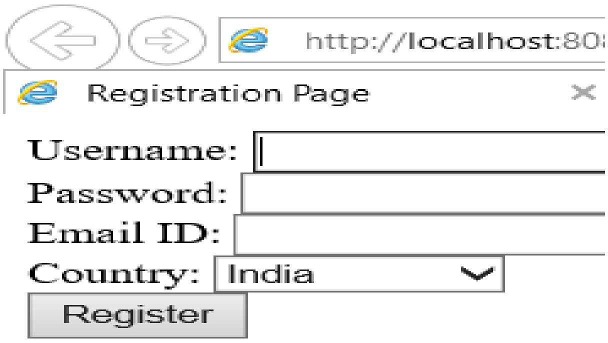
index.html:-

```
<html><head><title>New User Registration Page</title></head>
   <body>
      <form action="Register.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>France</option>
            <option>England</option>
            <option>Argentina</option>
            </select><br>
        <input type="submit" value="REGISTER" > <input type="RESET" >
      </form>
   </body>
</html>
```

Register.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/logindb","root","tiger");
         PreparedStatement stmt = con.prepareStatement("insert into user 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>
```

Output:



http://localhost:80

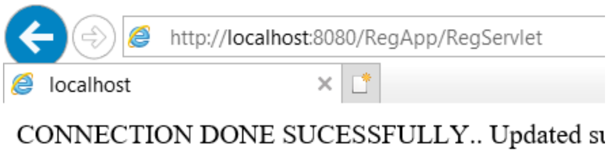
Registration Page

Username:

Password:

Email ID:

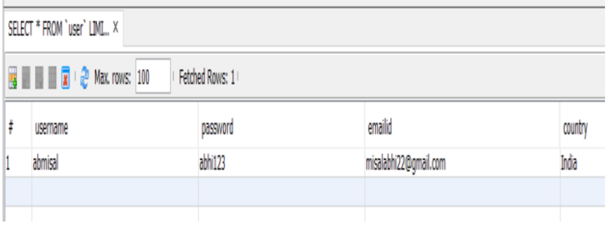
Country: India



http://localhost:8080/RegApp/RegServlet

localhost

CONNECTION DONE SUCESSFULLY.. Updated su



SELECT * FROM `user` LIMI... X

Max. rows: 100 Fetched Rows: 1

#	username	password	emailid	country
1	abmisal	abhi123	misalabhi22@gmail.com	India

Practical no:-2

Implement the following 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 Design a web page with different tables

index.jsp:-

```
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>Request Dispatcher</title>
   </head>
   <body>
      <form action="Validate" method="POST">
         Username: <input type="text" name="un" value="" /><br>
         Password: <input type="password" name="pwd" value="" /> <br>
         <input type="submit" value="Login" />
      </form>
   </body>
</html>
```

Validate.java:-

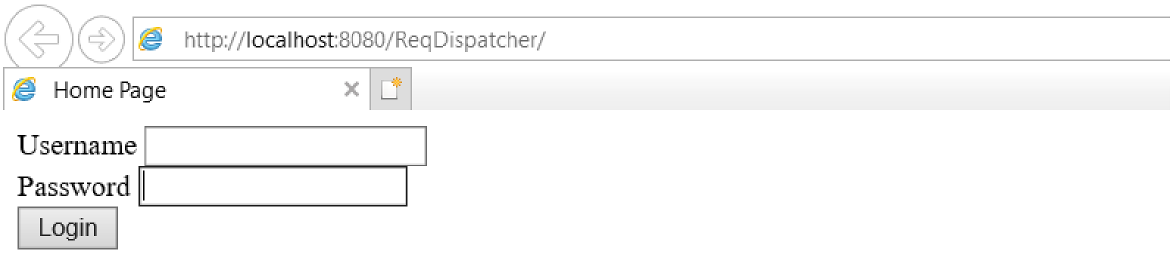
```
package servlet;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class Validate extends HttpServlet
{
   @Override
   protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
      resp.setContentType("text/html");
      PrintWriter out = resp.getWriter();
      String un = req.getParameter("un");
      String pass=req.getParameter("pwd");
      ServletContext sc = getServletContext();
      if(pass.equals("servlet"))
      {
         req.setAttribute("username", un);
         RequestDispatcher rd = sc.getRequestDispatcher("/Welcome");
         rd.forward(req, resp);
      }
      else
      {
         RequestDispatcher rd = sc.getRequestDispatcher("/index.jsp");
         out.println("Invalid Password");
         rd.include(req, resp);
      }
   }
}
```

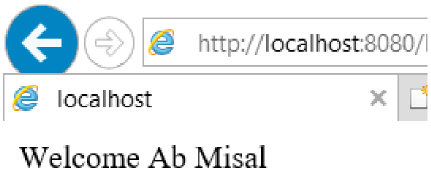
Welcome.java:-

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

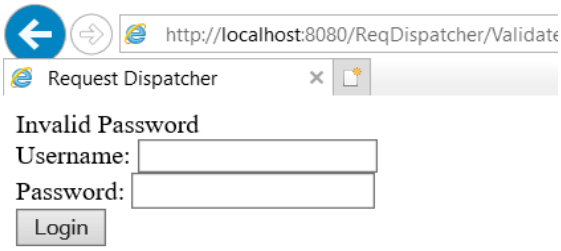
```
public class Welcome extends HttpServlet
{
   @Override
   protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
      resp.setContentType("text/html");
      PrintWriter out = resp.getWriter();
      String un=(String)req.getAttribute("username");
      out.print("Welcome "+un);
   }
}
```

Output:





Welcome Ab Misal



Invalid Password

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

index.jsp:-

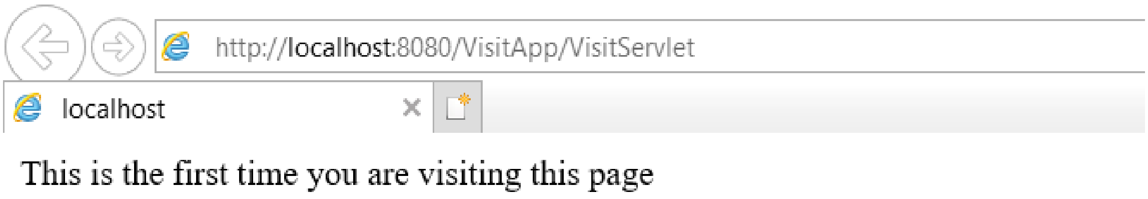
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>JSP Page</title>
   </head>
   <body>
      <h1>Hello World!</h1>
   </body>
</html>
```

CookieServlet:-

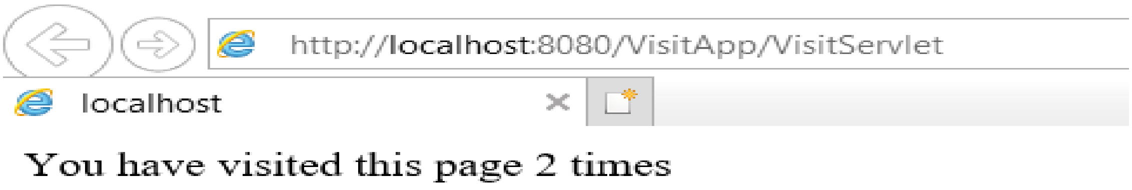
```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class CookieServlet extends HttpServlet
{
   static int i=1;
```

```
    @Override
    public void doGet(HttpServletRequest req,HttpServletResponse res)
            throws IOException, ServletException
    {
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();
        String k= String.valueOf(i);
        Cookie c=new Cookie("Visit",k);
        int j=Integer.parseInt(c.getValue());
        if(j==1)
        {
            out.println("this is the first time you are visiting this page");
        }
        else
        {
            synchronized(this)
            {
                out.println("you visited this page"+i+"times");
            }
        }
        i++;
    }
}
```

<u>Output:</u>



This is the first time you are visiting this page



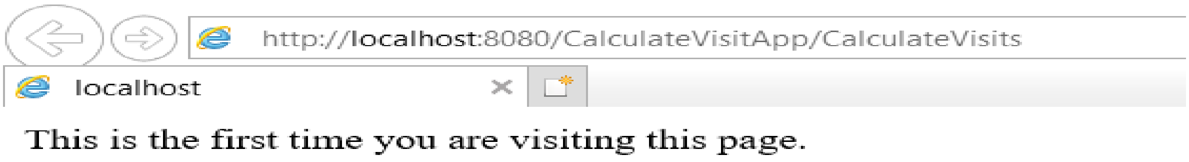
You have visited this page 2 times

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.

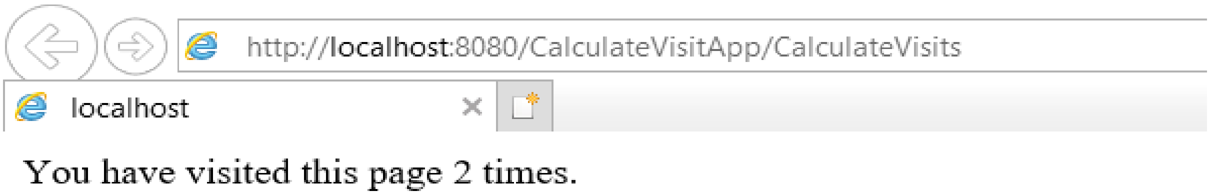
<u>VisitServlet.java:-</u>

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class VisitServlet extends HttpServlet
{
    private int counter;
    public void doGet(HttpServletRequest req,HttpServletResponse res)throws
ServletException,IOException
    {
        res.setContentType("text/html");
       PrintWriter out = res.getWriter();
        HttpSession session=req.getSession(true);
        if(session.isNew())
        {
            out.print("This is the first time you are visiting this page");
            ++counter;
        }
        else
        {
            synchronized(this)
            {
                if(counter==10)
                {
                    session.invalidate();
                    counter=0;
                    req.getSession(false);
                }
                else
                    out.print("You have visited this page "+(++counter)+" times");
            }
        }
    }
}
```

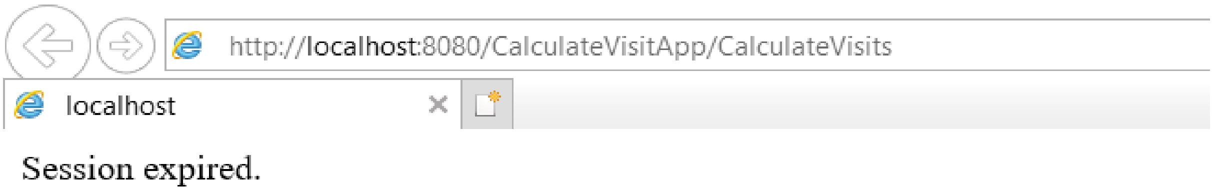
Output:



This is the first time you are visiting this page.



You have visited this page 2 times.



Session expired.

Practical no:-3

Implement the Servlet IO and File applications.

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

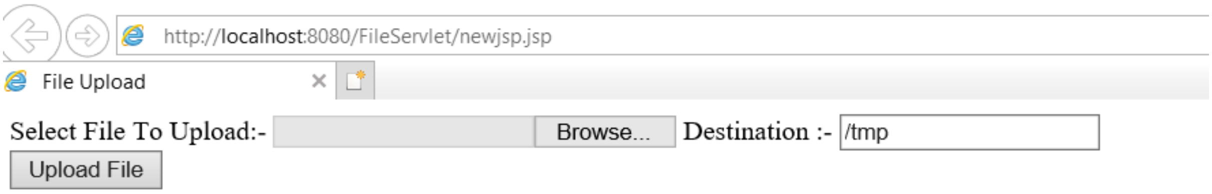
indexd.jsp:-

```
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>Download File</title>
   </head>
   <body>
      <h1> File Download App</h1>
      Click <a href="DownloadServlet?filename=SampleChapter.pdf">Sample Chapter</a>
      <br/><br/>
      Click <a href="DownloadServlet?filename=TOC.pdf">Table Of Contents</a>
   </body>
</html>
```

DownloadServlet.java:-

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class DownloadServlet extends HttpServlet
{
   public void doGet(HttpServletRequest req,HttpServletResponse res)throws
IOException,ServletException
   {
      res.setContentType("APPLICATION/OCTET-STREAM");
      String filename = req.getParameter("filename");
      ServletContext context = getServletContext();
      InputStream is= context.getResourceAsStream("/"+filename);
      ServletOutputStream os= res.getOutputStream();
      res.setHeader("Content-Disposition","attachment;filename=\""+filename+"\"");
      int i;
      byte b[]=new byte[1024];
      while ((i=is.read(b))!=-1)
      {
         os.write(b);
      }
      is.close();
      os.close();
   }
}
```

Output:



http://localhost:8080/FileServlet/newjsp.jsp

File Upload

Select File To Upload:- Destination :- /tmp

indexu.jsp:-

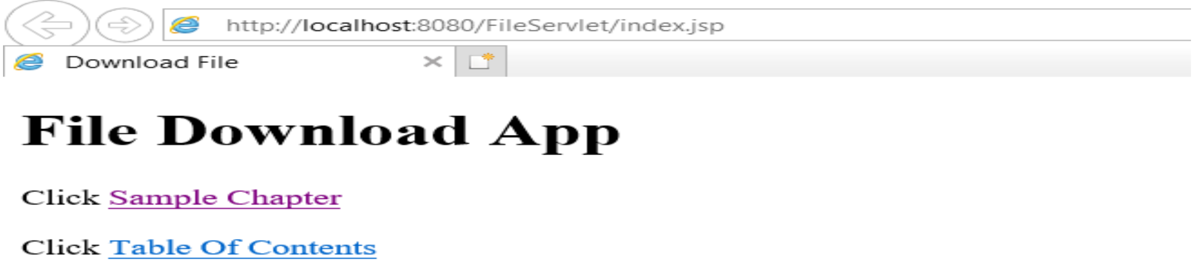
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
```

```
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>File Upload</title>
   </head>
   <body>
      <form action="UploadServlet" method="post" enctype="multipart/form-data">
         Select File To Upload:- <input type="file" name="f1" >
         Destination :- <input type="text" value="/tmp" name="destination">
         <br>
         <input type="submit" value="Upload File" name="Upload">
      </form>
   </body>
</html>
```

UploadServlet.java:-

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.annotation.MultipartConfig;
import javax.servlet.http.*;
@MultipartConfig
public class UploadServlet extends HttpServlet
{
   public void doPost(HttpServletRequest req,HttpServletResponse res)throws
IOException,ServletException
   {
      res.setContentType("text/html");
      PrintWriter out = res.getWriter();
      String path=req.getParameter("destination");
      Part filePart = req.getPart("f1");
      String sfilePart = req.getPart("f1").toString();
      out.print("<br>FilePart:"+sfilePart);
      String filename = filePart.getName().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 successfully...!!!");
      }
     catch(FileNotFoundException e)
      {
         out.print(e);
      }
   }
}
```

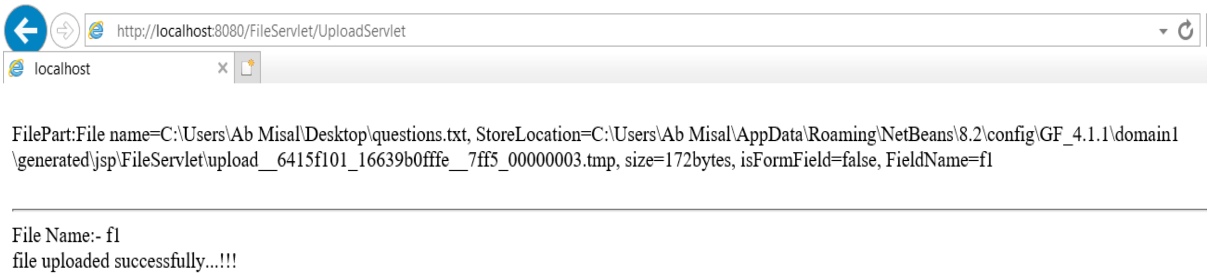
Output:



File Download App

Click Sample Chapter

Click Table Of Contents



```
FilePart:File name=C:\Users\Ab Misal\Desktop\questions.txt, StoreLocation=C:\Users\Ab Misal\AppData\Roaming\NetBeans\8.2\config\GF_4.1.1\domain1
\generated\jsp\FileServlet\upload__6415f101_16639b0fffe__7ff5_00000003.tmp, size=172bytes, isFormField=false, FieldName=f1

File Name:- f1
file uploaded successfully...!!!
```

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

MySql Command from mysql software:-

1. create database queansdb;
2. create table queans(qno int primary key, que varchar(200), op1 varchar(100), op2 varchar(100), op3 varchar(100), op4 varchar(100), ans varchar(3));
3. insert few records.
4. add mysql-connector to library folder of the current application

Index.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>Hello World!</h1>
   </body>
</html>
```

Marks.java:-

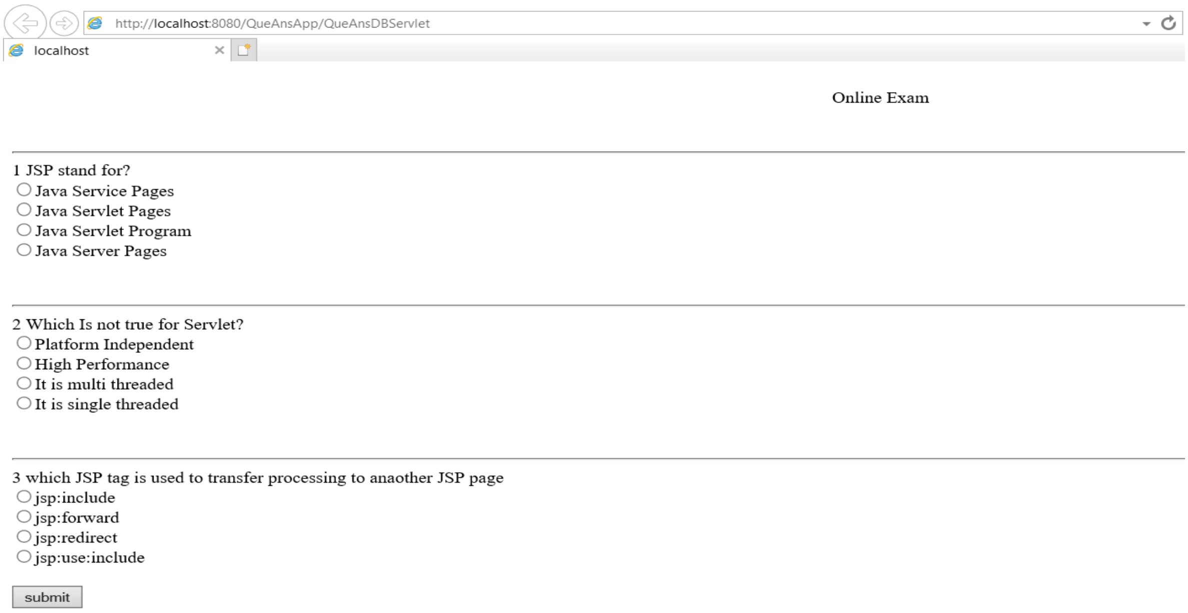
```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class Marks extends HttpServlet
{
  @Override
  public void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException
  {
     response.setContentType("text/html");
     PrintWriter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

QueAnsDBServlet.java:-

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.sql.*;
public class QueAnsDBServlet extends HttpServlet
{
   public void doGet(HttpServletRequest request, HttpServletResponse response)  throws
ServletException, IOException
  {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    try
    {
      out.print("<html><body><br>");
      out.println("<form method='post' action='Marks'>");
      Class.forName("com.mysql.jdbc.Driver");
      Connection
con=DriverManager.getConnection("jdbc:mysql://localhost/queansdb","root","tiger");
      Statement st = con.createStatement();
      String sql="select * from queans";
      ResultSet rs = st.executeQuery(sql);
      int i=0;
      out.print("<center>Online Exam</center>");
      while(rs.next())
        {
          i++;
         out.print("<br><br><hr>"+rs.getInt(1)+"     ");
         out.print(rs.getString(2));
         out.print("<br><input type=radio name="+i+" value=1>"+rs.getString(3));
         out.print("<br><input type=radio name="+i+" value=2>"+rs.getString(4));
         out.print("<br><input type=radio name="+i+" value=3>"+rs.getString(5));
         out.print("<br><input type=radio name="+i+" value=4>"+rs.getString(6));
         String ans="ans"+i;
```

```
            out.println("<br><input type=hidden name="+ans+" value="+rs.getString(7)+">");
          }
            out.println("<br><input type=hidden name=total value="+i+">");
            out.println("<input type=submit value=submit>");
            out.println("</form>");
            out.print("</body></html>");
    }
catch(Exception e)
    {
       out.println("ERROR "+e.getMessage());
     }
  }
}
```

Output:

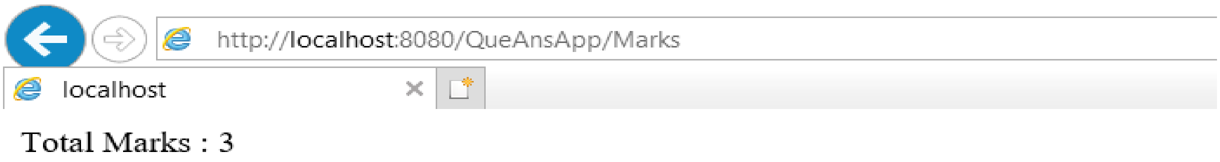


Online Exam

1 JSP stand for?
○ Java Service Pages
○ Java Servlet Pages
○ Java Servlet Program
○ Java Server Pages

2 Which Is not true for Servlet?
○ Platform Independent
○ High Performance
○ It is multi threaded
○ It is single threaded

3 which JSP tag is used to transfer processing to anaother JSP page
○ jsp:include
○ jsp:forward
○ jsp:redirect
○ jsp:use:include



Total Marks : 3

Q.3 c) Create simple Servlet application to demonstrate NonBlocking Read Operation.

Index.jsp:-

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Non-Blocking Page</title>
    </head>
    <body>
        <a href="NonBlockingServlet"> Non-Blocking </a>
    </body>
</html>
```

NonBlockingServlet.java:-

```
import java.io.*;
import java.net.HttpURLConnection;
import java.net.URL;
import javax.servlet.*;
import javax.servlet.http.*;

public class NonBlockingServlet extends HttpServlet
{
   protected void service(HttpServletRequest req,HttpServletResponse res)throws
ServletException,IOException
   {
      res.setContentType("text/html");
      PrintWriter out = res.getWriter();
      String filename = "booklist.txt";
      ServletContext c = getServletContext();
      InputStream is = c.getResourceAsStream("/"+filename);
      InputStreamReader isr = new InputStreamReader(is);
      BufferedReader br = new BufferedReader(isr);
      String path =
"http://"+req.getServerName()+":"+req.getServerPort()+req.getContextPath()+"/ReadingNon
BlockingServlet";
      out.println("<h1>FileReader/</h1>");
      URL url = new URL(path);
      HttpURLConnection hc = (HttpURLConnection)url.openConnection();
      hc.setChunkedStreamingMode(2);
      hc.setDoOutput(true);
      hc.connect();
      String text = "";
      out.println("Reading Started ...");
      BufferedWriter bw = new BufferedWriter(new
OutputStreamWriter(hc.getOutputStream()));
      while((text = br.readLine())!= null)
      {
         bw.write(text);
         bw.flush();
         out.println(text+"<br>");
         out.flush();
         try
         {
            Thread.sleep(1000);
         }
         catch(Exception ex)
         {
            out.print(ex);
         }
      }
      bw.write("Reading Completed");
      bw.flush();
      bw.close();
   }
}
```

ReadingListener:-

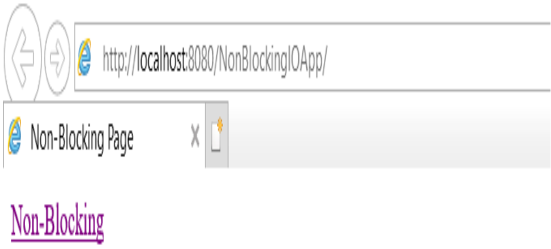
```
import java.io.*;
import javax.servlet.*;
import javax.servlet.AsyncContext;
public class ReadingListener implements ReadListener
```

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

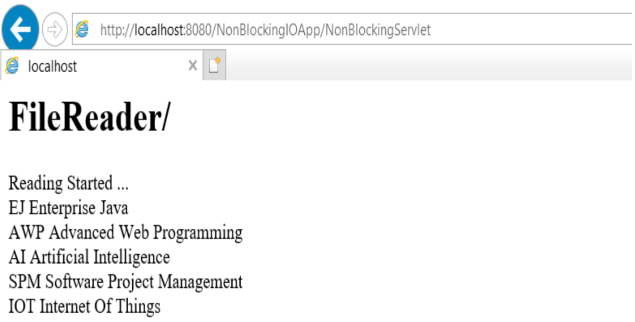
ReadingNonBlockingServlet.java:-

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.annotation.WebServlet;
@WebServlet(name ="ReadingNonBlockingServlet",urlPatterns =
{"/ReadingNonBlockingServlet"},asyncSupported = true)
public class ReadingNonBlockingServlet extends HttpServlet
{
   @Override
   protected void service(HttpServletRequest req,HttpServletResponse res)
         throws ServletException,IOException{
      res.setContentType("text/html");
      AsyncContext ac=req.startAsync();
      ServletInputStream in=req.getInputStream();
      in.setReadListener(new ReadingListener(in,ac));
   }
}
```

Output:



Non-Blocking



FileReader/

Reading Started ...
EJ Enterprise Java
AWP Advanced Web Programming
AI Artificial Intelligence
SPM Software Project Management
IOT Internet Of Things

Practical no:-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.

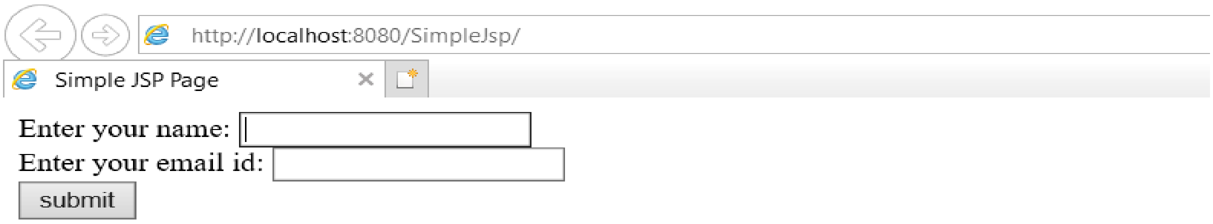
index.jsp:-

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
   </head>
   <body>
      <h1>Use of Intrinsic Objects in JSP</h1>
      <h1>Request Object </h1>
      Query String:  <%=request.getQueryString() %><br>
      Context Path:  <%=request.getContextPath() %><br>
      Remote Host:  <%=request.getRemoteHost()  %><br>
      <h1>Response Object </h1>
      Character Encoding Type: <%=response.getCharacterEncoding() %><br>
      Content Type:       <%=response.getContentType()  %><br>
      Locale:             <%=response.getLocale()  %><br>
      <h1>Session Object </h1>
      ID: <%=session.getId() %><br>
      Creation Time:  <%=new java.util.Date(session.getCreationTime()) %><br>
      Last Access Time:   <%=new java.util.Date(session.getLastAccessedTime())  %><br>
   </body>
</html>
```

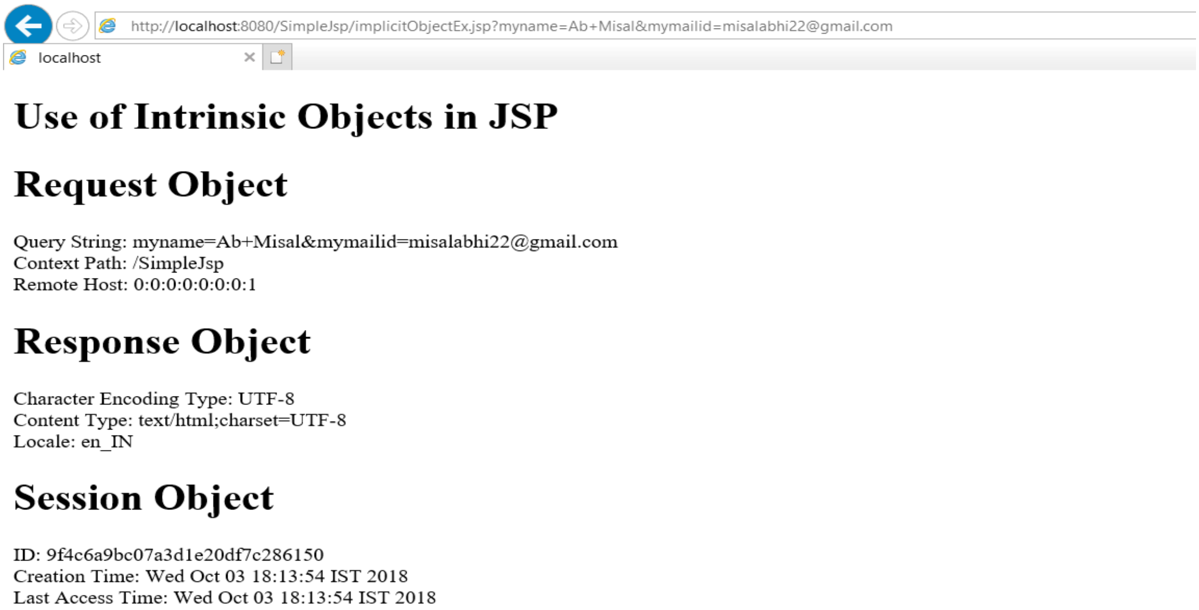
implicitObjectEx.jsp:-

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
   </head>
   <body>
      <h1>Use of Intrinsic Objects in JSP</h1>
      <h1>Request Object </h1>
      Query String:  <%=request.getQueryString() %><br>
      Context Path:  <%=request.getContextPath() %><br>
      Remote Host:  <%=request.getRemoteHost()  %><br>
      <h1>Response Object </h1>
      Character Encoding Type: <%=response.getCharacterEncoding() %><br>
      Content Type:       <%=response.getContentType()  %><br>
      Locale:             <%=response.getLocale()  %><br>
      <h1>Session Object </h1>
      ID: <%=session.getId() %><br>
      Creation Time:  <%=new java.util.Date(session.getCreationTime()) %><br>
      Last Access Time:   <%=new java.util.Date(session.getLastAccessedTime())  %><br>
   </body>
</html>
```

Output:



Enter your name:
Enter your email id:
submit



http://localhost:8080/SimpleJsp/implicitObjectEx.jsp?myname=Ab+Misal&mymailid=misalabhi22@gmail.com
localhost

Use of Intrinsic Objects in JSP

Request Object

Query String: myname=Ab+Misal&mymailid=misalabhi22@gmail.com
Context Path: /SimpleJsp
Remote Host: 0:0:0:0:0:0:0:1

Response Object

Character Encoding Type: UTF-8
Content Type: text/html;charset=UTF-8
Locale: en_IN

Session Object

ID: 9f4c6a9bc07a3d1e20df7c286150
Creation Time: Wed Oct 03 18:13:54 IST 2018
Last Access Time: Wed Oct 03 18:13:54 IST 2018

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).

index.jsp:-

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>Validate JSP Page</title>
   </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 E-mail:- <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>
```

Validate.jsp:-

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"
import="mypack.*">
    </head>
    <body>
        <h1>Validation Page</h1>
            <jsp:useBean id="obj"  scope="request"
                    class="mypack.CheckerBean"      >
            <jsp:setProperty name="obj" property="*"/>
            </jsp:useBean>
            <%if (obj.validate())
            { %>
            <jsp:forward page="successful.jsp"/>
            <%   }
            else  {%>
            <jsp:include page="index.html"/>
            <%}%>
            <%=obj.getError() %>
    </body>
</html>
```

Succesful.jsp:-

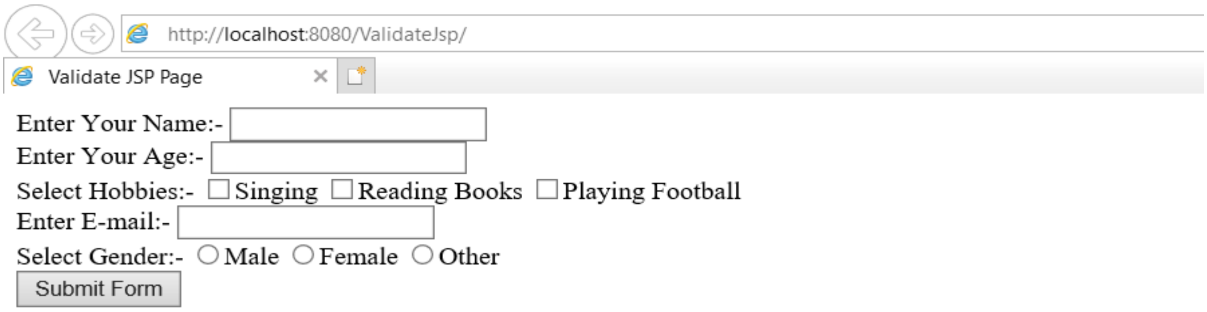
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    </head>
    <body>
            <h1>DATA VALIDATED SUCCESSFULLY </h1>
    </body>
</html>
```

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;
    }
}
```

Output:



http://localhost:8080/ValidateJsp/

Validate JSP Page

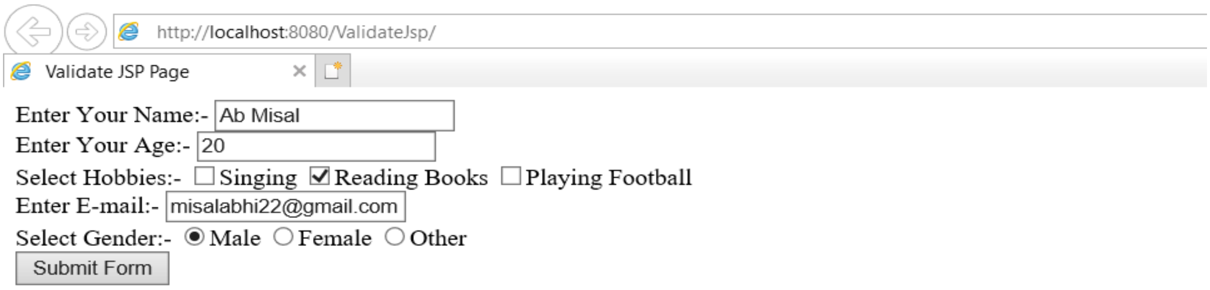
Enter Your Name:-

Enter Your Age:-

Select Hobbies:- □ Singing □ Reading Books □ Playing Football

Enter E-mail:-

Select Gender:- ○ Male ○ Female ○ Other



http://localhost:8080/ValidateJsp/

Validate JSP Page

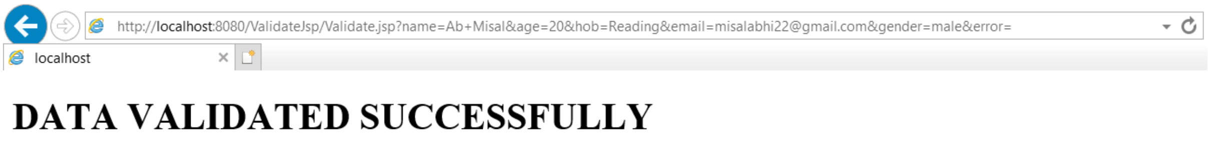
Enter Your Name:- Ab Misal

Enter Your Age:- 20

Select Hobbies:- □ Singing ☑ Reading Books □ Playing Football

Enter E-mail:- misalabhi22@gmail.com

Select Gender:- ● Male ○ Female ○ Other



http://localhost:8080/ValidateJsp/Validate.jsp?name=Ab+Misal&age=20&hob=Reading&email=misalabhi22@gmail.com&gender=male&error=

localhost

DATA VALIDATED SUCCESSFULLY

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

index.html:-

```
<html><head><title>New User Registration Page</title></head>
    <body>
        <form action="Register.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>France</option>
            <option>England</option>
            <option>Argentina</option>
            </select><br>
        <input type="submit" value="REGISTER" > <input type="RESET" >
      </form>
   </body>
</html>
```

Login.html:-

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

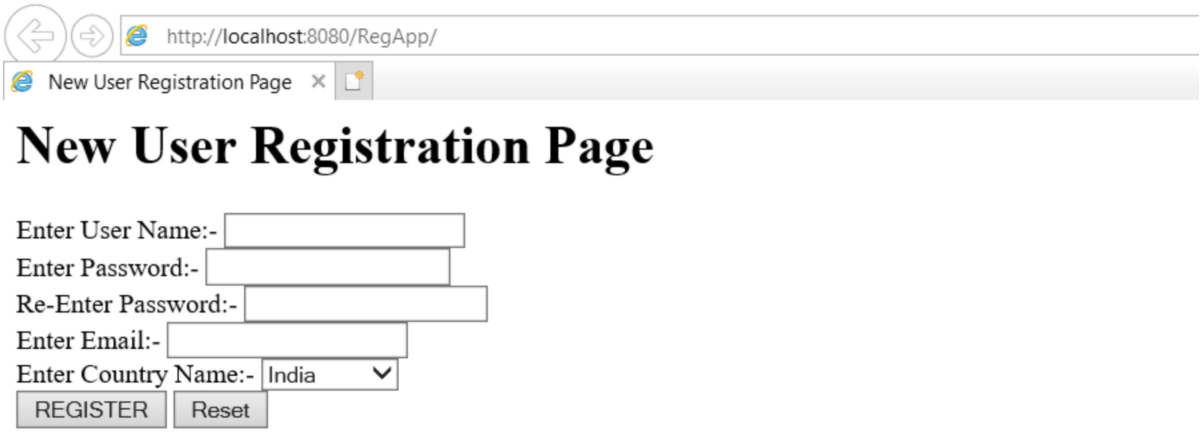
Login.jsp:-

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

Register.jsp:-

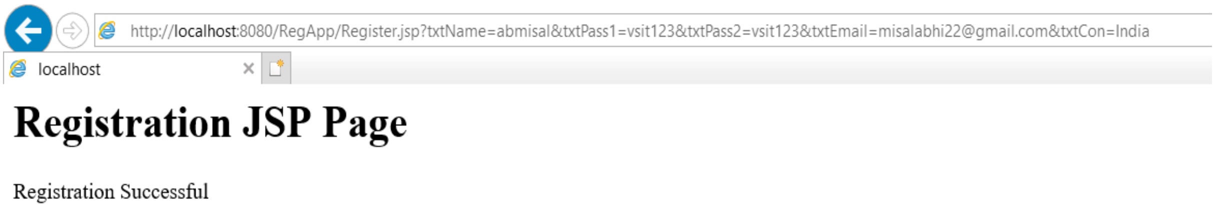
```
<%@page contentType="text/html" import="java.sql.*"%>
<html><body>
        <h1>Registration JSP Page</h1>
    <%
        String uname=request.getParameter("txtName");
        String pass1 = request.getParameter("txtPass1");
        String pass2 = request.getParameter("txtPass2");
        String email = 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/logindb","root","tiger");
            PreparedStatement stmt = con.prepareStatement("insert into user 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>
```

Output:



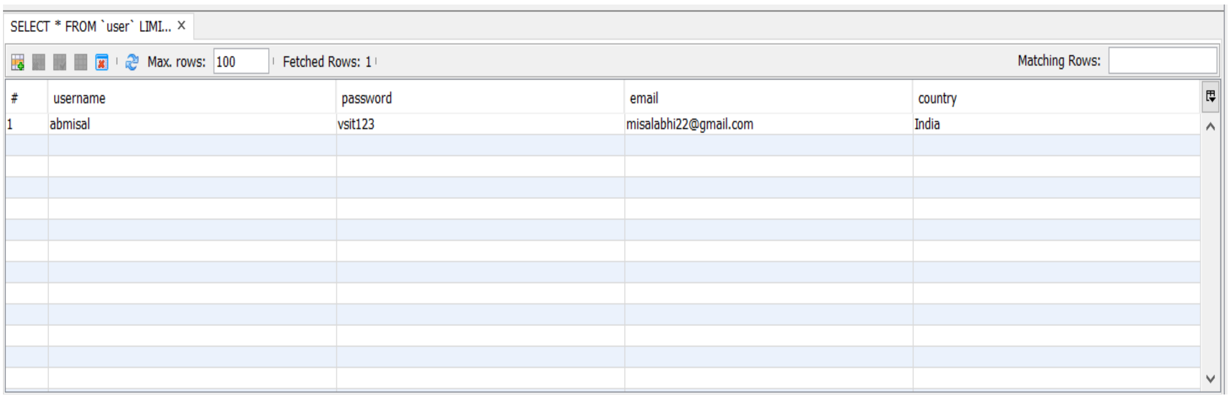
New User Registration Page

Enter User Name:-
Enter Password:-
Re-Enter Password:-
Enter Email:-
Enter Country Name:- India

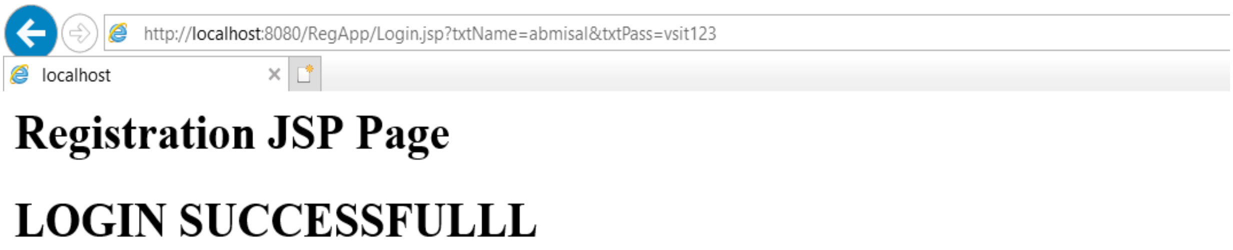

http://localhost:8080/RegApp/Register.jsp?txtName=abmisal&txtPass1=vsit123&txtPass2=vsit123&txtEmail=misalabhi22@gmail.com&txtCon=India
localhost

Registration JSP Page

Registration Successful



#	username	password	email	country
1	abmisal	vsit123	misalabhi22@gmail.com	India


http://localhost:8080/RegApp/Login.jsp?txtName=abmisal&txtPass=vsit123
localhost

Registration JSP Page

LOGIN SUCCESSFULLL

Practical no:-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

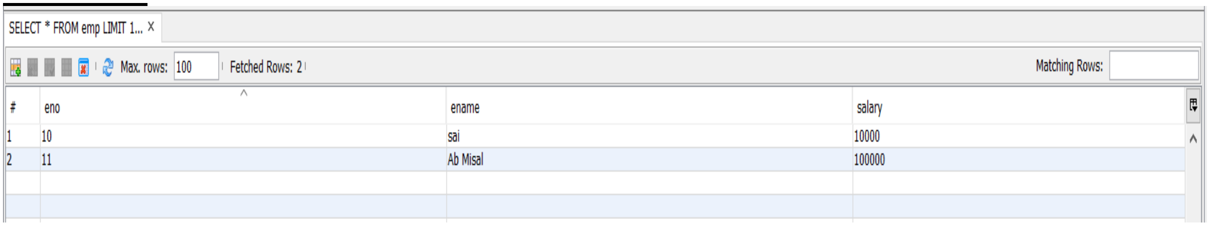
Index.html:-

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

UpdateEmp.jsp:-

```
<%@page contentType="text/html" import="java.sql.*" %>
<html>
   <body>
      <h1>Updating Employee Record</h1>
      <%
      String eno=request.getParameter("txtEno");
      String sal = request.getParameter("txtSal");
     try{
      Class.forName("com.mysql.jdbc.Driver");
      Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/empdb","root","tiger");
      PreparedStatement stmt = con.prepareStatement("select * from emp where empno=?");
      stmt.setString(1, eno);
      ResultSet rs = stmt.executeQuery();
      if(rs.next()){
      out.println("<h1> Employee "+rs.getString(2)+" Exist </h1>");
      PreparedStatement pst= con.prepareStatement("update emp set salary=? where
empno=?");
      pst.setString(1, sal);
      pst.setString(2, eno);
      pst.executeUpdate();
      out.println("<h1>Employee Record updated !!!!!</h1>");
   }
   Else
   {
          out.println("<h1>Employee Record not exist !!!!!</h1>");
    }
      }
catch(Exception e){out.println(e);}
  %>
</body>
</html>
```

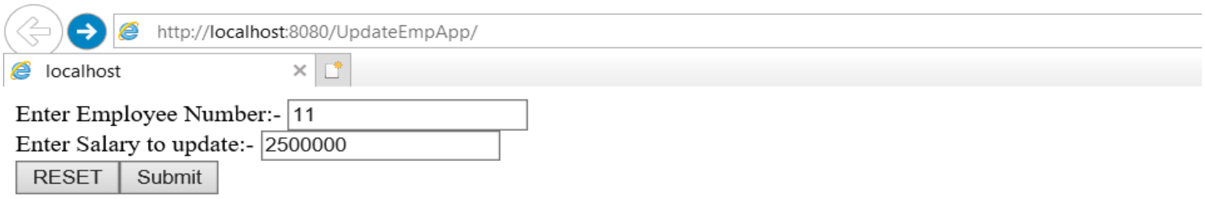
OUTPUT:



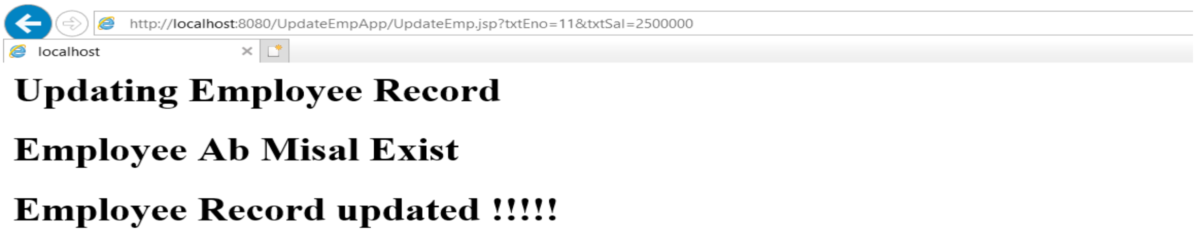
SELECT * FROM emp LIMIT 1... ×

Max. rows: 100 Fetched Rows: 2 Matching Rows:

#	eno	ename	salary
1	10	sai	10000
2	11	Ab Misal	100000



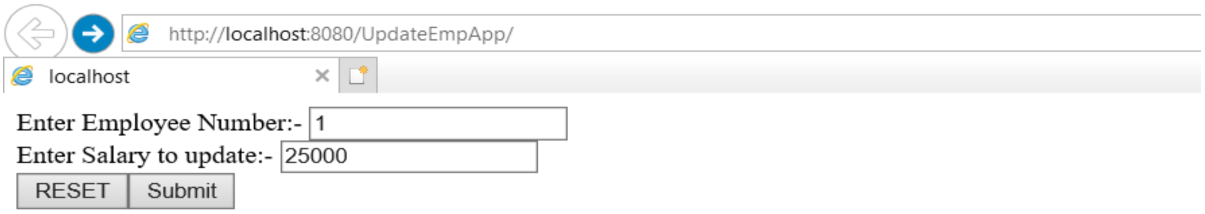
Enter Employee Number:- 11
Enter Salary to update:- 2500000


http://localhost:8080/UpdateEmpApp/UpdateEmp.jsp?txtEno=11&txtSal=2500000
localhost

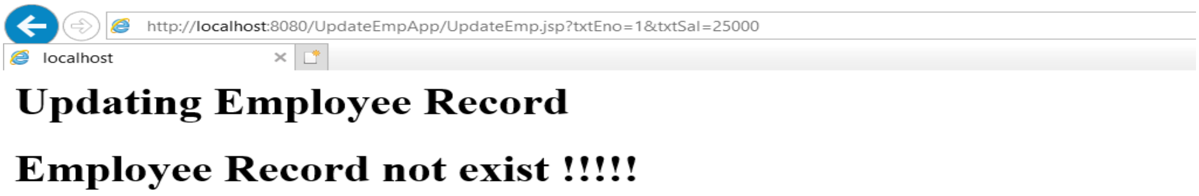
Updating Employee Record

Employee Ab Misal Exist

Employee Record updated !!!!!



Enter Employee Number:- 1
Enter Salary to update:- 25000


http://localhost:8080/UpdateEmpApp/UpdateEmp.jsp?txtEno=1&txtSal=25000
localhost

Updating Employee Record

Employee Record not exist !!!!!

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

Index.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>
    <h3>welcome to index page</h3>
  <%
  session.setAttribute("user","Admin");
  %>
  <%
  Cookie ck=new Cookie("name","mycookie");
  response.addCookie(ck);
  %>
  <form action="ExpressionLanguage.jsp">
  Enter Name:<input type="text" name="name" /><br/><br/>
  <input type="submit" value="Submit"/>
  </form>
  </body>
```

```
</html>
```

ExpressionLanguage.jsp:-

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
   </head>
   <body>
      Welcome, ${ param.name }
      Session Value is ${ sessionScope.user }
      Cookie name is  , ${cookie.name.value}
   </body>
</html>
```

ELArithemeticOperator.jsp:-

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
   </head>
   <body>
            <%-- arithmetic op --%>
      5*5+4: ${5*5+4} <br>
      1.4E4+1.4: ${1.4E4+1.4}<br>
      10 mod 4: ${10 mod 4}<br>
      15 div 3: ${15 div 3}<br>
   </body>
</html>
```

ELLogicalOperator.jsp:-

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
   </head>
   <body>
            <%-- LogicalOperator --%>
      <h2>Logical Operator</h2>
      true and true: ${true and true}<br>
      true && false: ${true && false}<br>
      true or true: ${true or true}<br>
      true || false: ${true || false}<br>
      not true: ${not true}<br>
      !false: ${!false}
   </body>
</html>
```

ELRelationalOperator.jsp:-

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

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

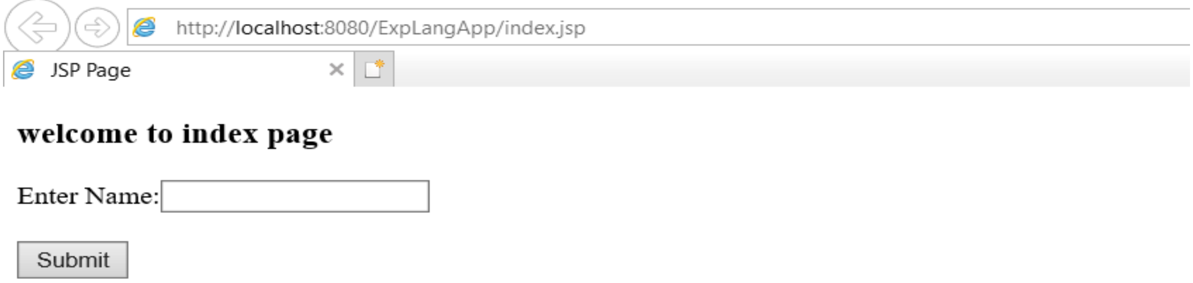
ELconditional.jsp:-

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    </head>
    <body>
        <h2>Conditional Operator</h2>
      The result of 10>2 is: "${(10>1)?'greater':'lesser'}"
    </body>
</html>
```

Empty.jsp:-

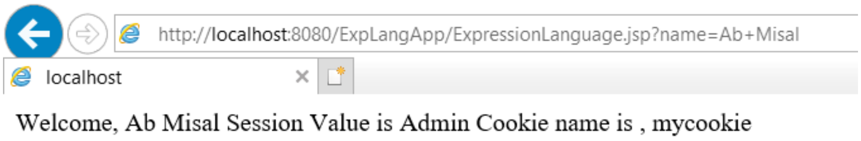
```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    </head>
    <body>
        <H1>Empty Operator Example</H1>
The Value for the Empty operator is:: ${empty "txxt"}
    </body>
</html>
```

Output:

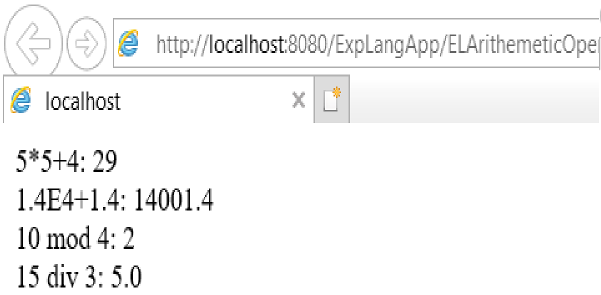


welcome to index page

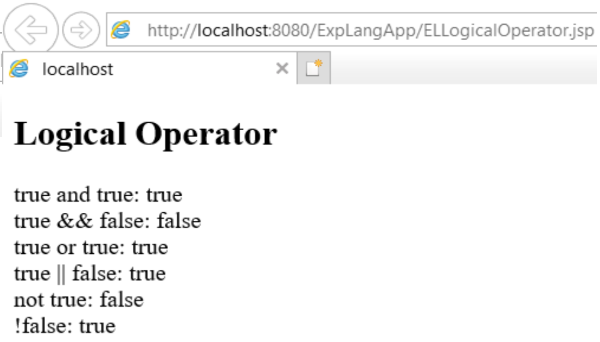
Enter Name:


http://localhost:8080/ExpLangApp/ExpressionLanguage.jsp?name=Ab+Misal
localhost

Welcome, Ab Misal Session Value is Admin Cookie name is , mycookie

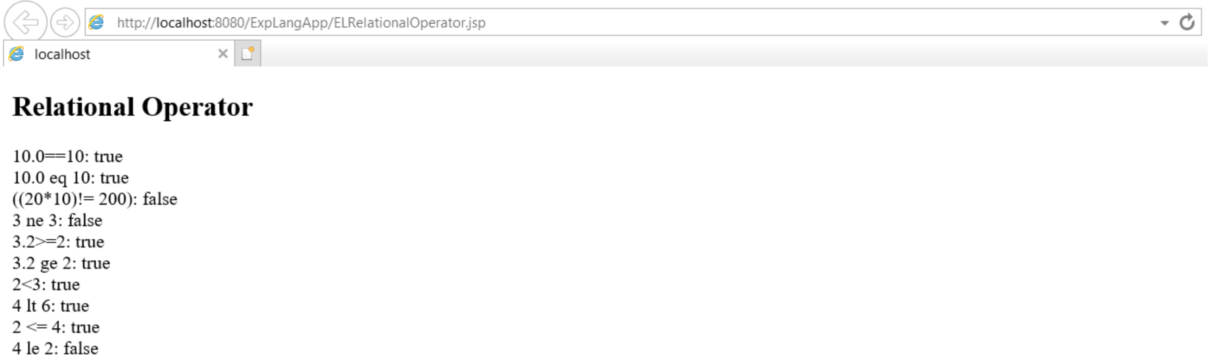


```
5*5+4: 29
1.4E4+1.4: 14001.4
10 mod 4: 2
15 div 3: 5.0
```



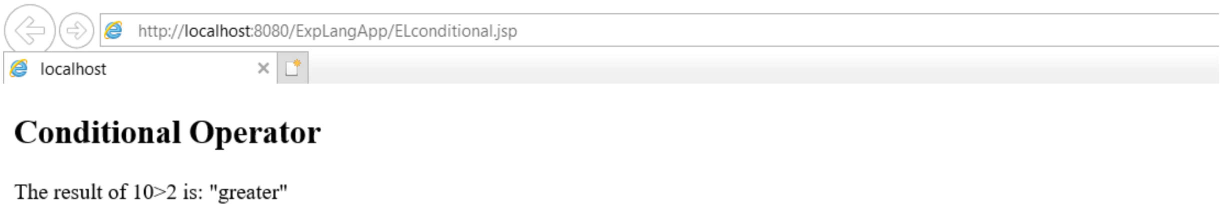
Logical Operator

```
true and true: true
true && false: false
true or true: true
true || false: true
not true: false
!false: true
```



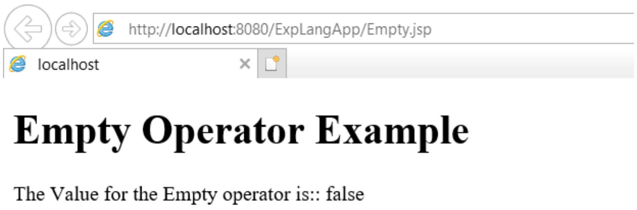
Relational Operator

```
10.0==10: true
10.0 eq 10: true
((20*10)!= 200): false
3 ne 3: false
3.2>=2: true
3.2 ge 2: true
2<3: true
4 lt 6: true
2 <= 4: true
4 le 2: false
```



Conditional Operator

The result of 10>2 is: "greater"



Empty Operator Example

The Value for the Empty operator is:: false

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

Index.html:-

```
<!DOCTYPE html>
<html>
    <head>
        <title>JSP Page</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <a href="SetDemo.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>
```

URlDemo.jsp:-

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

SetDemo.jsp:-

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="c" uri="http://java.sun.com/jstl/core"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <c:set var="pageTitle" scope="application"
               value="Dukes Soccer Premier League:Registartion"/>
            ${pageTitle}
    </body>
</html>
```

OutDemo.jsp:-

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

MaxIF.jsp:-

```
<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>
        <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"%>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <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>
```

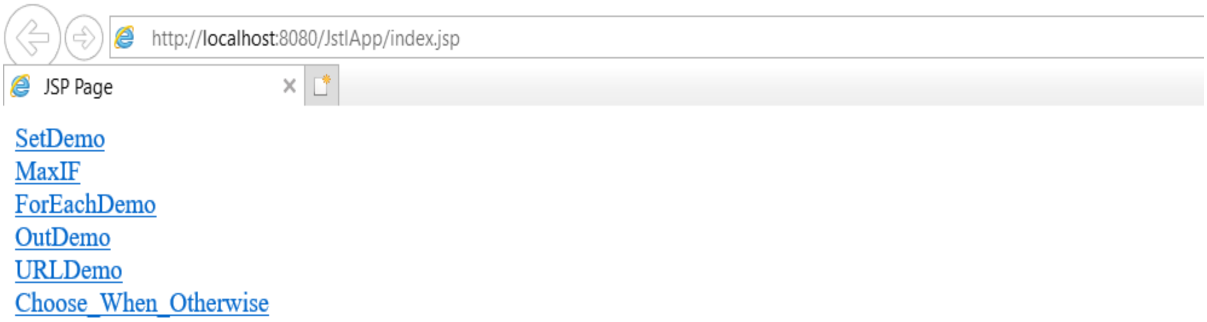
ForEachDemo.jsp:-

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <c:forEach begin="1" end="10" var="i">
            The Square of <c:out value="${i}=${i*i}"/><br>
        </c:forEach>
    </body>
</html>
```

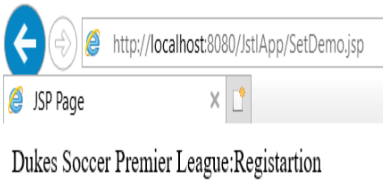
Choose_When_Otherwise.jsp:-

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
    </head>
    <body>
        <c:set var="income" value="${4000*4}"/>
        Your Income is: <c:out value="${income}"/>
        <c:choose>
            <c:when test="${income<=1000}">
                Income is not good
            </c:when>
            <c:when test="${income > 10000}">
                Income is very good
            </c:when>
            <c:otherwise>
                Income is undetermined
                </c:otherwise>
        </c:choose>
    </body>
</html>
```

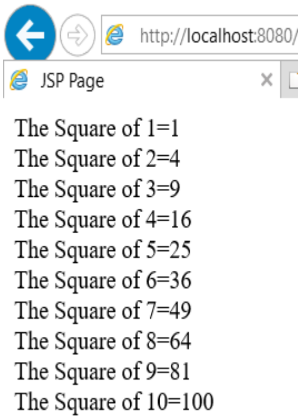
Output:



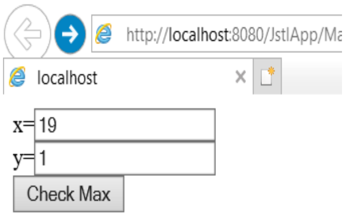
SetDemo
MaxIF
ForEachDemo
OutDemo
URLDemo
Choose_When_Otherwise



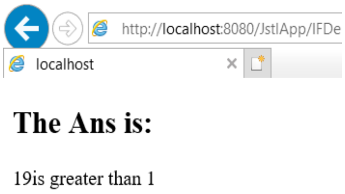
Dukes Soccer Premier League:Registartion



```
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
```

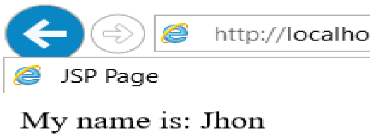


x= 19
y= 1

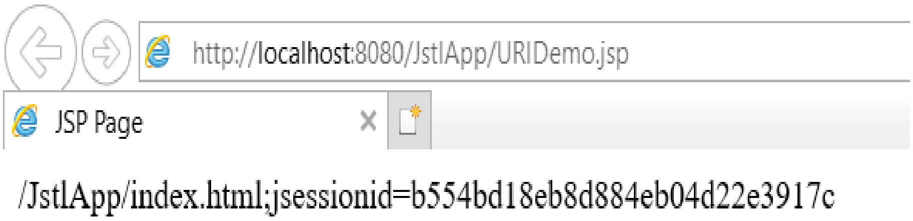


The Ans is:

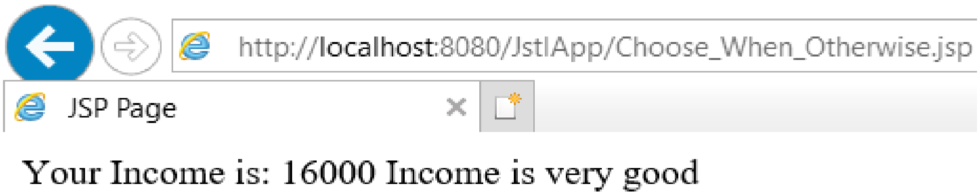
19is greater than 1



My name is: Jhon



/JstlApp/index.html;jsessionid=b554bd18eb8d884eb04d22e3917c



Your Income is: 16000 Income is very good

Practical no:-6

Implement the following EJB Applications.

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

Step 1 : Create a session bean named as *CCBean* in the package named *mybeans*. Select the option Stateless and click on Local Interface. Here you will find two files created in the mybeans package named as *CCBean.java and CCBeanLocal.java*

Step 2: Create a Servlet file name CCServlet in the package mypack.

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<br>
         <input type="Radio" name="type" value="r2d" checked>Rupee to Dollar
         <input type="Radio" name="type" value="d2r">Dollar to Rupee<br>
         <input type="Reset"><br><input type="submit" value="Convert">
      </form>
   </body>
</html>
```

CCBean.java:-

```
package mybeans;
import javax.ejb.Stateless;
@Stateless
public class CCBean implements CCBeanLocal
{
   @Override
   public double r2Dollar(double r)
   {
      return r/65.65;
   }
   @Override
   public double d2Rupee(double d)
   {
      return d*65.65;
   }
}
```

CCBeanLocal.java:-

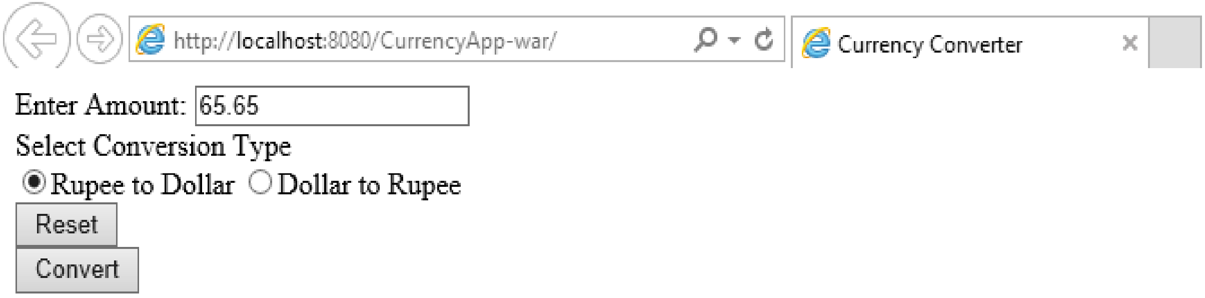
```
package mybeans;
import javax.ejb.Local;
@Local
public interface CCBeanLocal
{
   public double r2Dollar(double r);
   public double d2Rupee(double d);
}
```

CCServlet.java:-

```
package mypack;
```

```
import java.io.*;
import javax.ejb.EJB;
import javax.servlet.*;
import javax.servlet.http.*;
import mybeans.CCBeanLocal;
public class CCServlet extends HttpServlet
{
   @EJB CCBeanLocal obj;
   @Override
   public void doGet(HttpServletRequest req,HttpServletResponse res)throws
ServletException,IOException
   {
      res.setContentType("text/html");
      PrintWriter out=res.getWriter();
      double amt= Double.parseDouble(req.getParameter("amt"));
      if(req.getParameter("type").equals("r2d"))
      {
         out.println("<h1>"+amt+" Rupee= "+obj.r2Dollar(amt)+" Dollar</h1>");
      }
      if(req.getParameter("type").equals("d2r"))
      {
         out.println("<h1>"+amt+" Dollar= "+obj.d2Rupee(amt)+" Rupee</h1>");
      }
   }
}
```

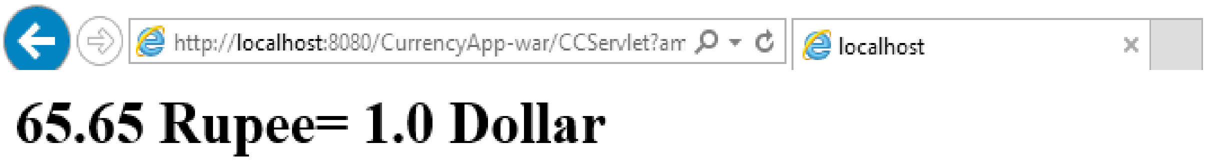
Output:



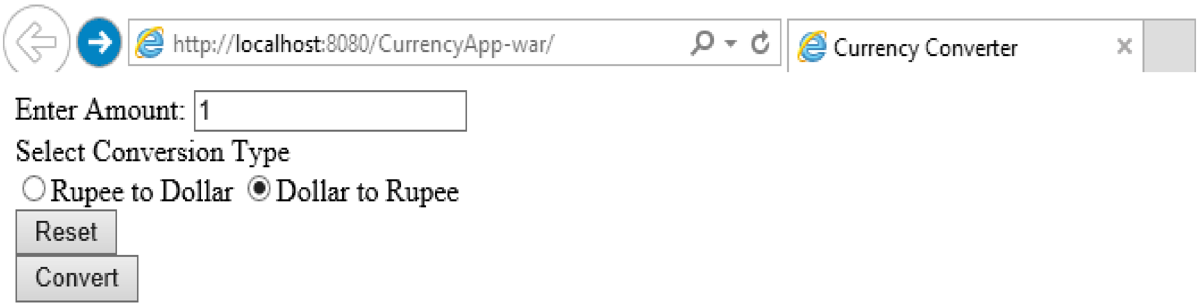
Enter Amount: 65.65

Select Conversion Type

● Rupee to Dollar ○ Dollar to Rupee



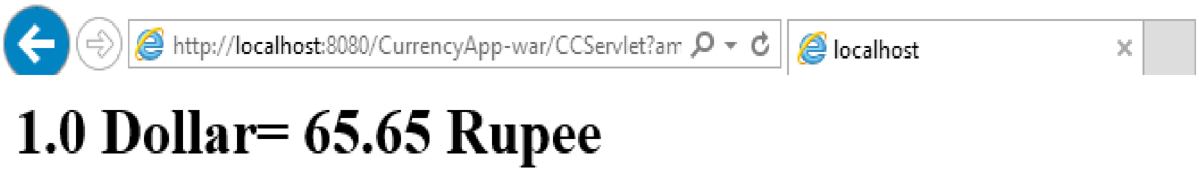
65.65 Rupee= 1.0 Dollar



Enter Amount: 1

Select Conversion Type

○ Rupee to Dollar ● Dollar to Rupee



1.0 Dollar= 65.65 Rupee

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

Step 1: Create a session bean named as *__RoomBean__* in the package named *__ejb__*. Select the option Stateless and click on Local Interface. Here you will find two files created in the ejb package named as *__RoomBean.java and RoomBeanLocal.java__*

Step 2: Create a Servlet file named as RoomClient. Do not click on web.xml (Deployment Descriptor)

__Index.html:-__

```
<!DOCTYPE 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 action="RoomClient" method="post">
         <br> No 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>
```

__RoomBean.java:-__

```
package ejb;
import java.sql.*;
import javax.ejb.Stateless;
@Stateless
public class RoomBean implements RoomBeanLocal
{
   @Override
   public int checkin(int no)
   {
      try
      {
         Class.forName("com.mysql.jdbc.Driver");
         Connection con =
DriverManager.getConnection("jdbc:mysql://localhost/roomdb","root","tiger");
         String sql1="select * from room";
         Statement st=con.createStatement();
         ResultSet rs=st.executeQuery(sql1);
         rs.next();
         int total=rs.getInt(1);
         int occ=rs.getInt(2);
         int free=total-occ;
         System.out.println(total);
         System.out.println(free);
         if(free>=no)
         {
            String sql2="update room set occ=?";
            PreparedStatement ps=con.prepareStatement(sql2);
            ps.setInt(1, occ+no);
```

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

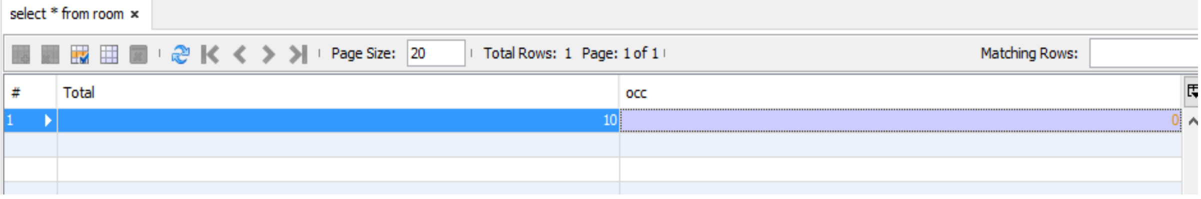
RoomBeanLocal.java:-

```
package ejb;
import javax.ejb.Local;
@Local
public interface RoomBeanLocal
{
    public int checkin(int no);
    public int checkout(int no);
}
```

RoomClient.java:-

```
package mypack;
import ejb.RoomBeanLocal;
import java.io.*;
import javax.ejb.EJB;
import javax.servlet.*;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;
@WebServlet(name = "RoomClient", urlPatterns = {"/RoomClient"})
public class RoomClient extends HttpServlet
{
   @EJB RoomBeanLocal obj;
   @Override
   protected void doPost(HttpServletRequest req,HttpServletResponse res)throws
ServletException,IOException
   {
      res.setContentType("text/html");
      PrintWriter out=res.getWriter();
      try
      {
         int no=Integer.parseInt(req.getParameter("t1"));
         String b=req.getParameter("btn");
         int rse=0;
         if(b.equals("CheckIN"))
         {
            rse=obj.checkin(no);
            if(rse==1)
               out.println(no+" rooms check-in");
         }
         if(b.equals("CheckOUT"))
         {
            rse=obj.checkout(no);
            if(rse==1)
               out.println(no+"rooms check-out");
         }
         if(rse==0)
         {
            out.println("not Possible to do Check In/Out");
            out.println("<br><br><a href=index.html> Back </a>");
         }
      }
      finally
      {
         out.close();
      }
   }
}
```

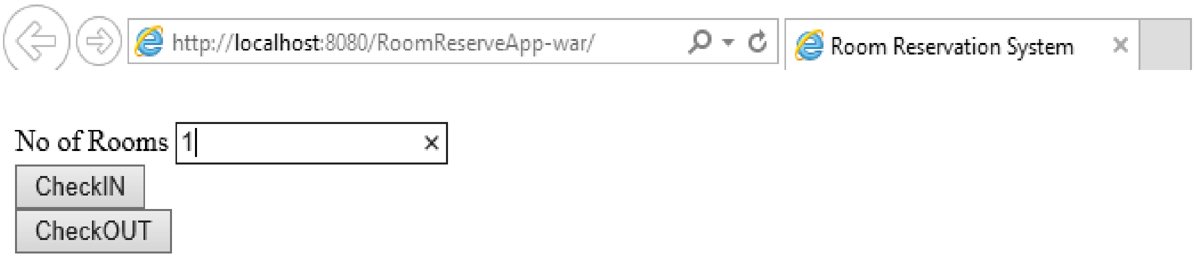
Output:



select * from room

Page Size: 20 Total Rows: 1 Page: 1 of 1 Matching Rows:

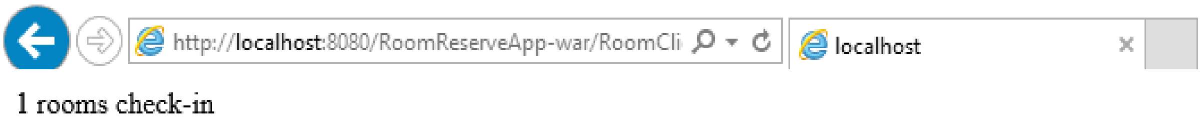
#	Total	occ
1	10	0



http://localhost:8080/RoomReserveApp-war/

Room Reservation System

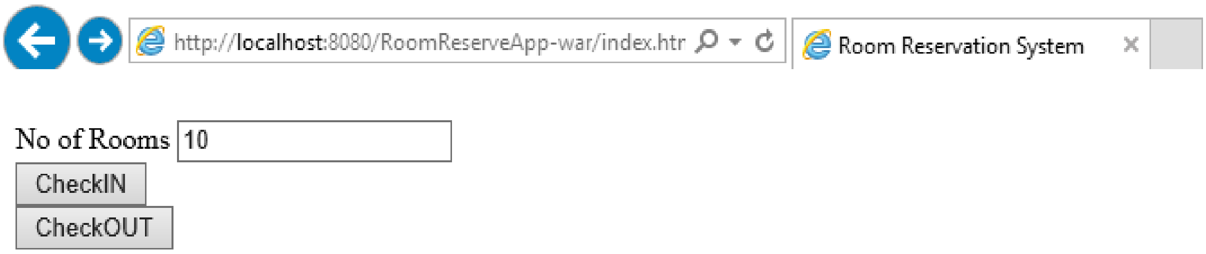
No of Rooms 1



http://localhost:8080/RoomReserveApp-war/RoomCli

localhost

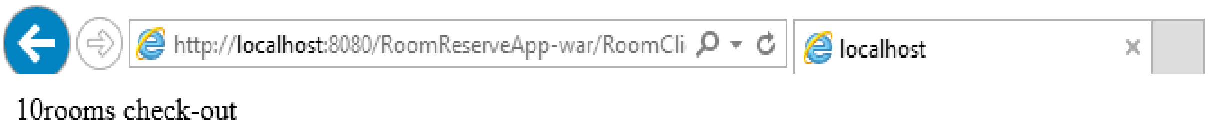
1 rooms check-in



http://localhost:8080/RoomReserveApp-war/index.htr

Room Reservation System

No of Rooms 10



http://localhost:8080/RoomReserveApp-war/RoomCli

localhost

10rooms check-out

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

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.

Step 3: creating a web client using index.jsp Right click on wewb pages -> new -> JSP -> filename -> index -> finish.

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.

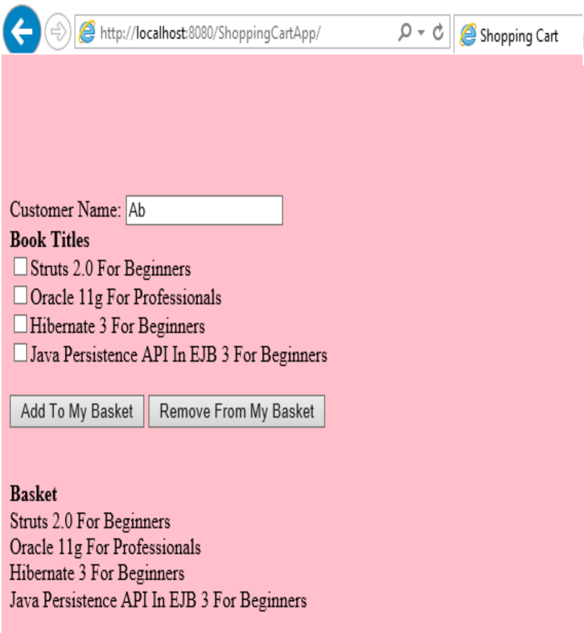
Index.jsp:-

```
<%@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/ShoppingCartApp/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]);
            }
        }
    }
%>
    <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>
```

<u>ShoppingCart.java:-</u>

```
package ejb;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;
import javax.ejb.Remove;
import javax.ejb.Stateful;
@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;
    }
}
```

Output:



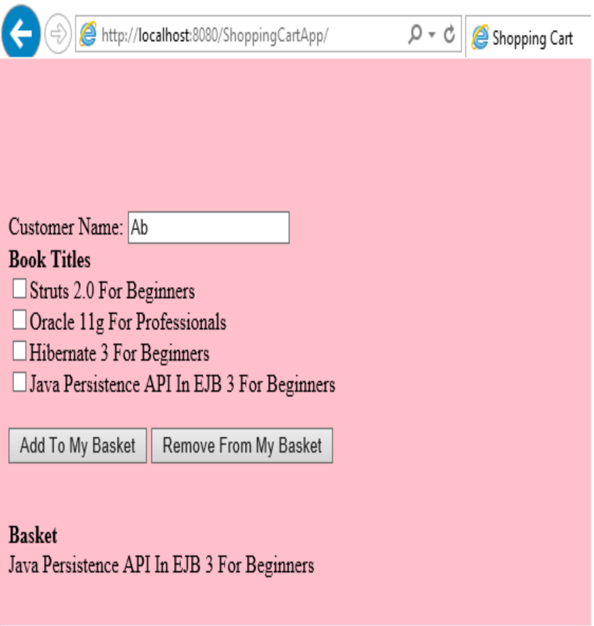
Customer Name: Ab

Book Titles

□ Struts 2.0 For Beginners
□ Oracle 11g For Professionals
□ Hibernate 3 For Beginners
□ Java Persistence API In EJB 3 For Beginners

Basket

Struts 2.0 For Beginners
Oracle 11g For Professionals
Hibernate 3 For Beginners
Java Persistence API In EJB 3 For Beginners



Customer Name: Ab

Book Titles

□ Struts 2.0 For Beginners
□ Oracle 11g For Professionals
□ Hibernate 3 For Beginners
□ Java Persistence API In EJB 3 For Beginners

Basket

Java Persistence API In EJB 3 For Beginners

Practical no:-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.

Step 1: Java Web-> web application -> Pract7AServletHitsSingltonApp -> finish.

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

Step 3: Create a Servlet File name ServletClient in the package name as servlet.

Do not select the Deployment Discriptor file

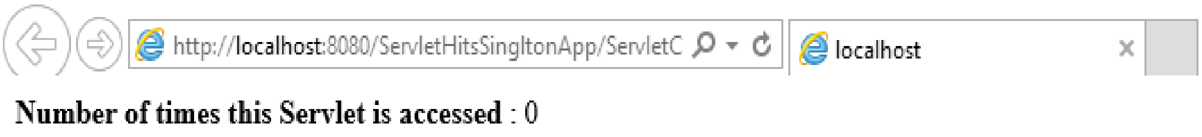
<u>ServletClient:-</u>

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

<u>CountServletHitsBean.java:-</u>

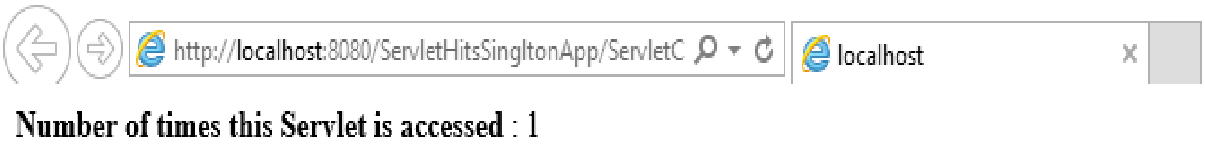
```
package ejb;
import javax.ejb.Singleton;
@Singleton
public class CountServletHitsBean
{
   private int hitCount;
   public synchronized int getCount()
   {
         return hitCount++;
   }
}
```

<u>Output:</u>



http://localhost:8080/ServletHitsSingltonApp/ServletC

Number of times this Servlet is accessed : 0



http://localhost:8080/ServletHitsSingltonApp/ServletC

Number of times this Servlet is accessed : 1

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

Step 1: Web-> web application -> Pract7BVisitorStatisticsMDBApp -> select dedicated folders for storing libraries -> finish.

Step2: Create a Database name visitorstat → Create table name → userstat → column names

Firstvisitdt – timestamp

Hostname – varchar 30 Primary Key

Visits – int

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

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

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.

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/QueryFactory");
        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>
```

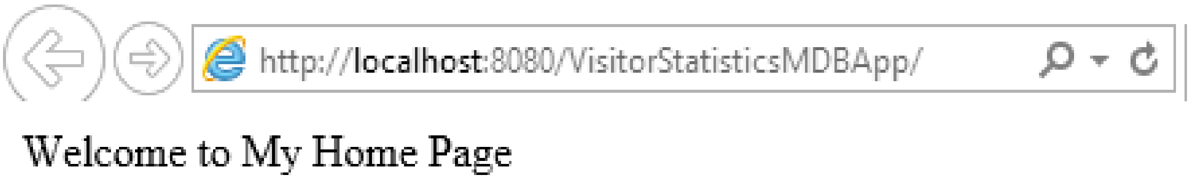
BasicMessageBean:-

```
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();
      }
    }
}
```

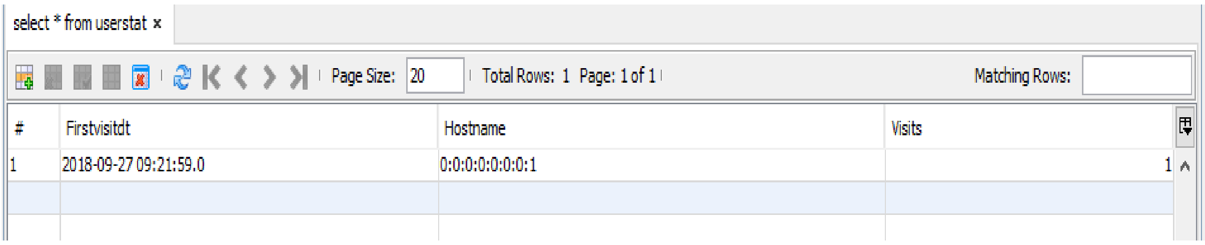
VisitorStatBean:-

```
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());
       }
   }
}
}
```

Output:



Welcome to My Home Page



#	Firstvisitdt	Hostname	Visits
1	2018-09-27 09:21:59.0	0:0:0:0:0:0:0:1	1

Q.7 c) Develop simple Marks Entry Application to demonstrate accessing Database using EJB.
Step 1: Create web application as pract7CMarksApp.
Step 2: Create database marksdb
Step 3: Create tables marks in marksdb database as:
create table marks (id int primary key auto_increment, sname varchar(35), marks1 int, marks2 int, marks3 int);
Step 4: create stateful java bean as
select source package -> session bean -> class name -> MarksEntryBean -> package -> ejb -> bean type-> stateful -> don’t select Local / Remote -> finish.

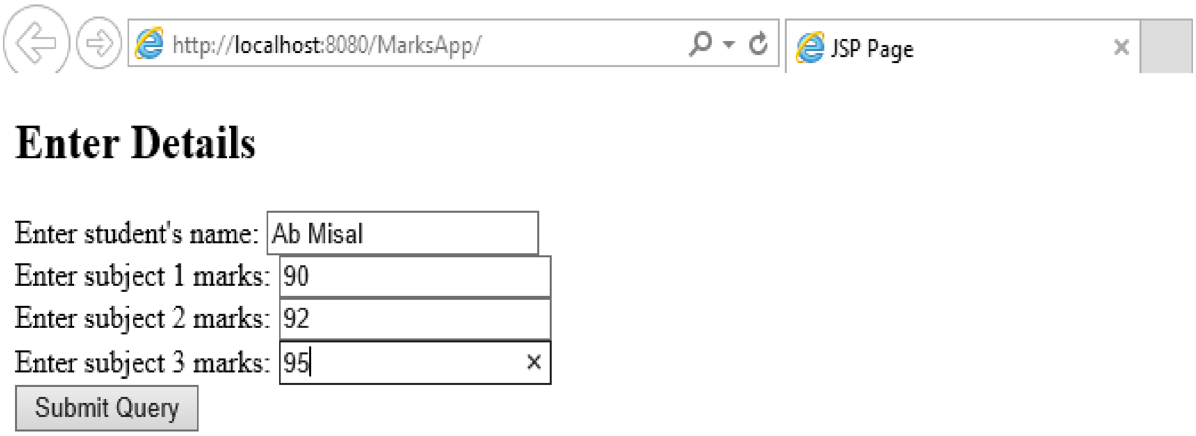
Index.jsp:-

```
<%@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/MarksApp/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>
```

MarksEntryBean:-

```
package ejb;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
import java.sql.ResultSet;
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:



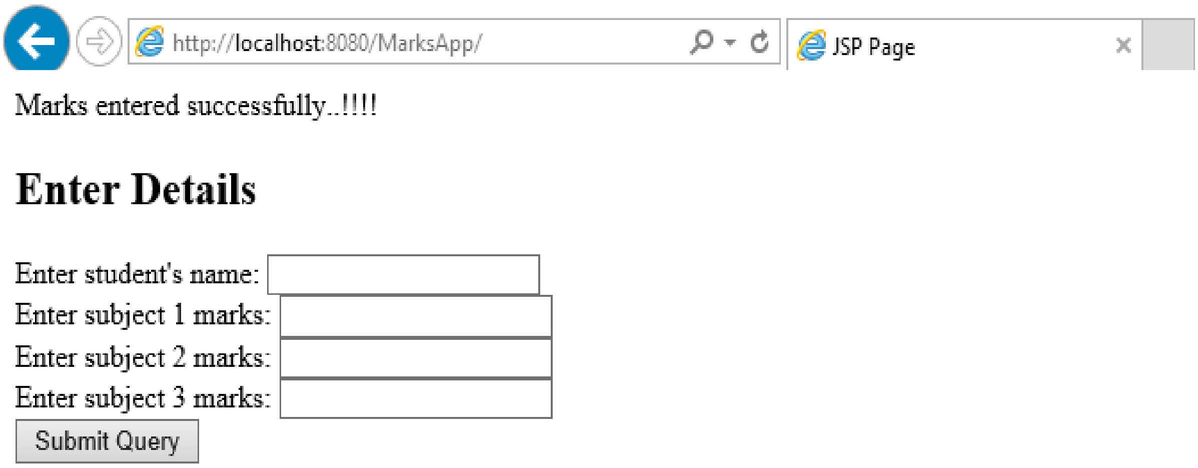
Enter Details

Enter student's name: Ab Misal

Enter subject 1 marks: 90

Enter subject 2 marks: 92

Enter subject 3 marks: 95



Marks entered successfully..!!!!

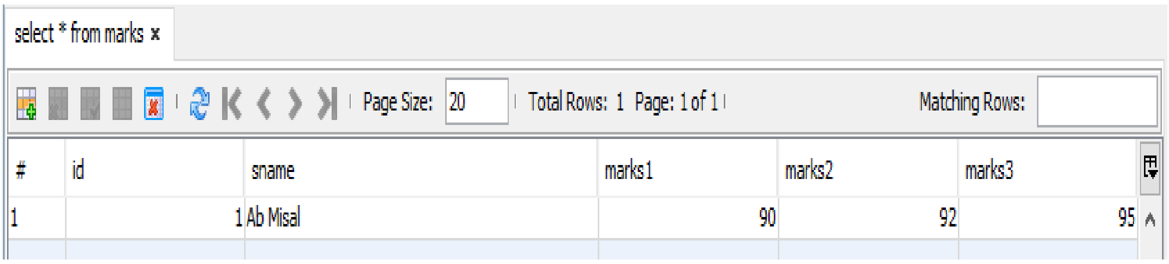
Enter Details

Enter student's name:

Enter subject 1 marks:

Enter subject 2 marks:

Enter subject 3 marks:


select * from marks

#	id	sname	marks1	marks2	marks3
1	1	Ab Misal	90	92	95

Practical no:-9

Implement the following JPA applications with ORM and Hibernate.

Q.9 a) Develop a JPA Application to demonstrate use of ORM associations.

Same steps to be followed as 9b

Index.jsp:-

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>User View</title>
    </head>
    <body>
        <form action="userview.jsp" >
            User Name:- <input type="text" name="uname" maxlength="20"><br>
            User Type:- <input type="text" name="utype" maxlength="35">
            <br><input type="submit" value="submit">
        </form>
    </body>
</html>
```

Userview.jsp:-

```
<%@page import="hibernet.User"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@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"%>
<%!
  SessionFactory sf;
  org.hibernate.Session ss;
  List<hibernet.User> ubook;
%>
<%
    sf = new Configuration().configure().buildSessionFactory();
    ss= sf.openSession();
    Transaction tx=null;
    User us= new User();
    try
    {
        tx=ss.beginTransaction();
        String uname=request.getParameter("uname");
        String utype=request.getParameter("utype");
        us.setUname(uname);
        us.setUtype(utype);
        ss.save(us);
        tx.commit();
    }
    catch(Exception e){ out.println("Error"+e.getMessage()); }
    try
    {
        ss.beginTransaction();
        ubook=ss.createQuery("from User").list();
```

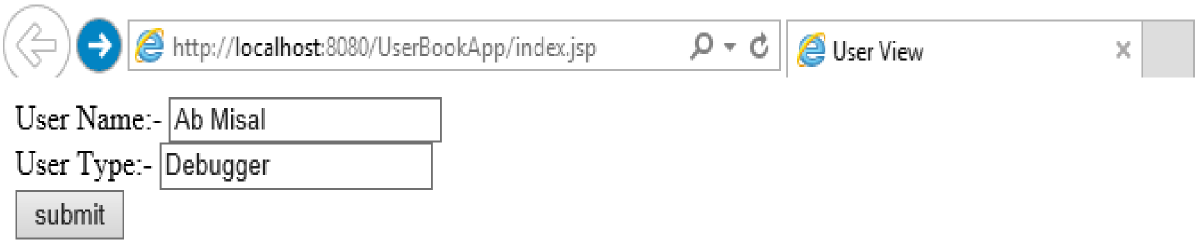
```
   }
   catch(Exception e){ }
%>
<html>
<head>
<title>User View</title>
</head>
<body>
      User View
      Click here to go <a href="index.jsp"> BACK </a>
<br><br>
<%
        Iterator it=ubook.iterator();
        while(it.hasNext())
        {
          User eachrecord=(User)it.next();
          out.print(eachrecord.getUid()+"   ");
          out.print(eachrecord.getUname()+"<br>");
          out.print(eachrecord.getUtype()+"<br><hr>");
        }
      %>
   </body>
</html>
```

User.java:-

```
package hibernet;
// Generated 4 Oct, 2018 8:02:52 AM by Hibernate Tools 4.3.1
/**
 * User generated by hbm2java
 */
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:

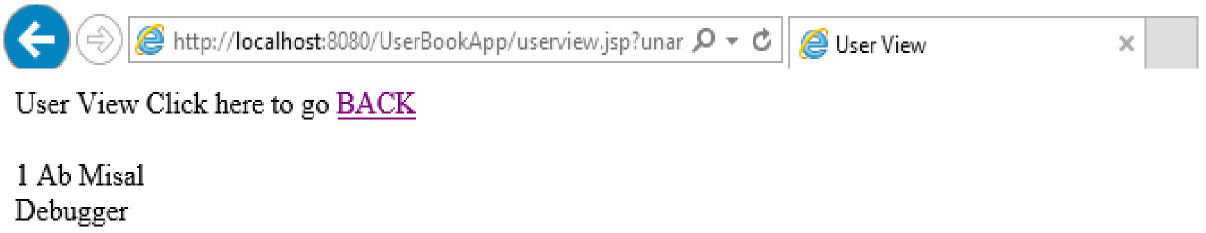


http://localhost:8080/UserBookApp/index.jsp

User View

User Name:- Ab Misal

User Type:- Debugger

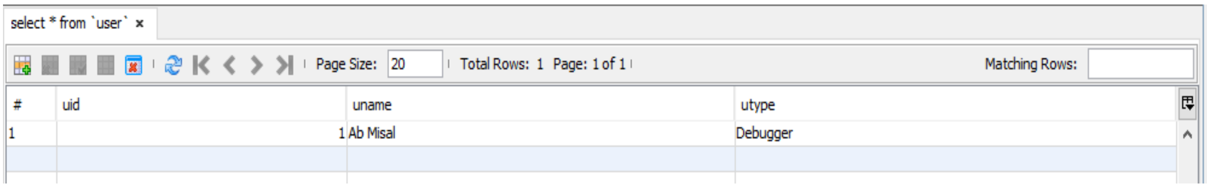


http://localhost:8080/UserBookApp/userview.jsp?unar

User View

User View Click here to go BACK

1 Ab Misal
Debugger



select * from `user`

#	uid	uname	utype
1	1	Ab Misal	Debugger

Q.9 b) Develop a Hibernate application to store Feedback of Website Visitor in MySQL Database.

Step 1: Create MySql Database

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.

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:-

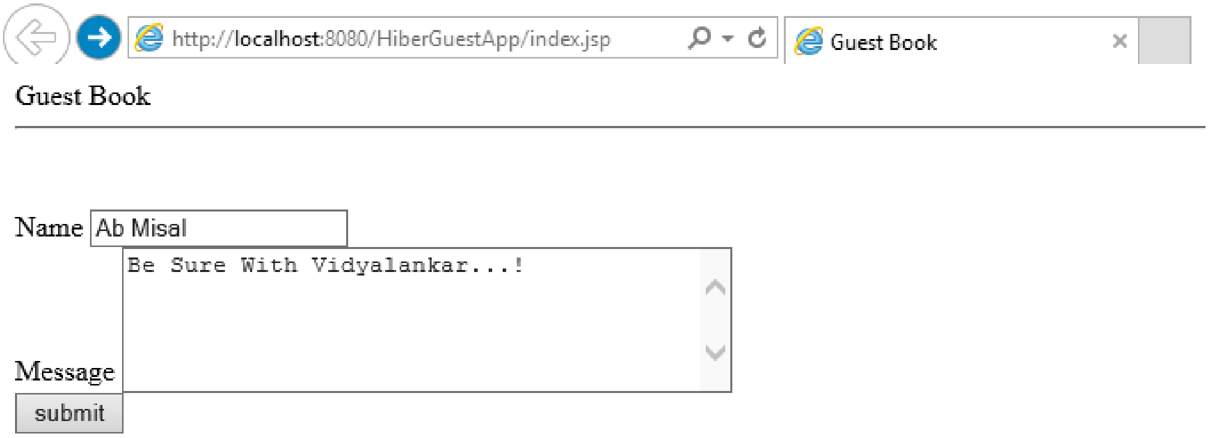
```
<%@page import="org.hibernate.SessionFactory"%>
<%@page import="org.hibernate.Session"%>
<%@page import="org.hibernate.cfg.Configuration"%>
<%@page import="org.hibernate.Transaction"%>
<%@page import="java.util.List"%>
<%@page import="java.util.Iterator"%>
<%@page import="hibernate.Guestbook"%>
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

Guestbook.java:-

```
package hibernate;
// Generated 4 Oct, 2018 9:40:17 AM by Hibernate Tools 4.3.1
/**
 * Guestbook generated by hbm2java
 */
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;
    }
}
```

Output:



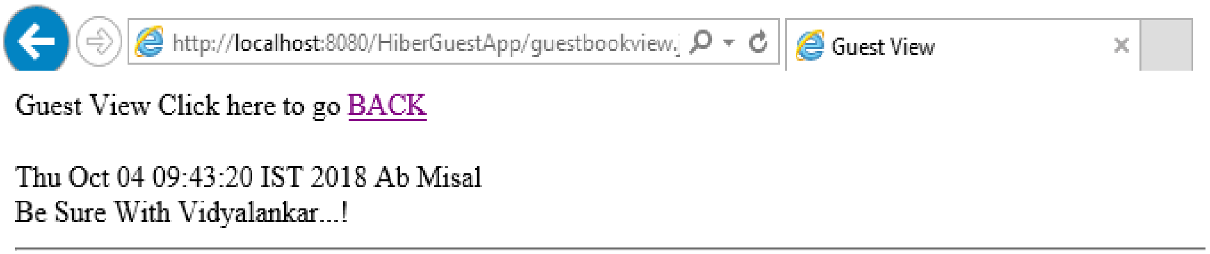
http://localhost:8080/HiberGuestApp/index.jsp

Guest Book

Guest Book

Name Ab Misal

Message Be Sure With Vidyalankar...!

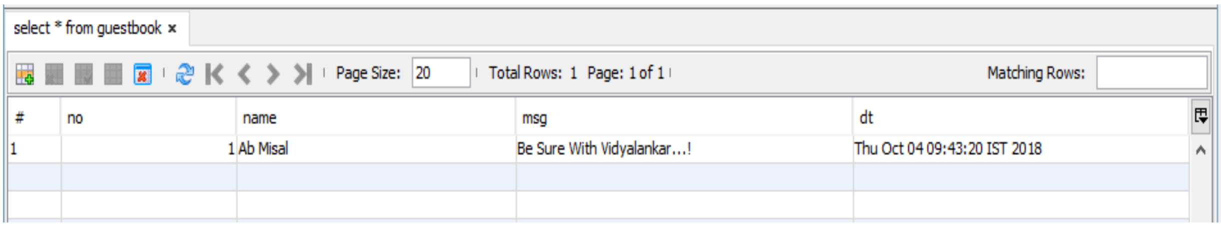


http://localhost:8080/HiberGuestApp/guestbookview.

Guest View

Guest View Click here to go BACK

Thu Oct 04 09:43:20 IST 2018 Ab Misal
Be Sure With Vidyalankar...!



select * from guestbook ×

Page Size: 20 Total Rows: 1 Page: 1 of 1 Matching Rows:

#	no	name	msg	dt
1	1	Ab Misal	Be Sure With Vidyalankar...!	Thu Oct 04 09:43:20 IST 2018

Q.9 c) Develop a Hibernate application to store and retrieve employee details in MySQL Database.

Same steps to be followed as 9b

Index.jsp:-

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Emp App</title>
    </head>
    <body>
        <form action="empview.jsp" >
            Emp Name:- <input type="text" name="ename" maxlength="20"><br>
            Emp Salary:- <input type="text" name="esal" maxlength="10"><br>
            Emp Designation:- <input type="text" name="edesag" maxlength="100"><br>
            <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"%>
```

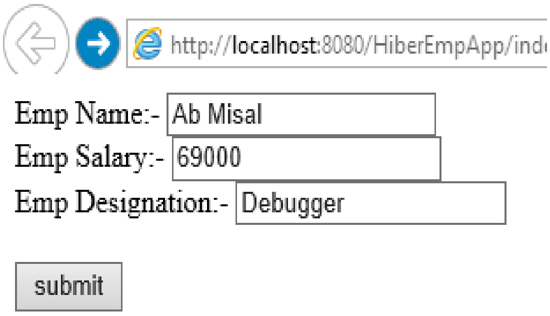
```
<%@page import="hibernate.Emp"%>
<%!
 SessionFactory sf;
 org.hibernate.Session ss;
 List<hibernate.Emp> ebook;
%>
<%
   sf = new Configuration().configure().buildSessionFactory();
   ss= sf.openSession();
   Transaction tx=null;
   Emp ep = new Emp();
   try{
      tx=ss.beginTransaction();
      String name=request.getParameter("ename");
      int sal=Integer.parseInt(request.getParameter("esal"));
      String disg=request.getParameter("edesag");
      ep.setEmpname(name);
      ep.setEmpsal(sal);
      ep.setEmpdisg(disg);
      ss.save(ep);
      tx.commit();
   }
   catch(Exception e){ out.println("Error"+e.getMessage()); }
   try{
      ss.beginTransaction();
      ebook=ss.createQuery("from Emp").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=ebook.iterator();
         while(it.hasNext()){
            Emp eachrecord=(Emp)it.next();
            out.print(eachrecord.getEmpno() +"   ");
            out.print(eachrecord.getEmpname() +"<br>");
            out.print(eachrecord.getEmpsal() +"<br>");
            out.print(eachrecord.getEmpdisg() +"<br><hr>");
         }
      %>
</body>
</html>
```

Emp.java:-

```
package hibernate;
// Generated 4 Oct, 2018 9:15:44 AM by Hibernate Tools 4.3.1
/**
 * Emp generated by hbm2java
 */
```

```
public class Emp  implements java.io.Serializable {
     private Integer empno;
     private String empname;
     private Integer empsal;
     private String empdisg;
    public Emp() {
    }
    public Emp(String empname, Integer empsal, String empdisg) {
       this.empname = empname;
       this.empsal = empsal;
       this.empdisg = empdisg;
    }
    public Integer getEmpno() {
        return this.empno;
    }
    public void setEmpno(Integer empno) {
        this.empno = empno;
    }
    public String getEmpname() {
        return this.empname;
    }
    public void setEmpname(String empname) {
        this.empname = empname;
    }
    public Integer getEmpsal() {
        return this.empsal;
    }
    public void setEmpsal(Integer empsal) {
        this.empsal = empsal;
    }
    public String getEmpdisg() {
        return this.empdisg;
    }
    public void setEmpdisg(String empdisg) {
        this.empdisg = empdisg;
    }
}
```

Output:

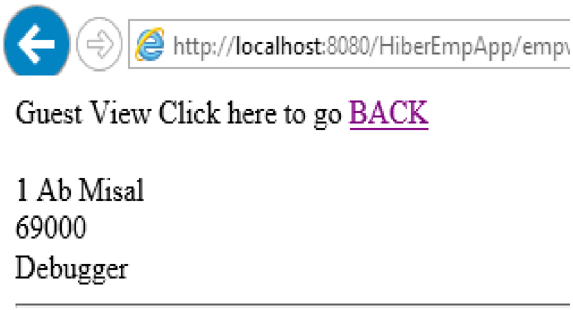


http://localhost:8080/HiberEmpApp/ind

Emp Name:- Ab Misal

Emp Salary:- 69000

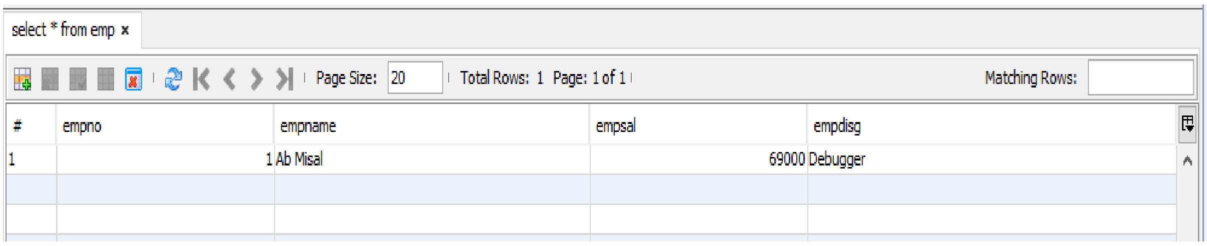
Emp Designation:- Debugger



http://localhost:8080/HiberEmpApp/emp

Guest View Click here to go BACK

1 Ab Misal
69000
Debugger



select * from emp ×

Page Size: 20 Total Rows: 1 Page: 1 of 1 Matching Rows:

#	empno	empname	empsal	empdisg
1	1	Ab Misal	69000	Debugger