Practical No.1

Aim: 1a. Create a simple calculator application using servlet.

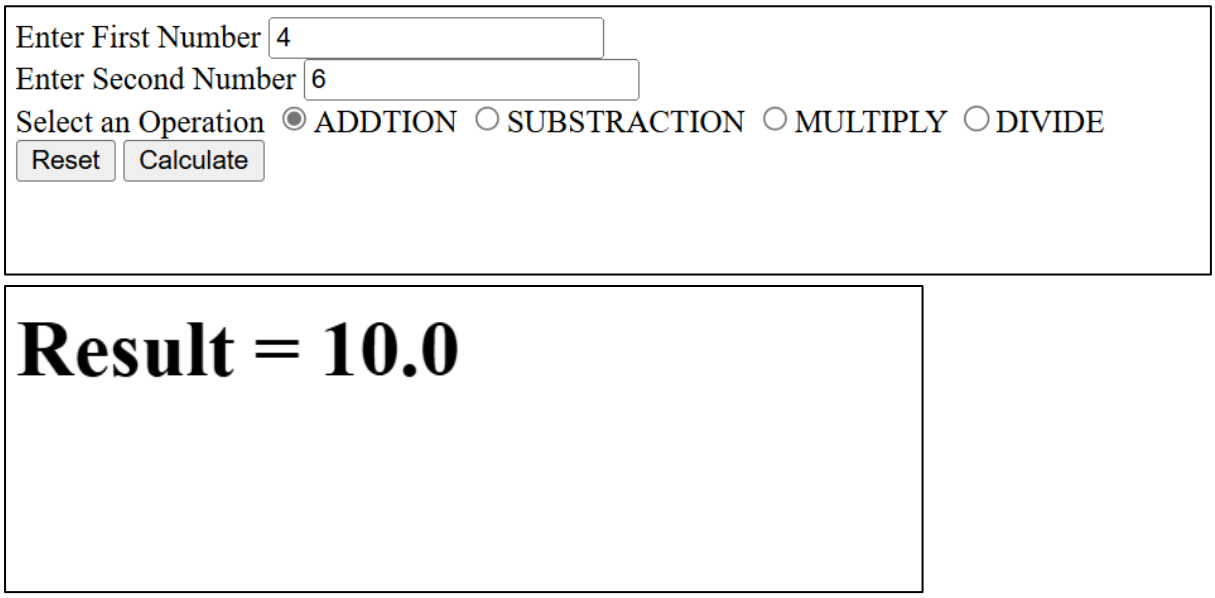
index.html

```
<html><head><title>Calculator App</title></head><body>
<form action="CalculatorServlet" >
Enter First Number <input type="text" name="txtN1" ><br>
Enter Second Number <input type="text" name="txtN2" ><br>
Select an operation
<input type="radio" name="opr" value="+">ADDTION
<input type="radio" name="opr" value="-">SUBSTRACTION
<input type="radio" name="opr" value="*">MULTIPLY
<input type="radio" name="opr" value="/">DIVIDE <br>
<input type="reset">
<input type="submit" value="Calculate" >
</form></body></html>
```

CalculatorServlet.java

```
package mypack;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class CalculatorServlet extends HttpServlet {
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
out.println("<html><head><title>Servlet CalculatorServlet</title></head><body>");
double n1 = Double.parseDouble(request.getParameter("txtN1"));
double n2 = Double.parseDouble(request.getParameter("txtN2"));
double result =0;
String opr=request.getParameter("opr");
if(opr.equals("+")) result=n1+n2; if(opr.equals("-")) result=n1-n2;
if(opr.equals("*")) result=n1*n2; if(opr.equals("/")) result=n1/n2;
out.println("<h1> Result = "+result); out.println("</body></html>");} }
```

Output:


Enter First Number 4
Enter Second Number 6
Select an Operation ADDTION SUBSTRACTION MULTIPLY DIVIDE
Reset
Calculate

Result = 10.0

Aim: 1b. 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”

Code:

index.html

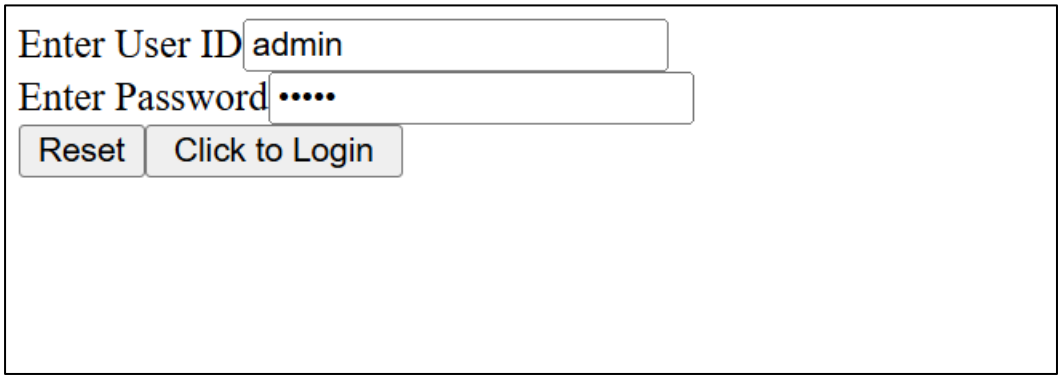
```
<html><head><title>Login Form</title></head>
<form action="LoginServlet" >
Enter User ID<input type="text" name="txtId"><br>
Enter Password<input type="password" name="txtPass"><br>
<input type="reset"><input type="submit" value=" Click to Login " ></form></html>
```

LoginServlet.java

```
package mypack;
import java.io.*;
import javax.servlet.ServletException;
import javax.servlet.http.*;
public class LoginServlet extends HttpServlet {
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
out.println("<html><head><title>Servlet LoginServlet</title></head>");
String uname = request.getParameter("txtId");
String upass = request.getParameter("txtPass");
if(uname.equals("admin") && upass.equals("12345")){
out.println("<body bgcolor=blue >");
out.println("<h1> Welcome !!! "+uname+"</h1>");
}
else{
out.println("<body bgcolor=red >");out.println("<h1> Login Fail !!! </h1>");
}
out.println("</body></html>");}}
```

Output:

Login page:



Login Success:

Login Fail:

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

Code:

MySql queries:

```
create database LoginDB;
use LoginDB;
create table user(username varchar(20) PRIMARY KEY, password varchar(20), email varchar(20),
country varchar(20));
insert into user values ('admin','admin','admin@admin.com','India');
select * from user;
```

index.html

```
<html><head><title>Registration Page</title></head>
<body>
<form action="RegisterServlet" >
<H1>Welcome to Registration page</H1>
Enter User Name <input type="text" name="txtUid"><br>
Enter Password <input type="password" name="txtPass"><br>
Enter Email <input type="text" name="txtEmail" ><br>
Enter Country <input type="text" name="txtCon" ><br>
<input type="reset" ><input type="submit" value="REGISTER" >
</form>
</body>
</html>
```

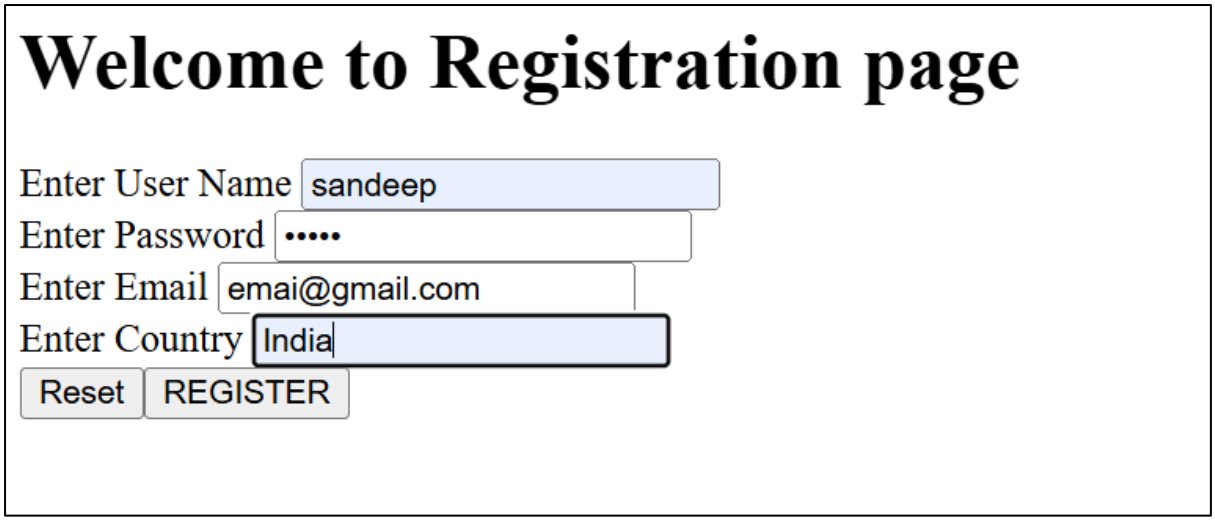
RegisterServlet.java

```
package mypack;
import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class RegisterServlet extends HttpServlet {
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
String id = request.getParameter("txtUid");
String ps = request.getParameter("txtPass");
String em = request.getParameter("txtEmail");
String co = request.getParameter("txtCon");
```

```
try{
Class.forName("com.mysql.jdbc.Driver");
Connection con
=DriverManager.getConnection("jdbc:mysql://localhost:3306/logindb");
PreparedStatement pst = con.prepareStatement("insert into user values(?,?,?,?)");
pst.setString(1,id);
pst.setString(2,ps);
pst.setString(3,em);
pst.setString(4,co);
int row = pst.executeUpdate();
out.println("<h1>"+row+ " Inserted Succesfullyyyyy");
}catch(Exception e){out.println(e);}
}
}
```

Output:

Registration page:



Date saved in Database:

Practical No.2

Aim: 2a. Using Request Dispatcher Interface create a Servlet which will validate the password entered by the user, if the user has entered "Servlet" as password, then he will be forwarded to Welcome Servlet else the user will stay on the index.html page and an error message will be displayed.

Code:

index.html

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

LoginServlet.java

```
package mypack;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.RequestDispatcher;
public class LoginServlet extends HttpServlet {
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
out.println("<html><head>");
out.println("<title>Servlet LoginServlet</title></head>");
String uname = request.getParameter("txtId");
String upass = request.getParameter("txtPass");
if(uname.equals("admin") && upass.equals("servlet")){
RequestDispatcher rd = request.getRequestDispatcher("WelcomeServlet");
rd.forward(request, response);
}
else{
out.println("<body bgcolor=red >");
out.println("<h1> Login Fail !!! </h1>");
RequestDispatcher rd = request.getRequestDispatcher("index.html");
```

```
rd.include(request, response);
}
out.println("</body>");
out.println("</html>");
}
```

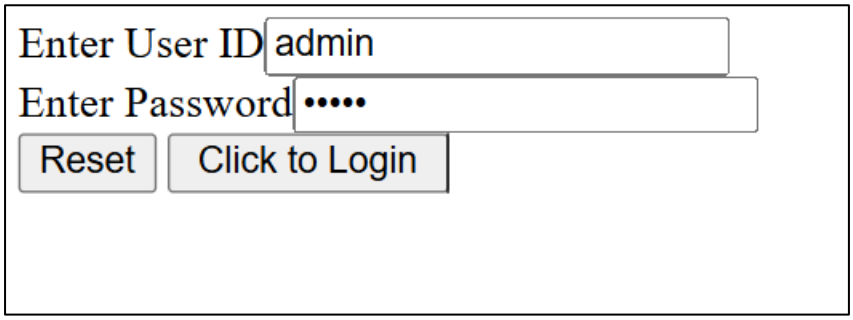
Welcomeservlet.java

```
package mypack;
import java.io.IOException;
import java.io.PrintWriter;
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 = "Welcomeservlet", urlPatterns = {"/Welcomeservlet"})
public class Welcomeservlet extends HttpServlet {
   protected void processRequest(HttpServletRequest request, HttpServletResponse
response)
         throws ServletException, IOException {
      response.setContentType("text/html;charset=UTF-8");
      try (PrintWriter out = response.getWriter()) {
         /* TODO output your page here. You may use following sample code. */
         out.println("<!DOCTYPE html>");
         out.println("<html>");
         out.println("<head>");
         out.println("<title>Servlet Welcomeservlet</title>");
         out.println("</head>");
         out.println("<body bgcolor=green>");
         out.println("<h1> Login Sucessfully</h1>");
         out.println("</body>");
         out.println("</html>");
      }
   }
   @Override
   protected void doGet(HttpServletRequest request, HttpServletResponse response)
         throws ServletException, IOException {
      processRequest(request, response);
   }
   @Override
   protected void doPost(HttpServletRequest request, HttpServletResponse response)
         throws ServletException, IOException {
      processRequest(request, response);
   }
```

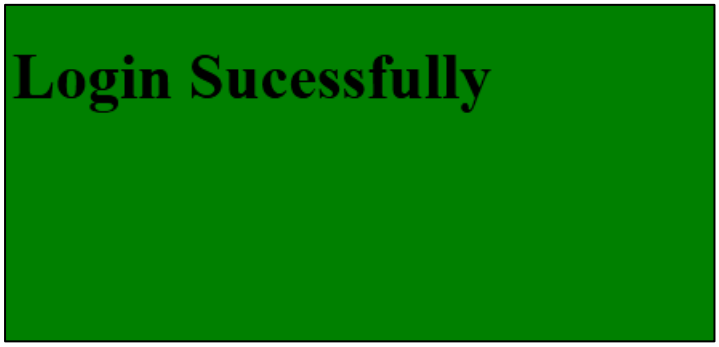
```
    @Override
    public String getServletInfo() {
        return "Short description";
    }
}
```

Output:

Login Page:



Login Successfully:



Login Fail:

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

index.html

```
<html>
<head><title>Cookie Demo</title></head>
<body>
<form action="Page1" >
Enter Your Name <input type="text" name="txtName"><br>
<input type="submit" value="~~~ Click to Enter ~~~">
</form>
</body>
</html>
```

Page1.java

```
package mypack;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.Cookie;
public class Page1 extends HttpServlet {
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
out.println("<html><head><title>Page1</title></head>");
out.println("<body bgcolor=pink >");
String uname = request.getParameter("txtName");
out.println("<h1>~~~ Welcome "+uname+"</h1>");
Cookie ck1 = new Cookie("username", uname);
Cookie ck2 = new Cookie("visit","1");
response.addCookie(ck1); response.addCookie(ck2);
out.println("<h1><a href=Page2 >Click to visit Page 2 </a></h1>");
out.println("</body>");
out.println("</html>");
}
}
```

Page2.java

```
package mypack;
```

```
import java.io.*;
import javax.servlet.ServletException;
import javax.servlet.http.*;
public class Page2 extends HttpServlet {
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
out.println("<html><head><title>Page2</title></head>");
out.println("<body bgcolor=yellow >");
Cookie [] ck = request.getCookies();
for(int i=0;i<ck.length;i++){
if(ck[i].getName().equals("visit")){
int count = Integer.parseInt(ck[i].getValue())+1;
out.println("<h1>Visit No : "+count+"</h1>");
ck[i] = new Cookie("visit",count+"");
response.addCookie(ck[i]);
}
else {
out.println(ck[i].getName()+ " = "+ck[i].getValue()); }
out.println("<h1><a href=Page3 >Click to visit Page 3 </a></h1>");
out.println("<h1><a href=Page4 >Click to visit Page 4 </a></h1>");
out.println("<h1><a href=Page5 >Click to visit Page 5 </a></h1>");
out.println("</body>");
out.println("</html>");
} }
```

Page3.java

```
package mypack;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.Cookie;
@WebServlet(urlPatterns = {"/Page3"})
public class Page3 extends HttpServlet {
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
out.println("<html><head><title>Page3</title></head>");
```

```
out.println("<body bgcolor=red >");
Cookie [] ck = request.getCookies();
for(int i=0;i<ck.length;i++){
if(ck[i].getName().equals("visit")){
int count = Integer.parseInt(ck[i].getValue())+1;
out.println("<h1>Visit No : "+count+"</h1>");
ck[i] = new Cookie("visit",count+"");
response.addCookie(ck[i]);
}
else {
out.println(ck[i].getName()+ " = "+ck[i].getValue()); }
out.println("<h1><a href=Page3 >Click to visit Page 3 </a></h1>");
out.println("<h1><a href=Page4 >Click to visit Page 4 </a></h1>");
out.println("<h1><a href=Page5 >Click to visit Page 5 </a></h1>");
out.println("</body>");
out.println("</html>");
} }}
```

Page4.java

```
package mypack;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.Cookie;
@WebServlet(urlPatterns = {"/Page4"})
public class Page4 extends HttpServlet {
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
out.println("<html><head><title>Page3</title></head>");
out.println("<body bgcolor=red >");
Cookie [] ck = request.getCookies();
for(int i=0;i<ck.length;i++){
if(ck[i].getName().equals("visit")){
int count = Integer.parseInt(ck[i].getValue())+1;
out.println("<h1>Visit No : "+count+"</h1>");
ck[i] = new Cookie("visit",count+"");
response.addCookie(ck[i]);
}
```

```
else {
out.println(ck[i].getName()+ " = "+ck[i].getValue()); }
out.println("<h1><a href=Page3 >Click to visit Page 3 </a></h1>");
out.println("<h1><a href=Page4 >Click to visit Page 4 </a></h1>");
out.println("<h1><a href=Page5 >Click to visit Page 5 </a></h1>");
out.println("</body>");
out.println("</html>");
} }}
```

Page5.java

```
package mypack;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.Cookie;
@WebServlet(urlPatterns = {"/Page5"})
public class Page5 extends HttpServlet {
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
out.println("<html><head><title>Pag5</title></head>");
out.println("<body bgcolor=white >");
Cookie [] ck = request.getCookies();
for(int i=0;i<ck.length;i++){
if(ck[i].getName().equals("visit")){
int count = Integer.parseInt(ck[i].getValue())+1;
out.println("<h1>Visit No : "+count+"</h1>");
ck[i] = new Cookie("visit",count+"");
response.addCookie(ck[i]);
}
else {
out.println(ck[i].getName()+ " = "+ck[i].getValue()); }
out.println("<h1><a href=Page3 >Click to visit Page 3 </a></h1>");
out.println("<h1><a href=Page4 >Click to visit Page 4 </a></h1>");
out.println("<h1><a href=Page5 >Click to visit Page 5 </a></h1>");
out.println("</body>");
out.println("</html>");
} }}
```

Output:


Enter Your Name Mr.Unknown
~~~ Click to Enter ~~~
~~~ Welcome Mr.Unknown
Click to visit Page 2
username = Mr.Unknown
Click to visit Page 3
Click to visit Page 4
Click to visit Page 5
Visit No : 2
Click to visit Page 3
Click to visit Page 4
Click to visit Page 5
username = Mr.Unknown
Click to visit Page 3
Click to visit Page 4
Click to visit Page 5
Visit No : 3
Click to visit Page 3
Click to visit Page 4
Click to visit Page 5

Aim: 2c. 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.

index.html

```
<html>
<head><title>Session Demo</title></head>
<form action="Page1" method="get" >
Enter User ID <input type="text" name="txtName"><br>
<input type="reset" ><input type="submit" >
</form>
</html>
```

Page1.java

```
package mypack;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
public class Page1 extends HttpServlet {
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
out.println("<html><head><title>Servlet Page1</title></head>");
HttpSession hs = request.getSession(true);
if(hs.isNew()){
out.println("<body bgcolor=yellow>");
String name = request.getParameter("txtName");
hs.setAttribute("uname", name);
hs.setAttribute("visit", "1");
out.println("<h1>Welcome First Time</h1>");
}
else{
out.println("<h1>Welcome Again</h1>");
int visit = Integer.parseInt((String)hs.getAttribute("visit"))+1;
out.println("<h1>You Visited "+visit+"Times</h1>");
hs.setAttribute("visit", ""+visit);
}
out.println("<h1>Your Session ID "+hs.getId()+"</h1>");
```

```
out.println("<h1>You Logged in at "+new java.util.Date(hs.getCreationTime())+"</h1>");
out.println("<h1><a href=Page2>Click for Page 2 </a></h1>");
out.println("<h1><a href=Page3>Click for Page 3 </a></h1>");
out.println("<h1><a href=Page4>Click for Page 4 </a></h1>");
out.println("<h1><a href=LogoutServlet>Click to Terminate Session </a></h1>");
out.println("</body>");
out.println("</html>");
}
}
```

Page2.java

```
package mypack;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
public class Page2 extends HttpServlet {
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
out.println("<html><head><title>Servlet Page2</title></head>");
HttpSession hs = request.getSession(false);
out.println("<h1>Welcome Again on Page No. 2</h1>");
int visit = Integer.parseInt((String)hs.getAttribute("visit"))+1;
out.println("<h1>You Visited "+visit+"Times</h1>");
hs.setAttribute("visit", ""+visit);
out.println("<h1>Your Session ID "+hs.getId()+"</h1>");
out.println("<h1>You Logged in at "+new java.util.Date(hs.getCreationTime())+"</h1>");
out.println("<h1><a href=Page1>Click for Page 1 </a></h1>");
out.println("<h1><a href=Page3>Click for Page 3 </a></h1>");
out.println("<h1><a href=Page4>Click for Page 4 </a></h1>");
out.println("<h1><a href=LogoutServlet>Click for Terminate Session </a></h1>");
out.println("</body>");
out.println("</html>");
}}
```

Page3.java

```
package mypack;
import java.io.IOException;
import java.io.PrintWriter;
```

```
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
@WebServlet(name = "Page3", urlPatterns = {"/Page3"})
public class Page3 extends HttpServlet {
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
out.println("<html><head><title>Servlet Page3</title></head>");
HttpSession hs = request.getSession(false);
out.println("<h1>Welcome Again on Page No. 3</h1>");
int visit = Integer.parseInt((String)hs.getAttribute("visit"))+1;
out.println("<h1>You Visited "+visit+"Times</h1>");
hs.setAttribute("visit", ""+visit);
out.println("<h1>Your Session ID "+hs.getId()+"</h1>");
out.println("<h1>You Logged in at "+new java.util.Date(hs.getCreationTime())+"</h1>");
out.println("<h1><a href=Page1>Click for Page 1 </a></h1>");
out.println("<h1><a href=Page3>Click for Page 3 </a></h1>");
out.println("<h1><a href=Page4>Click for Page 4 </a></h1>");
out.println("<h1><a href=LogoutServlet>Click for Terminate Session </a></h1>");
out.println("</body>");
out.println("</html>");
}
}
```

Page4.java

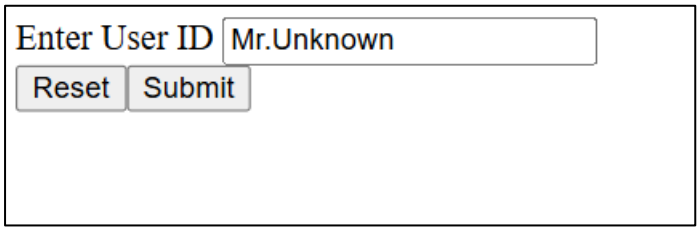
```
package mypack;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
@WebServlet(name = "Page4", urlPatterns = {"/Page4"})
public class Page4 extends HttpServlet {
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
```

```
PrintWriter out = response.getWriter();
out.println("<html><head><title>Servlet Page4</title></head>");
HttpSession hs = request.getSession(false);
out.println("<h1>Welcome Again on Page No. 4</h1>");
int visit = Integer.parseInt((String)hs.getAttribute("visit"))+1;
out.println("<h1>You Visited "+visit+"Times</h1>");
hs.setAttribute("visit", ""+visit);
out.println("<h1>Your Session ID "+hs.getId()+"</h1>");
out.println("<h1>You Logged in at "+new java.util.Date(hs.getCreationTime())+"</h1>");
out.println("<h1><a href=Page1>Click for Page 1 </a></h1>");
out.println("<h1><a href=Page3>Click for Page 3 </a></h1>");
out.println("<h1><a href=Page4>Click for Page 4 </a></h1>");
out.println("<h1><a href=LogoutServlet>Click for Terminate Session </a></h1>");
out.println("</body>");
out.println("</html>");
}
}
```

LogoutServlet.java

```
package mypack;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class LogoutServlet extends HttpServlet {
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
out.println("<html><head><title>Servlet LogoutServlet</title></head>");
out.println("<body>");
javax.servlet.http.HttpSession hs = request.getSession();
if(hs != null) hs.invalidate();
out.println("<h1>You are Logged out now.....</h1>");
out.println("</body>");
out.println("</html>");
}
}
```

Output:



Welcome First Time

Your Session ID bbb347014ea8c6eeedad4d1bcd92

You Logged in at Tue Oct 08 16:14:43 IST 2024

Click for Page 2

Click for Page 3

Click for Page 4

Click to Terminate Session

Welcome Again on Page No. 2

You Visited 2Times

Your Session ID bbb347014ea8c6eeedad4d1bcd92

You Logged in at Tue Oct 08 16:14:43 IST 2024

Click for Page 1

Click for Page 3

Click for Page 4

Click for Terminate Session

You are Logged out now.....

Practical No.3

Aim: 3a. Create a Servlet application to upload and download a file.

index.html

```
<html>
<body>
<form action="FileUploadServlet" method="post" enctype="multipart/form-data">
Select File to Upload:<input type="file" name="file" id="file">
Destination <input type="text" value="/tmp" name="destination">
<br>
<input type="submit" value="Upload file" name="upload" id="upload">
</form>
</body>
</html>
```

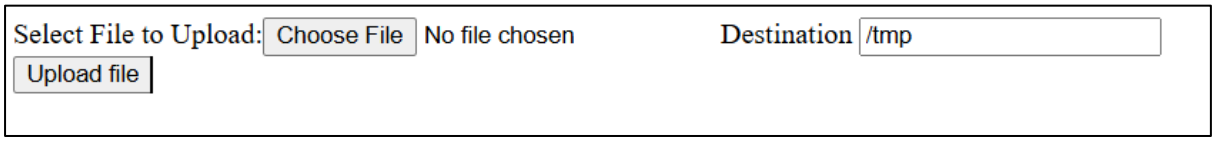
DownloadServlet.java

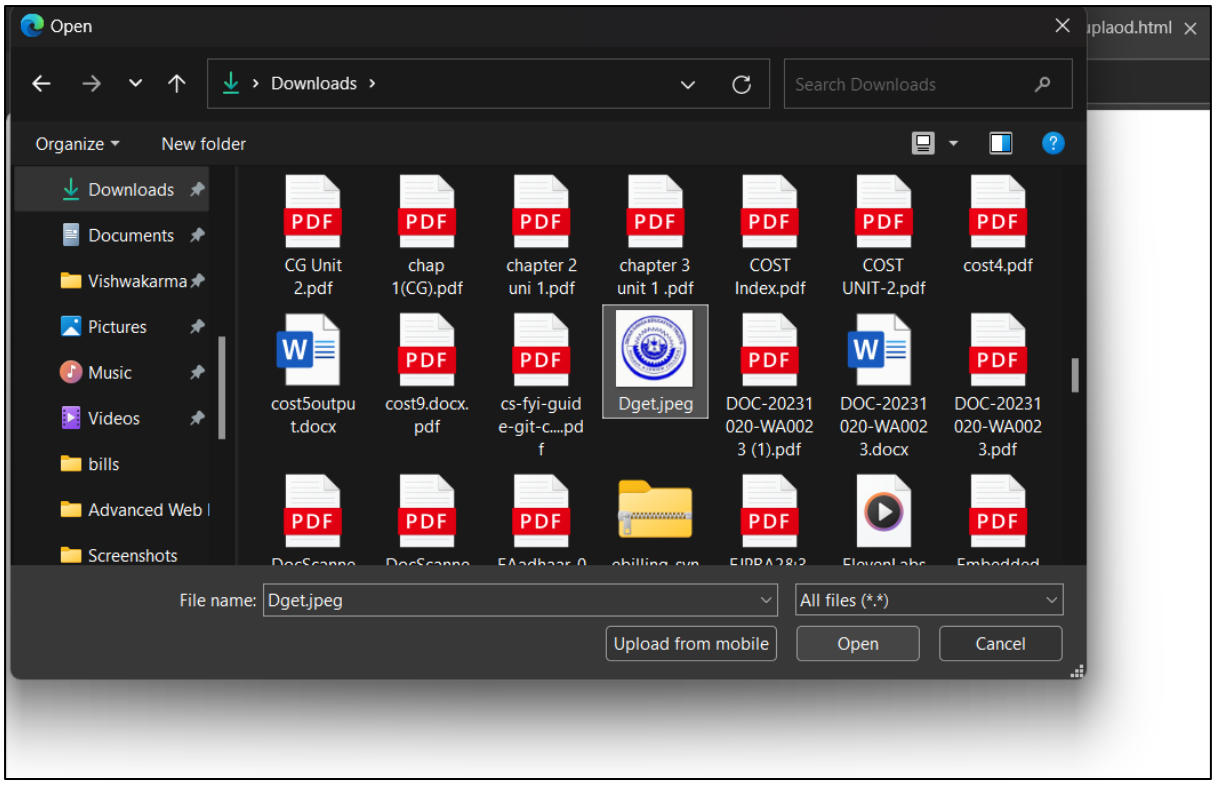
```
package filedownloadapp;
import java.io.IOException;
import java.io.InputStream;
import java.io.PrintWriter;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.ServletOutputStream;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet(name = "DownloadServlet", urlPatterns = {"/DownloadServlet"})
public class DownloadServlet extends HttpServlet {
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("APPLICATION/OCTET-STREAM");
String filename = request.getParameter("filename");
ServletContext context = getServletContext();
InputStream is = context.getResourceAsStream("/" + filename);
//ServletOutputStream out = response.getOutputStream(); // any of the two works
PrintWriter out=response.getWriter();
response.setHeader("Content-Disposition","attachment; filename=\"" + filename + "\"");
int i;
while ((i=is.read()) != -1) {
out.write(i);
}
is.close();
```

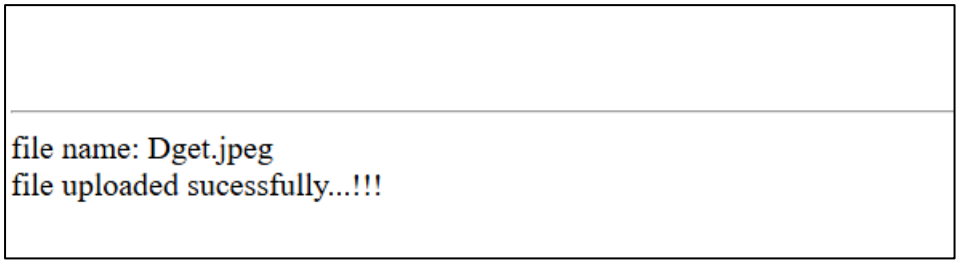
```
out.close();
}}
```

Output:

File Upload:





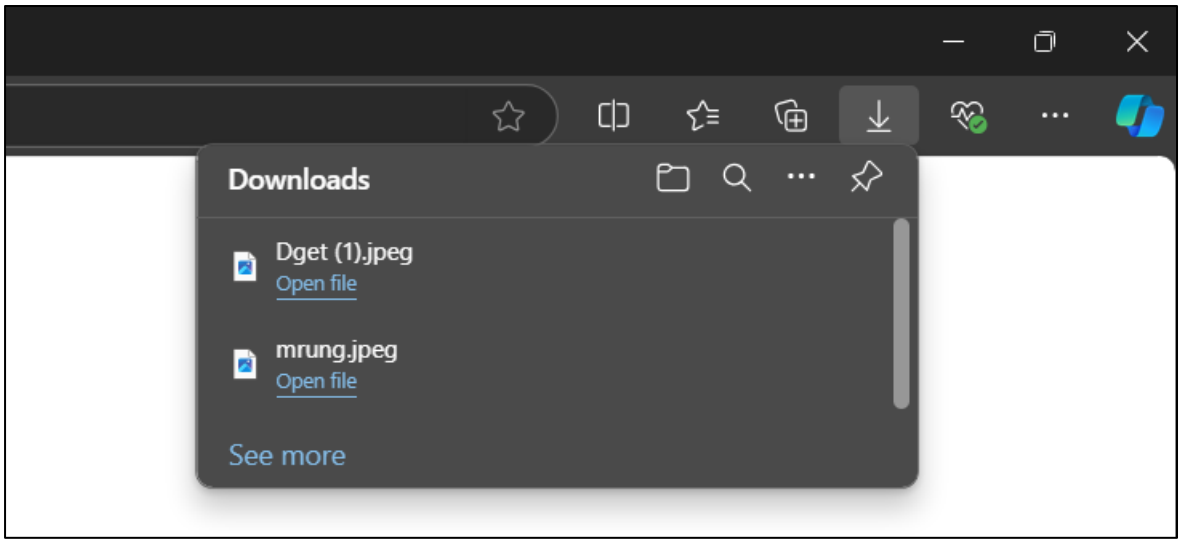


File Download:

File Download Application

Click Mr.Unknown

Click Hoimes Grp



Mr.Unknown.jpeg:

Dget.jpeg:

Aim: 3b. Develop Simple Servlet Question Answer Application using Database.

My Sql Queries:

```
create database qadb;
use qabd;
create table quiz (qno varchar(5) PRIMARY KEY,question varchar(100), op1 varchar(50),
op2 varchar(50), op3 varchar(50), op4 varchar(50), ans varchar(50))
insert into quiz values('001','What is the capital of India??', 'NewDelhi','Kolkata',
'Chennai', 'Mumbai', 'New Delhi');
insert into quiz values('002','Who was the First President of India??','Dr. Rajendra
Prasad','Dr. S. Radhakrishnan','Ram Nath Kovind','V. V. Giri','Dr. Rajendra Prasad');
insert into quiz values('003','What is ORM','Object Ratio Mean','Object Rotation
Measure','Object Relation Mapping','Oracle Request Management','Object Relation
Mapping');
insert into quiz values('004','Unit of Energy is ___','Dozon','Kilo Meter ','Joul','Hertz','Joul')
insert into quiz values('005',' --- is the smallest memory unit.','bit','byte','Kilo Byte','Giga
Byte','bit')
```

index.html

```
<html><head><title>Quiz Application</title></head>
<body>
<h1>Welcome to Quiz Servlet </h1>
<h1><a href="QuizServlet" >CLICK TO START QUIZ</a></h1>
</body>
</html>
```

QuizServlet.java

```
package mypack;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.sql.*;
@WebServlet(name = "QuizServlet", urlPatterns = {"/QuizServlet"})
public class QuizServlet extends HttpServlet {
@Override
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
out.println("<form action=ShowResult >");
try {
Class.forName("com.mysql.jdbc.Driver");
```

```
Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/qadb","root","root");
Statement stmt = con.createStatement();
ResultSet res = stmt.executeQuery("select * from quiz");
out.println("<table border=1 >");
int qno=0;
while(res.next()){
qno++;
out.println("<tr><td>"+res.getString(1)+"</td>");
out.println("<td>"+res.getString(2)+"</td></tr>");
out.println("<tr><td><input type=radio name="+qno+"
value="+res.getString(3)+"></td><td>"+res.getString(3)+"</td></tr>");
out.println("<tr><td><input type=radio name="+qno+"
value="+res.getString(4)+"></td><td>"+res.getString(4)+"</td></tr>");
out.println("<tr><td><input type=radio name="+qno+"
value="+res.getString(5)+"></td><td>"+res.getString(5)+"</td></tr>");
out.println("<tr><td><input type=radio name="+qno+"
value="+res.getString(6)+"></td><td>"+res.getString(6)+"</td></tr>");
}
}catch(Exception e){out.println(e);}
out.println("</table>");
out.println("<input type=reset >");
out.println("<input type=submit value=SUBMIT >");
out.println("</form>"); } }
```

ShowResult.java

```
package mypack;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.*;
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 = "ShowResult", urlPatterns = {"/ShowResult"})
public class ShowResult extends HttpServlet {
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
try {
Class.forName("com.mysql.jdbc.Driver");
```

```
Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/qadb","root","root");
Statement stmt = con.createStatement();
ResultSet res = stmt.executeQuery("select ans from quiz");
//out.print(request.getParameter("qno"));
int count =0, qno=0;
while(res.next()){
if(res.getString(1).equals(request.getParameter(""+(++qno))))
{ count++;
out.println("<h1>Correct </h1>");
}
else {
out.println("<h1>Incorrect </h1>");
}}
out.println("<h1>Your Score is "+count+" </h1>");
}catch(Exception e){out.println(e);}}}
```

Output:

Welcome to Quiz Servlet

CLICK TO START QUIZ

001	What is the capital of India??
◉	NewDelhi
○	Kolkata
○	Chennai
○	Mumbai
002	Who was the First President of India??
◉	Dr.RajendraPrasad
○	Dr.S.Radhakrishnan
○	RamNathKovind
○	V.V.Giri
003	What is ORM
◉	ObjectRatioMean
○	ObjectRotationMeasure
○	ObjectRelationMapping
○	OracleRequestManagement
004	Unit of Energy is
○	Dozon
○	Kilo Meter
◉	Joul
○	Hertz
005	is the smallest memory unit.
◉	bit
○	byte
○	Kilo Byte
○	Giga Byte

Reset SUBMIT

Correct

Correct

Incorrect

Correct

Correct

Your Score is 4

Aim: 3c. Create simple Servlet application to demonstrate Non-Blocking Read Operation.

index.html

```
<html>
<head>
<title>Non Blocking IO</title>
<meta charset="UTF-8">
<meta http-equiv="Refresh" content="0; URL=NonBlockingServlet">
</head>
<body>
</body>
</html>
```

NonBlockingServlet.java

```
package Tyit;
import java.io.*;
import java.net.*;
import javax.servlet.*;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;
@WebServlet(name = "NonBlockingServlet", urlPatterns = {"/NonBlockingServlet"})
public class NonBlockingServlet extends HttpServlet {
protected void service(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
try (PrintWriter out = response.getWriter()) {
out.println("<h1>FileReader</h1>");
String filename="/WEB-INF/booklist.txt";
ServletContext c=getServletContext();
InputStream in=c.getResourceAsStream(filename);
String
path="http://"+request.getServerName()+":"+request.getServerPort()+request.getConte
xtPath()+"/ReadingNonBloclingServlet";
URL url=new URL(path);
HttpURLConnection conn=(HttpURLConnection)url.openConnection();
conn.setChunkedStreamingMode(2);
conn.setDoOutput(true);
conn.connect();
if(in!=null)
{
InputStreamReader inr=new InputStreamReader(in);
BufferedReader br = new BufferedReader(inr);
```

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

ReadingListener.java

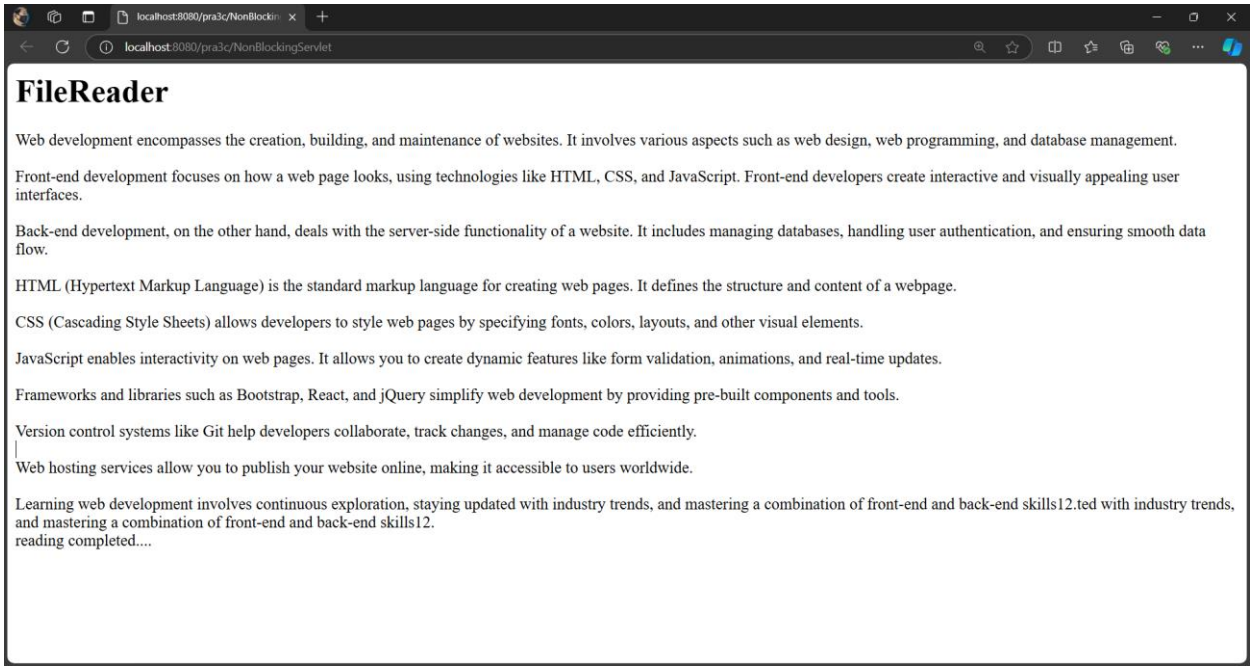
```
package Tyit;
import java.io.IOException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.AsyncContext;
import javax.servlet.ReadListener;
import javax.servlet.ServletInputStream;
import javax.servlet.annotation.WebServlet;
@WebServlet(name = "ReadingListener", urlPatterns = {"/ReadingListener"})
public class ReadingListener implements ReadListener
{
private ServletInputStream input = null;
private AsyncContext ac = null;
ReadingListener(ServletInputStream in, AsyncContext c) {
input = in;
ac = c;
}
@Override
public void onDataAvailable() throws IOException {
}
@Override
public void onAllDataRead() throws IOException {
ac.complete();
}
@Override
public void onError(final Throwable t) {
ac.complete();
```

```
t.printStackTrace();
}}
```

ReadingNonBlockingServlet.java

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

Output:



Practical No.4

Aim: 4a. 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">
    <title>JSP Page</title>
  </head>
  <body>
    <h1>Use of Intrinsic Objects in JSP</h1>
<h1>Request Object </h1>
Query String <%=request.getQueryString() %><br>
Context Path <%=request.getContextPath() %><br>
Remote Host <%=request.getRemoteHost() %><br>
<h1>Response Object </h1>
Character Encoding Type <%=response.getCharacterEncoding() %><br>
Content Type <%=response.getContentType() %><br>
Locale <%=response.getLocale() %><br>
<h1>Session Object </h1>
ID <%=session.getId() %><br>
Creation Time <%=new java.util.Date(session.getCreationTime()) %><br>
Last Access Time<%=new java.util.Date(session.getLastAccessedTime()) %><br>
  </body>
</html>
```

Output:


JSP Page
localhost:8080/pra4ajsp/newjsp.jsp
Use of Intrinsic Objects in JSP
Request Object
Query String null
Context Path /pra4ajsp
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_US
Session Object
ID fcfa6c8b1456ff9165f6695f1e6a
Creation Time Wed Oct 09 11:15:32 IST 2024
Last Access TimeWed Oct 09 11:15:32 IST 2024

Aim: 4b. 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.html

```
<html><head><title>User Information Paage</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" import="mypack.*" %>
<html><head><title>JSP Page</title></head>
<body>
<h1>Validation Page</h1>
<jsp:useBean id="obj" scope="request"
 class="mypack.CheckerBean" >
<jsp:setProperty name="obj" property="*"/>
</jsp:useBean>
<%if (obj.validate())
{ %>
<jsp:forward page="successful.jsp"/>
<% }
else {%>
<jsp:include page="index.html"/>
<%}%>
<%=obj.getError() %>
```

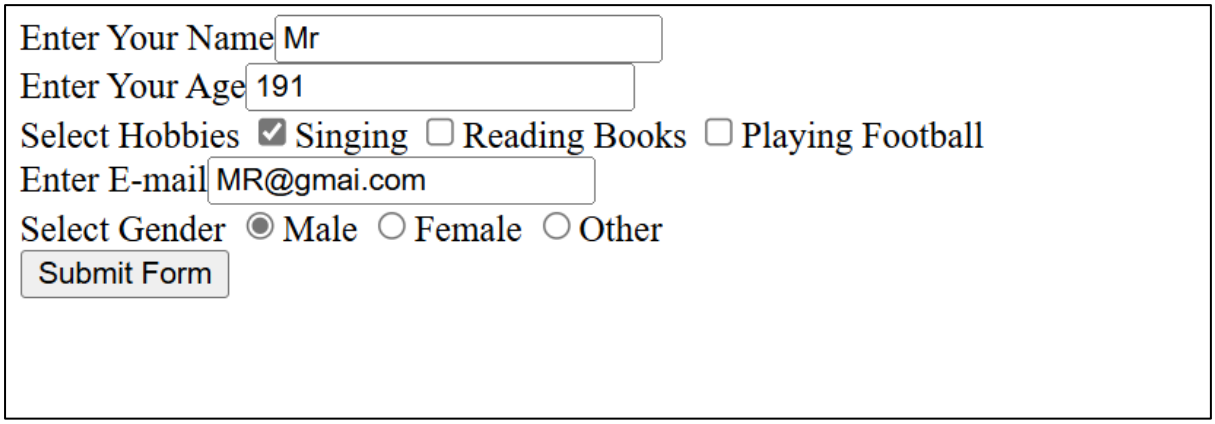
```
</body></html>
```

CheckerBean.java

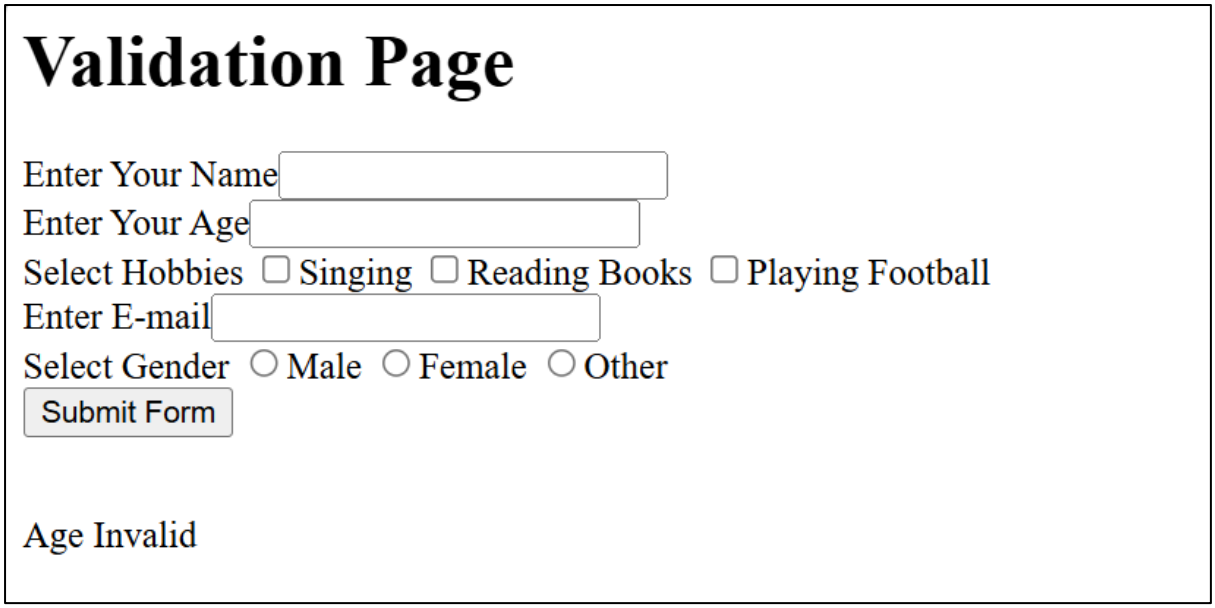
```
package mypack;
public class CheckerBean {
private String name, age, hob, email, gender, error;
public CheckerBean(){error="";}
public void setName(String n){name=n;}
public void setAge(String a){age=a;}
public void setHob(String h){hob=h;}
public void setEmail(String e){email=e;}
public void setGender(String g){gender=g;}
public void setError(String e){error=e;}
public String getName(){return name;}
public String getAge(){return age;}
public String getHob(){return hob;}
public String getEmail(){return email;}
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.length() > 2 )
{error+="<br>Age Invalid";res=false;}
return res;
}
}
```

Output:

Index.html



Validate.jsp



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

MySql queries:

```
create database LoginDB;
use LoginDB;
create table user(username varchar(20) PRIMARY KEY, password varchar(20), email varchar(20),
country varchar(20));
insert into user values ('admin','admin','admin@admin.com','India');
```

Register.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 <input type="text" name="txtCon" ><br>
<input type="reset" ><input type="submit" value="REGISTER" >
</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");
 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 FFFFFAAAIIILLLL !!!!");
 %><jsp:include page="Register.html" ></jsp:include>
<%
 }
 }catch(Exception e){out.println(e);}
 }
 else
 {
 out.println("<h1>Password Mismatch</h1>");
%>
<jsp:include page="Register.html" ></jsp:include>
<% }
 %>
</body>
</html>
```

Login.html

```
<html>
<body>
<h1>Login Page</h1>
<form action="Login.jsp" >
 Enter User Name <input type="text" name="txtName" ><br>
 Enter Password <input type="password" name="txtPass" ><br>
<input type="reset" ><input type="submit" value="~~~LOGIN~~" >
</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");
 try{
 Class.forName("com.mysql.jdbc.Driver");
 Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/logindb");
```

```
 PreparedStatement stmt = con.prepareStatement("select password from user where
username=?");
 stmt.setString(1, uname);
 ResultSet rs = stmt.executeQuery();
 if(rs.next()){
 if(pass.equals(rs.getString(1)))
 {
 out.println("<h1>~~~ LOGIN SUCCESSFULLL ~~~ </h1>");
 } }
 else{
 out.println("<h1>User Name not exist !!!!!</h1>");
 %>
<jsp:include page="Register.html" ></jsp:include>
<%
 }
 }catch(Exception e){out.println(e);}
 %>
</body></html>
```

Output:

Registration:

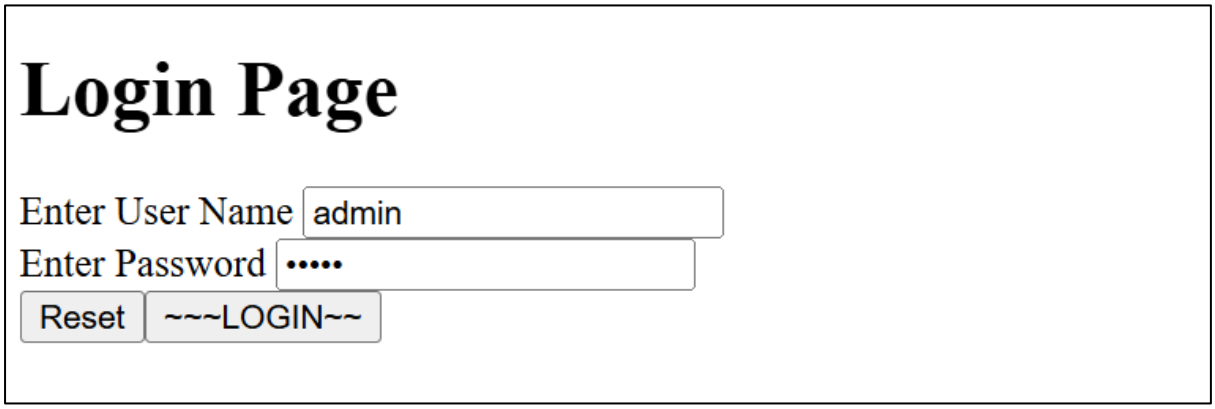
Database

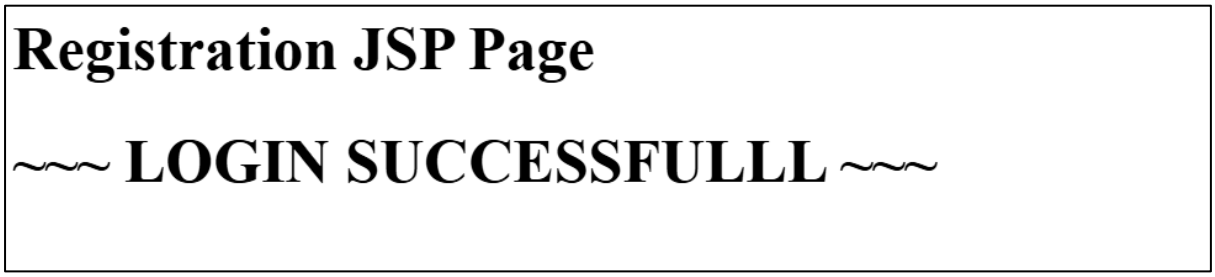
select * from user ×

Max. rows: 100 Fetched Rows: 9 Matching Rows:

#	username	password	email	country
1	admin	admin	admin@admin.com	India
2	lelala	wdw	sa	India
3	priyanshu	12345	emai@gmail.com	India
4	nobita	123456	sakk@gmail.com	India
5	shizuka	cat	zat	lat
6	sushil	sandeep	san@gmail.com	India
7	Sandeep Vishwakarma	123	sa	India
8	Ritesh	12345	sakk@gmail.com	India
9	sharon	suraj	saj@gmail.com	India

Login:





Practical No. 5

Aim: 5a. 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.

MySQL queries:

```
create table emp(empid varchar(10) PRIMARY KEY, ename varchar(50), salary
varchar(50),age
varchar(50) )
insert into emp values('1','aaa','221234','11')
insert into emp values('2','bbb','334567','22')
insert into emp values('3','ccc','44454','33')
insert into emp values('4','ddd','55123','44')
```

index.html

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

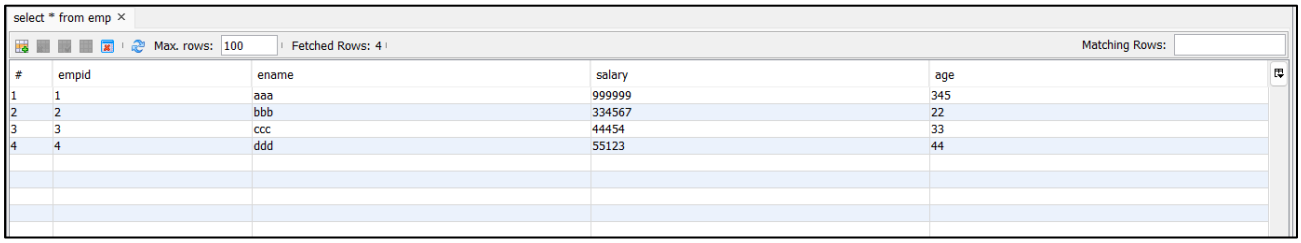
UpdateEmp.java

```
<%@page contentType="text/html" import="java.sql.*" %>
<html><body>
<h1>Employee Record Update</h1>
<%
 String eno=request.getParameter("txtEno");
 String name=request.getParameter("txtName");
 String age = request.getParameter("txtAge");
 String sal = request.getParameter("txtSal");
 try{
 Class.forName("com.mysql.jdbc.Driver");
 Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/empdb");
 PreparedStatement stmt = con.prepareStatement("select * from emp where empid=?");
 stmt.setString(1, eno);
 ResultSet rs = stmt.executeQuery();
```

```
 if(rs.next()){
 out.println("<h1>~~~ Employee "+name+" Exist ~~~ </h1>");
 PreparedStatement pst1= con.prepareStatement("update emp set salary=? where
empid=?");
 PreparedStatement pst2= con.prepareStatement("update emp set age=? where
empid=?");
 pst1.setString(1, sal); pst1.setString(2, eno);
 pst2.setString(1, age); pst2.setString(2, eno);
 pst1.executeUpdate(); pst2.executeUpdate();
 }
else{
 out.println("<h1>Employee Record not exist !!!!!</h1>");
 }
 }catch(Exception e){out.println(e);}
 %></body></html>
```

Output:

Before updating:

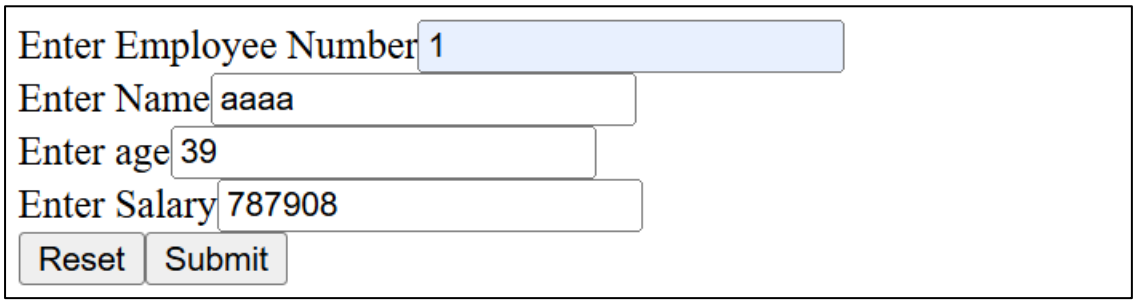


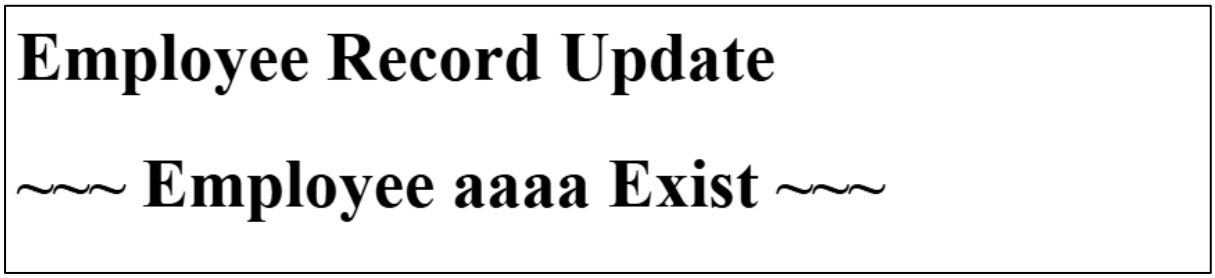
select * from emp

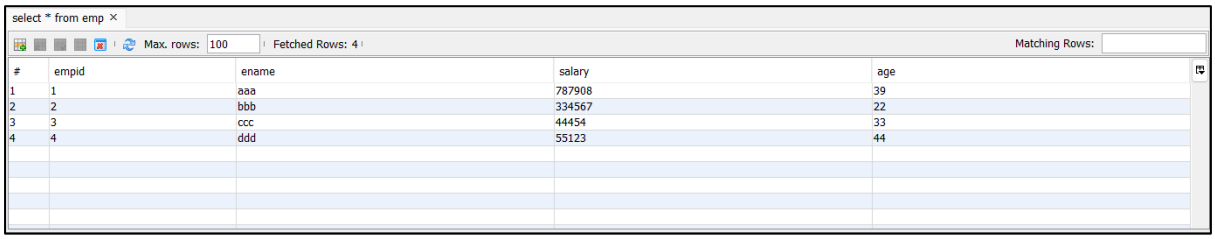
Max. rows: 100 Fetched Rows: 4 Matching Rows:

#	empid	ename	salary	age
1	1	aaa	999999	345
2	2	bbb	334567	22
3	3	ccc	44454	33
4	4	ddd	55123	44

After updated:







select * from emp

Max. rows: 100 Fetched Rows: 4 Matching Rows:

#	empid	ename	salary	age
1	1	aaa	787908	39
2	2	bbb	334567	22
3	3	ccc	44454	33
4	4	ddd	55123	44

Aim: 5b. Create a JSP page to demonstrate the use of Expression language.

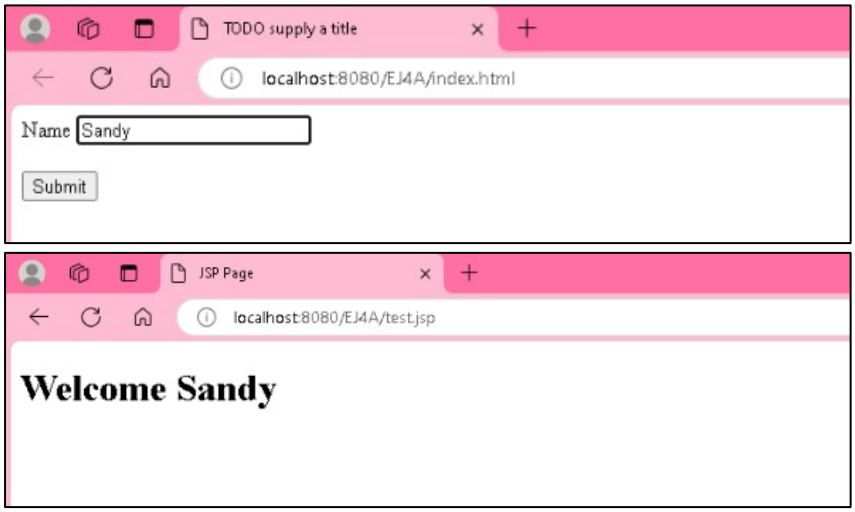
index.html

```
<html>
  <head>
    <title>Expression language.</title>
    <meta charset="UTF-8"T>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <form method="post" action="test.jsp">
      Name <input type="text" name="t1"><br><br>
      <input type="submit">
    </form>
  </body>
</html>
```

test.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>Welcome ${param.t1} </h1>
  </body>
</html>
```

Output:



Aim: 5c. Create a JSP application to demonstrate the use of JSTL

MySql queries:

```
CREATE DATABASE IF NOTEXISTS sampleDB;
CREATE TABLE `product` (
`id` int(10) unsigned NOTNULL auto_increment,
`pname` varchar(45) NOTNULL,
`quantity` int(10) unsigned NOTNULL,
PRIMARYKEY (`id`)
);
INSERT INTO`product` (`id`,`pname`,`quantity`) VALUES
(1,'Mouse',50),
(2,'Keyboard',5),
(3,'Monitor',34);
```

index.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<metahttp-equiv="Content-Type"content="text/html; charset=UTF-8">
<title>JSP Page</title>
</head>
<body>
<h1>Choose Option</h1>
<ahref="insert.jsp">Insert Record</a><p></p>
<ahref="display.jsp">Display Record</a>
</body>
</html>
```

insert.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
<head>
<metahttp-equiv="Content-Type"content="text/html; charset=UTF-8">
<title>JSP Page</title>
</head>
<body>
<formaction="insertdb.jsp"method="post">
<tableborder="0"cellspacing="2"cellpadding="5">
<thead>
<tr>
```

```
<thcolspan="2">Purchase Product</th>
</tr>
</thead>
<tbody>
<tr>
<td><label>Product Name</label></td>
<td><inputtype="text"name="pname"/></td>
</tr>
<tr>
<td><label>Quantity</label></td>
<td><inputtype="text"name="qty"/></td>
</tr>
<tr>
<td><inputtype="submit"value="Save"/></td>
<td><inputtype="reset"value="reset"/></td>
</tr>
</tbody>
</table>
</form>
<fontcolor="red"><c:iftest="${not empty param.errMsg}">
<c:outvalue="${param.errMsg}"/>
<ahref="index.jsp">Go Back</a>
</c:if></font>
<fontcolor="green"><c:iftest="${not empty param.susMsg}">
<c:outvalue="${param.susMsg}"/>
<ahref="index.jsp">Go Back</a>
</c:if></font>
</body>
</html>
```

insertdb.jsp

```
<%@ page import="java.io.*,java.util.*,java.sql.*"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql"%>
<html>
<head>
<title>INSERT Operation</title>
</head>
<body>
<c:iftest="${ empty param.pname or empty param.qty}">
<c:redirecturl="insert.jsp">
<c:paramname="errMsg"value="Please Enter Product and Quantity"/>
</c:redirect>
</c:if>
```

```
<sql:setDataSourcevar="dbsource"driver="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost/sampleDB"
user="root" password="pass"/>
<sql:updatedataSource="${dbsource}"var="result">
INSERT INTO product(pname, quantity) VALUES (?,?);
<sql:paramvalue="${param.pname}"/>
<sql:paramvalue="${param.qty}"/>
</sql:update>
<c:iftest="${result>=1}">
<fontsize="5"color='green'> Congratulations ! Data inserted
successfully.</font>
<c:redirecturl="insert.jsp">
<c:paramname="susMsg"value="Congratulations ! Data inserted
successfully." />
</c:redirect>
</c:if>
</body>
</html>
```

display.jsp

```
<%@ page import="java.io.*,java.util.*,java.sql.*"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql"%>
<html>
<head>
<title>SELECT Operation</title>
<script>
function confirmGo(m,u) {
if ( confirm(m) ) {
window.location = u;
}
}
</script>
</head>
<body>
<sql:setDataSourcevar="dbsource"driver="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost/sampleDB"
user="root" password="pass"/>
<sql:querydataSource="${dbsource}"var="result">
SELECT * from product;
</sql:query>
<center>
<form>
<tableborder="1"width="40%">
```

```
<caption>Product List</caption>
<tr>
<th>Product ID</th>
<th>Product Name</th>
<th>Quantity</th>
<thcolspan="2">Action</th>
</tr>
<c:forEachvar="row"items="${result.rows}">
<tr>
<td><c:outvalue="${row.id}"/></td>
<td><c:outvalue="${row.pname}"/></td>
<td><c:outvalue="${row.quantity}"/></td>
<td><ahref="update.jsp?id=<c:out
value="${row.id}"/>">Update</a></td>
<td><ahref="javascript:confirmGo('Sure to delete this
record?','deletedb.jsp?id=<c:out value="${row.id}"/>')">Delete</a></td>
</tr>
</c:forEach>
</table>
</form>
<ahref="index.jsp">Go Home</a>
</center>
</body>
</html>
```

update.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ page import="java.io.*,java.util.*,java.sql.*"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql"%>
<!DOCTYPE html>
<html>
<head>
<metahttp-equiv="Content-Type"content="text/html; charset=UTF-8">
<title>JSP Page</title>
</head>
<body>
<sql:setDataSourcevar="dbsource"driver="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost/sampleDB"
user="root" password="pass"/>
<sql:querydataSource="${dbsource}"var="result">
SELECT * from product where id=?;
<sql:paramvalue="${param.id}"/>
</sql:query>
```

```
<formaction="updatedb.jsp"method="post">
<tableborder="0"width="40%">
<caption>Update Product</caption>
<tr>
<th>Product Name</th>
<th>Quantity</th>
</tr>
<c:forEachvar="row"items="${result.rows}">
<tr>
<td><inputtype="hidden"value="${param.id}"name="id"/>
<inputtype="text"value="${row.pname}"name="pname"/></td>
<td><inputtype="text"value="${row.quantity}"name="qty"/></td>
<td><inputtype="submit"value="Update"/></td>
</tr>
</c:forEach>
</table>
<ahref="index.jsp">Go Home</a>
</form>
</body>
</html>
```

updatedb.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ page import="java.io.*,java.util.*,java.sql.*"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql"%>
<!DOCTYPE html>
<html>
<head>
<metahttp-equiv="Content-Type"content="text/html; charset=UTF-8">
<title>JSP Page</title>
</head>
<body>
<sql:setDataSourcevar="dbsource"driver="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost/sampleDB"
user="root" password="pass"/>
<sql:updatedataSource="${dbsource}"var="count">
UPDATE product SET pname = ?, quantity=?
WHERE id='${param.id}'
<sql:paramvalue="${param.pname}"/>
<sql:paramvalue="${param.qty}"/>
</sql:update>
<c:iftest="${count>=1}">
<fontsize="5"color='green'> Congratulations ! Data updated
```

```
successfully.</font>
<ahref="index.jsp">Go Home</a>
</c:if>
</body>
</html>
```

deletedb.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ page import="java.io.*,java.util.*,java.sql.*"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/sql" prefix="sql"%>
<!DOCTYPE html>
<html>
<head>
<metahttp-equiv="Content-Type"content="text/html; charset=UTF-8">
<title>JSP Page</title>
</head>
<body>
<sql:setDataSourcevar="dbsource"driver="com.mysql.jdbc.Driver"
url="jdbc:mysql://localhost/sampleDB"
user="root" password="pass"/>
<sql:updatedataSource="${dbsource}"var="count">
DELETE FROM product
WHERE id='${param.id}'
</sql:update>
<c:iftest="${count>=1}">
<fontsize="5"color='green'> Congratulations ! Data deleted
successfully.</font>
<ahref="index.jsp">Go Home</a>
</c:if>
</body>
</html>
```

Output:

Choose Option

Insert Record

Display Record

Insert Record:

Purchase Product

Product Name: Gaming Laptop

Quantity: 5

Save reset

Purchase Product

Product Name:

Quantity:

Save reset

Congratulations ! Data inserted successfully. Go Back

Display Record:

Product List

Product ID	Product Name	Quantity	Action	
1	Mouse	50	Update	Delete
2	Keyboard	5	Update	Delete
3	Monitor	34	Update	Delete
5	pen	0	Update	Delete
6	pen	0	Update	Delete
7	Gaming Laptop	5	Update	Delete

Go Home

Practical No. 6

Aim: 6a. Create a Currency Converter application using EJB.

index.html

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

CCServlet.java

```
package mypack;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.ejb.EJB;
import mybeans.CCBean;
public class CCServlet extends HttpServlet {
 @EJB CCBean obj;
public void doGet(HttpServletRequest request, HttpServletResponse response)
 throws ServletException, IOException {
 response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
double amt = Double.parseDouble(request.getParameter("amt"));
if(request.getParameter("type").equals("r2d"))
{
out.println("<h1>"+amt+ " Rupees = "+obj.r2Dollor(amt)+" Dollors</h1>");
}
if(request.getParameter("type").equals("d2r"))
{
out.println("<h1>"+amt+ " Dollors = "+obj.d2Rupees(amt)+" Rupees</h1>");
}
}}
```

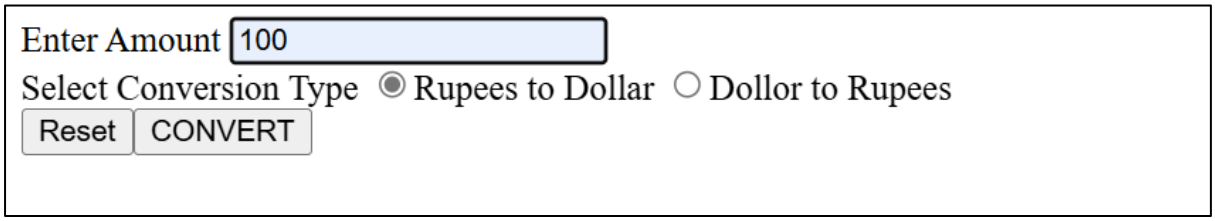
CCBean.java

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

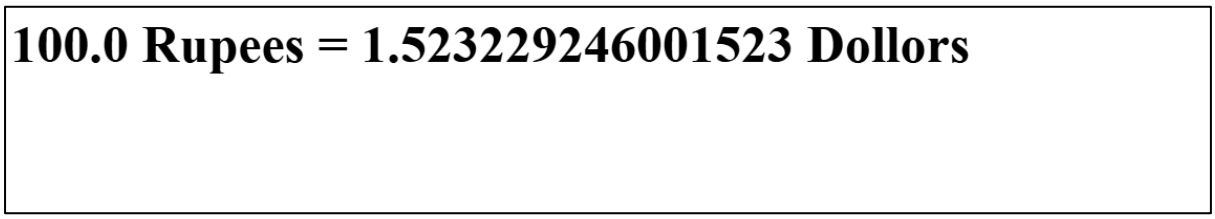
```
@Stateless
public class CCBean {
public CCBean(){}
public double r2Dollor(double r){ return r/65.65; }
public double d2Rupees(double d){ return d*65.65; }
}
```

Output:

Conversion Rupees to Dollar:



Rupees Converted to Dollar:



Aim: 6b. Develop a Simple Room Reservation System Application Using EJB.

MySQL queries:

```
Create table rookbook(RoomId varchar(4) PRIMARY KEY, RoomType varchar(20),
charges
number(5,2), cust varchar(20), mob varchar(20) , status varchar(10))
insert into roombook values('1001','Delux',5000.00,'','','Not Booked')
insert into roombook values('1002','Super Delux',7000.00,'','','Not Booked')
insert into roombook values('1003','Suit',9500.00,'','','Not Booked')
insert into roombook values('2001','Delux',5000.00,'','','Not Booked')
insert into roombook values('2002','Super Delux',7000.00,'','','Not Booked')
insert into roombook values('2003','Suit',9500.00,'','','Not Booked')
```

index.html

```
<form action="RBServlet" >
 Select a room Type
<input type="radio" name="txtType" value="Delux">Delux
<input type="radio" name="txtType" value="Super Delux">Super Delux
<input type="radio" name="txtType" value="Suit">Suit<br>
 Enter Your Name<input type="text" name="txtCust" ><br>
 Enter Mobile No.<input type="text" name="txtMob" ><br>
<input type="reset" ><input type="submit" value="Book Room">
</form>
```

RBServlet.java

```
package mypack;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.ejb.EJB;
import mybeans.RRBean;
public class RBServlet extends HttpServlet {
@EJB RRBean obj;
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException{
PrintWriter out=response.getWriter();
String rt=request.getParameter("txtType");
String cn=request.getParameter("txtCust");
String cm=request.getParameter("txtMob");
String msg = obj.roomBook(rt, cn, cm);
 out.println(msg);
}}
```

RRBean.java

```
package mypack;
import javax.ejb.Stateless;
import java.sql.*;
@Stateless
public class RRBean {
public RRBean(){}
public String roomBook(String rt, String cn, String cm){
String msg="";
try{
Class.forName("com.mysql.jdbc.Driver");
Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/rrdb","root","root");
String query="select * from roombook where RoomType=? and status='Not Booked'";
PreparedStatement pst = con.prepareStatement(query);
pst.setString(1,rt);
ResultSet rs= pst.executeQuery();
if(rs.next()){
 String rno=rs.getString(1);
PreparedStatement stm1 = con.prepareStatement("update roombook set cust=? where
RoomId=? ");
PreparedStatement stm2 = con.prepareStatement("update roombook set mobile=?
where RoomId=? ");
PreparedStatement stm3 = con.prepareStatement("update roombook set status=?
where RoomId=? ");
stm1.setString(1,cn); stm1.setString(2,rno);
stm2.setString(1,cm); stm2.setString(2,rno);
stm3.setString(1, "Booked"); stm3.setString(2,rno);
stm1.executeUpdate();
stm2.executeUpdate();
stm3.executeUpdate();
msg = "Room "+rno+ " Booked <br> Charges = "+rs.getString(3);
}
else
{
msg = "Room "+rt+ " currently Not available”.
}
} catch(Exception e){msg=""+e;}
return msg;}}
```

Output:

Room Reservation:

Dget Room Reservation Page

Select a room Type ○ Delux ○ Super Delux ◉ Suit
Enter Your Name Raju
Enter Mobile No. 123456789
Reset Book Room

```
Room 1003 Booked <br> Charges = 9500
```

Database:

SELECT * FROM roombook LI... ×

Max. rows: 100 Fetched Rows: 6 Matching Rows:

#	RoomId	RoomType	charges	Customber	Mob	Status
1	1001	Delux	5000	yash	09920861483	Booked
2	1002	Super Delux	7000			Not Booked
3	1003	Suit	9500	Raju	123456789	Booked
4	2001	Delux	5000	Ritesh	099208614	Booked
5	2002	Super Delux	7000	priyanshu	44357889	Booked
6	2003	Suit	9500	sharon	09785643	Booked

Aim: 6c. Develop simple shopping cart application using EJB [Stateful Session Bean]

CartBeanLocal.java

```
package cart;
import java.util.List;
import javax.ejb.Local;
@Local
public interface CartBeanLocal {
 public void initialize(String person) throws Exception;
 public void initialize(String person, String id)
 throws Exception;
 public void addBook(String title);
 public void removeBook(String title) throws Exception;
 public List<String> getContents();
 public void remove();
}
```

CartBean.java

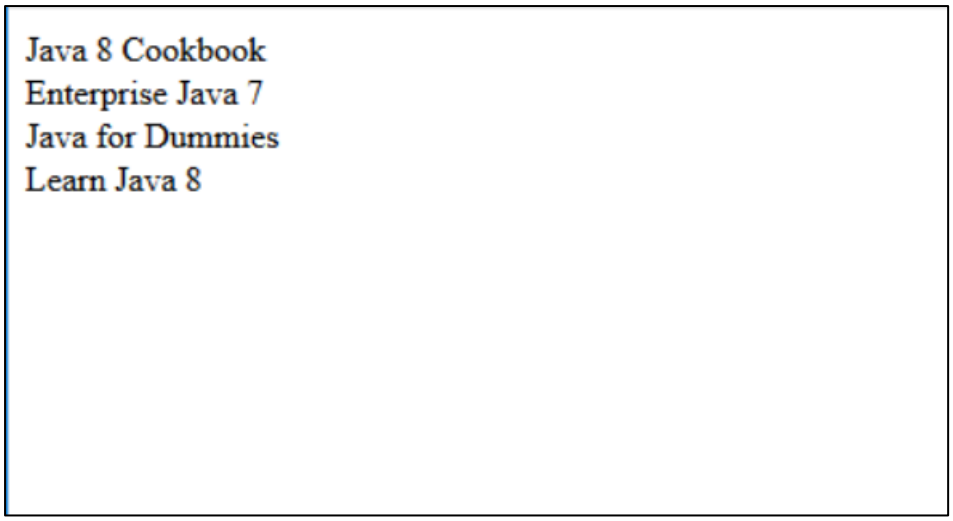
```
package cart;
import java.util.ArrayList;
import java.util.List;
import javax.ejb.Remove;
import javax.ejb.Stateful;
@Stateful
public class CartBean implements CartBeanLocal {
 String customerName;
 String customerId;
 List<String> contents;
 public void initialize(String person, String id)
 throws Exception {
 if (person == null) {
 throw new Exception("Null person not allowed.");
 } else {
 customerName = person;
 }
 if ( person=="ABC" && id=="123") {
customerId = id;
 } else {
 throw new Exception("Invalid id: " + id);
 }
 contents = new ArrayList<String>();
 }
 public void addBook(String title) {
```

```
 contents.add(title);
 }
 public void removeBook(String title) throws Exception {
 boolean result = contents.remove(title);
 if (result == false) {
 throw new Exception(title + " not in cart.");
 }
 }
 public List<String> getContents() {
 return contents;
 }
 @Remove
 public void remove() {
 contents = null;
 }
}
```

```
package testcart;
import cart.CartBeanLocal;
import java.io.*;
import java.util.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.naming.*;
import javax.servlet.*;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;
@WebServlet(name = "CartTestServlet", urlPatterns = {"/CartTestServlet"})
public class CartTestServlet extends HttpServlet {
 CartBeanLocal cartBean = lookupCartBeanLocal();
 @Override
 protected void doGet(HttpServletRequest request, HttpServletResponse response)
 throws ServletException, IOException {
 response.setContentType("text/html;charset=UTF-8");
 try{
 cartBean.initialize("ABC", "123");
 }catch(Exception e){}
 cartBean.addBook("Java 8 Cookbook");
 cartBean.addBook("Enterprise Java 7 ");
 cartBean.addBook("Java for Dummies");
 cartBean.addBook("Learn Java 8");
 try (PrintWriter out = response.getWriter()) {
try{
 List<String> books = cartBean.getContents();
```

```
 for( String s : books)
 out.println(s +"<br />");
 }catch(Exception e){}
 }
 }
 private CartBeanLocal lookupCartBeanLocal() {
 try {
 Context c = new InitialContext();
 return (CartBeanLocal)
c.lookup("java:global/EnterpriseApplication1/EnterpriseApplication1-
ejb/CartBean!cart.CartBeanLocal");
 } catch (NamingException ne) {
 Logger.getLogger(getClass().getName()).log(Level.SEVERE, "exception caught", ne);
 throw new RuntimeException(ne);
 }
 }
}
```

Output:



Practical No. 7

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

Index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Servlet Client</title>
    <meta http-equiv="Refresh" content="0; URL=Servletclient">
  </head>
  <body>
  </body>
</html>
```

CountServletHitsBean.java

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

ServletClient.java

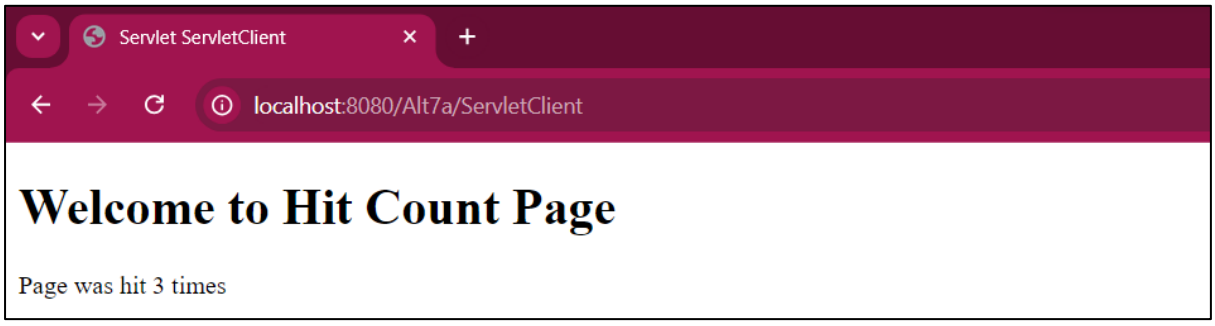
```
package servlet;
import ejb.CountServletHitsBean;
import java.io.IOException;
import java.io.PrintWriter;
import javax.ejb.EJB;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet(name = "ServletClient", urlPatterns = {"/ServletClient"})
public class ServletClient extends HttpServlet {
  @EJB
  CountServletHitsBean counterBean;
  @Override
```

```
    protected void service(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet ServletClient</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<h1>Welcome to Hit Count Page</h1>");
            out.println("page was hit " + counterBean.incrementAndGetHitCount() + " times");
            out.println("</body>");
            out.println("</html>");
        }
    }
}
```

Output:

Edge Browser:

Chrome Browser:



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

index.jsp

```
<%@page import="javax.jms.JMSException, javax.naming.InitialContext,
javax.jms.TextMessage, javax.jms.MessageProducer, javax.jms.Session,
javax.jms.Connection, javax.jms.Queue, 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>Welcome To Dget 's Home Page</title>
  </head>
  <body style="background-color: pink;">
    <h1>Welcome To Dget Home page</h1>
<%
  try {
    InitialContext ctx = new InitialContext();
    queue = (Queue) ctx.lookup("jms/Queue");
    connectionFactory = (ConnectionFactory) ctx.lookup("jms/QueueFactory");
    connection = connectionFactory.createConnection();
    mysession = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);
    messageProducer = mysession.createProducer(queue);
    message = mysession.createTextMessage();
    message.setText(request.getRemoteAddr());
    messageProducer.send(message);
  } catch (JMSException e) {
    System.out.println("Exception occurred: " + e.toString());
  }
%>
  </body>
</html>
```

BasicMessageBean.java

```
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 = "destinationType", propertyValue =
"javax.jms.Queue"),
  @ActivationConfigProperty(propertyName = "destinationLookup", propertyValue =
"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();
    }
  }
}
```

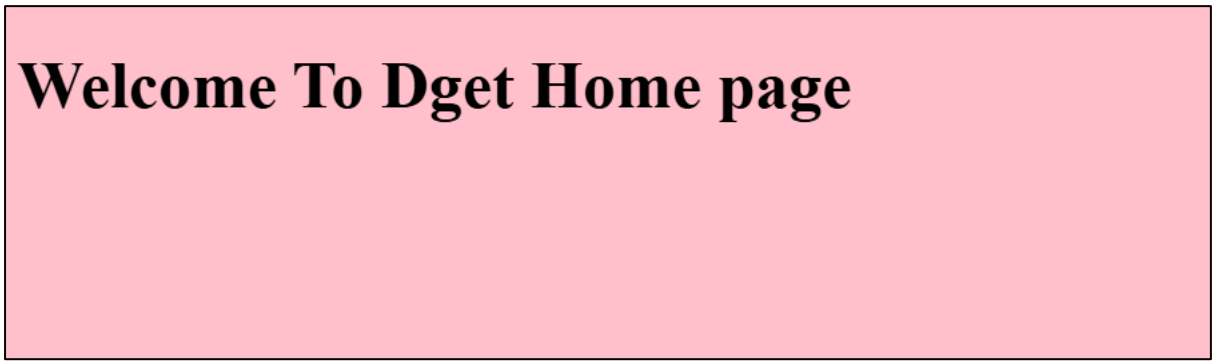
VisitorStatBean.java

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

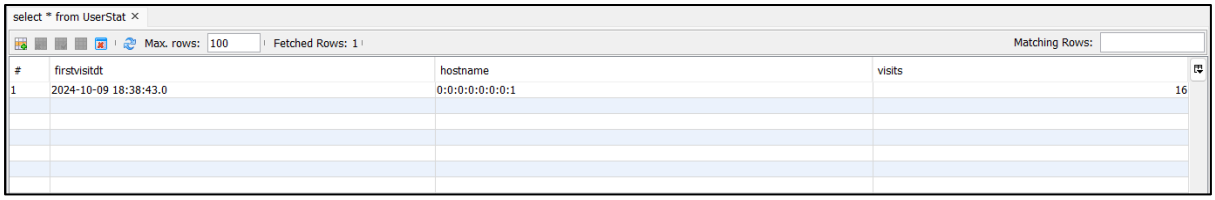
```
import javax.annotation.PostConstruct;
import javax.annotation.PreDestroy;
import javax.ejb.Stateless;
@Stateless
public class VisitorStatBean {
  private Connection conn = null;
  private ResultSet rs;
  private Statement stmt = null;
  private String query = null;
  @PostConstruct
  public void connect() {
    try {
      Class.forName("com.mysql.jdbc.Driver").newInstance();
      conn = DriverManager.getConnection("jdbc:mysql://localhost/mysql", "root",
"root");
      System.out.println("Database connection established successfully.");
    } catch(ClassNotFoundException | InstantiationException | IllegalAccessException |
SQLException e) {
      System.err.println("Sorry failed to connect to the Database.");
    }
  }
  @PreDestroy
  public void disconnect() {
    try {
      conn.close();
      System.out.println("Database connection closed successfully.");
    } catch(SQLException e) {
      System.err.println("Cannot close the database connection: " + e.getMessage());
    }
  }
  public void addVisitor(String host) {
    try {
      stmt = conn.createStatement();
      query = "INSERT INTO UserStat (hostname, visits) VALUES('" + host + "','1')";
      stmt.executeUpdate(query);
    } catch(SQLException e) {
      try {
        stmt = conn.createStatement();
        query = "UPDATE UserStat SET visits = visits + 1 WHERE hostname = '" + host +
"'";
        stmt.executeUpdate(query);
      } catch(SQLException ex) {
        System.err.println("Cannot update: " + ex.getMessage());
      }
```

```
    }
  }
}
```

Output:



Database:



Aim: 7c. Develop simple Marks Entry Application to demonstrate accessing Database using EJB.

index.html

```
<html>
   <head>
      <title>TODO supply a title</title>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
   </head>
   <body>
  <form action="studentServlet" >
Enter Your Roll no<input type="text" name="txtroll" ><br>
Enter Your Name<input type="text" name="txtname" ><br>
Enter Your Class<input type="text" name="txtclass" ><br>
Enter Your subject1 marks<input type="text" name="txts1" ><br>
Enter Your subject2 marks<input type="text" name="txts2" ><br>
Enter Your subject3 marks<input type="text" name="txts3" ><br>
<input type="reset" ><input type="submit" value="submit">
</form>
   </body>
</html>
```

Studentbean.java

```
package mypack;
import javax.ejb.Stateless;
import java.sql.*;
@Stateless
public class studentbean implements studentbeanLocal {
   @Override
   public String insertmark(int rno, String n, String c, int s1, int s2, int s3)
   {
   String msg="";
      try{
   Class.forName("com.mysql.jdbc.Driver");
  Connection con
=DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb2","root","1234");
PreparedStatement stm1 = con.prepareStatement("insert into student (rollno , name, class ,
sub1 ,sub2 ,sub3)values(?,?,?,?,?,?)");
       stm1.setInt(1,rno);
       stm1.setString(2,n);
       stm1.setString(3,c);
       stm1.setInt(4,s1);
       stm1.setInt(5,s2);
       stm1.setInt(6,s3);
       stm1.executeUpdate();
      msg="insert successful";
```

```
        }catch(Exception e)
            {
               msg=""+e;
            }
    return msg;
    } }
```

studentServlet.java

```
package mypack;
import javax.ejb.EJB;
import java.io.IOException;
import java.io.PrintWriter;
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 = "studentServlet", urlPatterns = {"/studentServlet"})
public class studentServlet extends HttpServlet {
    @EJB
    private studentbeanLocal studentbean;
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
            out.println("<!DOCTYPE html>");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>Servlet studentServlet</title>");
            out.println("</head>");
            out.println("<body>");
            int rno=Integer.parseInt(request.getParameter("txtroll"));
                    String n=request.getParameter("txtname");
                    String c=request.getParameter("txtclass");
            int s1=Integer.parseInt(request.getParameter("txts1"));
            int s2=Integer.parseInt(request.getParameter("txts2"));
            int s3=Integer.parseInt(request.getParameter("txts3"));
           String  msg=studentbean.insertmark(rno,n,c,s1,s2,s3);

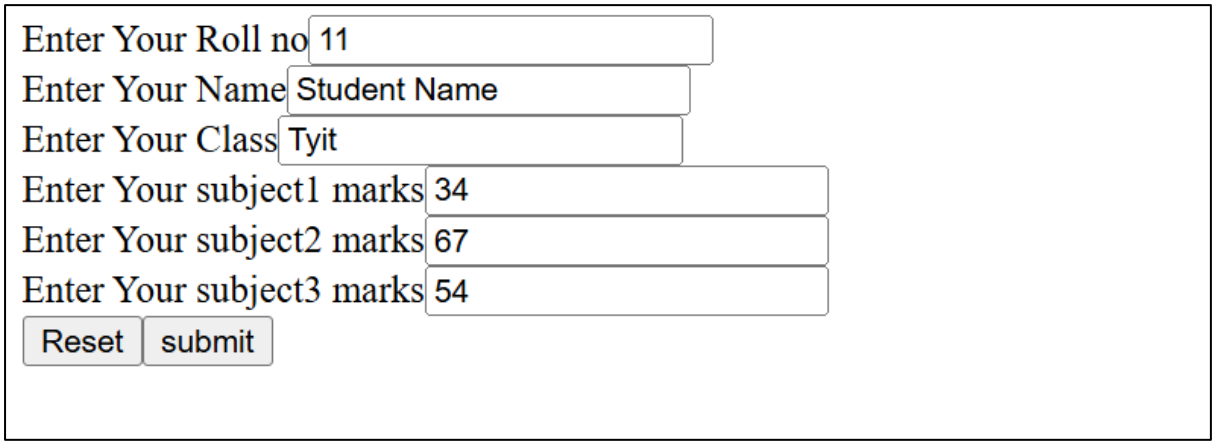
          out.println(msg);

            out.println("</body>");
            out.println("</html>");
        }
    }
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
```

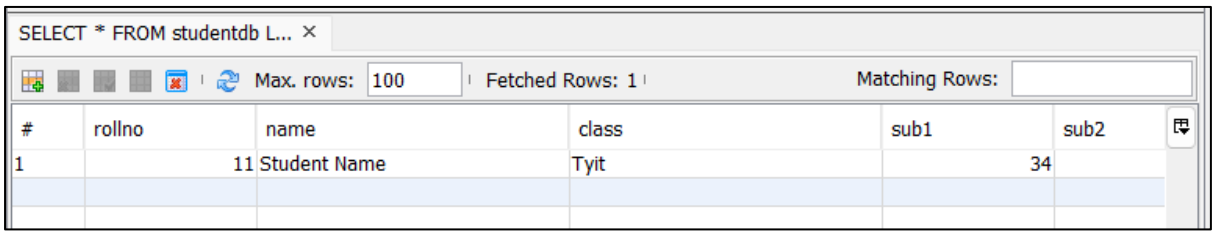
```
    }
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }
    @Override
    public String getServletInfo() {
        return "Short description";
    }

}
```

Output:



Database:



Practical No. 8

Aim: 8a. Develop a simple Inventory Application Using JPA.

MySQL queries:

```
 CREATE TABLE product (
   id INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,
   pname VARCHAR(45) NOT NULL,
   quantity INT(10) UNSIGNED NOT NULL,
   msg TEXT DEFAULT NULL,
   mdate TEXT DEFAULT NULL,
   PRIMARY KEY (id)
);
```

index.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <title>pra8a</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    Product Description
    <form action="ProductView.jsp" method="post">
      Product Name:<input name="pr" maxlength="25" size="50" />
      Price: <input type="text" name="message" />
      <input type="submit" name="btnSubmit" value="Submit" />
    </form>
  </body>
</html>
```

ProductView.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="java.util.*,javax.persistence.*,mypack.Product" %>
<!DOCTYPE html>
<%!
  private EntityManagerFactory entityManagerFactory;
  private EntityManager entityManager;
  private EntityTransaction entityTransaction;
  List<Product> Product;
%>
<%
```

```
  entityManagerFactory = Persistence.createEntityManagerFactory("altpra8PU");
  entityManager = entityManagerFactory.createEntityManager();
  String submit = request.getParameter("btnSubmit");
  if(submit != null && ("Submit").equals(submit)) {
    try {
      String pr = request.getParameter("pr");
      String message = request.getParameter("message");
      String messageDate = new java.util.Date().toString();
      Product gb = new Product();
      gb.setPname(pr);
      gb.setMsg(message);
      gb.setMdate(messageDate);
      entityTransaction = entityManager.getTransaction();
      entityTransaction.begin();
      entityManager.persist(gb);
      entityTransaction.commit();
    } catch (RuntimeException e) {
      if(entityTransaction != null) entityTransaction.rollback();
      throw e;
    }
    response.sendRedirect("ProductView.jsp");
  }

  try {
    Product = entityManager.createQuery("SELECT g from Product g").getResultList();
  } catch (RuntimeException e) {
    throw e;
  }
 entityManager.close();
%>
<html>
  <body>
    <hr />
    <%
      Iterator iterator = Product.iterator();
      while(iterator.hasNext()){
      Product obj = (Product) iterator.next();
      %>
      on <%= obj.getMdate()%><br />
      <b><%= obj.getPname()%></b>
      <%= obj.getMsg()%>
      <br /><br />
      <%
      }
```

```
%>
  </body>
</html>
```

Product.java

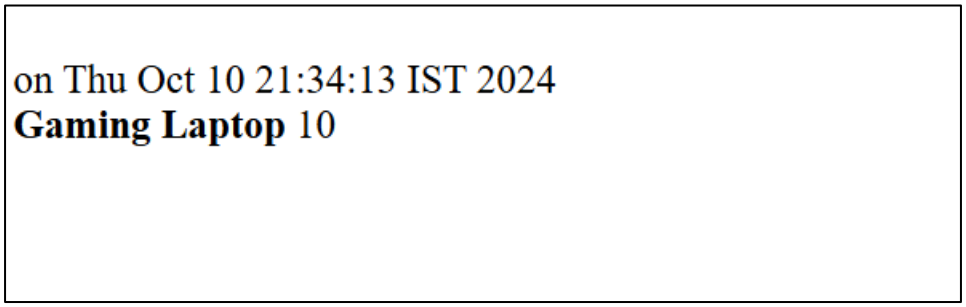
```
package mypack;
import java.io.Serializable;
import javax.persistence.*;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Size;
import javax.xml.bind.annotation.XmlRootElement;
@Entity
@Table(name = "product")
@XmlRootElement
@NamedQueries({
  @NamedQuery(name = "Product.findAll", query = "SELECT p FROM Product p")
  , @NamedQuery(name = "Product.findById", query = "SELECT p FROM Product p
WHERE p.id = :id")
  , @NamedQuery(name = "Product.findByPname", query = "SELECT p FROM Product p
WHERE p.pname = :pname")
  , @NamedQuery(name = "Product.findByQuantity", query = "SELECT p FROM Product
p WHERE p.quantity = :quantity")})
public class Product implements Serializable {
  private static final long serialVersionUID = 1L;
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  @Basic(optional = false)
  @Column(name = "id")
  private Integer id;
  @Basic(optional = false)
  @NotNull
  @Size(min = 1, max = 45)
  @Column(name = "pname")
  private String pname;
  @Basic(optional = false)
  @NotNull
  @Column(name = "quantity")
  private int quantity;
  @Lob
  @Size(max = 65535)
  @Column(name = "msg")
  private String msg;
  @Lob
  @Size(max = 65535)
```

```
@Column(name = "mdate")
private String mdate;
public Product() {
}
public Product(Integer id) {
  this.id = id;
}
public Product(Integer id, String pname, int quantity) {
  this.id = id;
  this.pname = pname;
  this.quantity = quantity;
}
public Integer getId() {
  return id;
}
public void setId(Integer id) {
  this.id = id;
}
public String getPname() {
  return pname;
}
public void setPname(String pname) {
  this.pname = pname;
}
public int getQuantity() {
  return quantity;
}
public void setQuantity(int quantity) {
  this.quantity = quantity;
}

public String getMsg() {
  return msg;
}
public void setMsg(String msg) {
  this.msg = msg;
}
public String getMdate() {
  return mdate;
}
public void setMdate(String mdate) {
  this.mdate = mdate;
}
@Override
```

```
  public int hashCode() {
    int hash = 0;
    hash += (id != null ? id.hashCode() : 0);
    return hash;
  }
  @Override
  public boolean equals(Object object) {
    if (!(object instanceof Product)) {
      return false;
    }
    Product other = (Product) object;
    if ((this.id == null && other.id != null) || (this.id != null && !this.id.equals(other.id))) {
      return false;
    }
    return true;
  }
  @Override
  public String toString() {
    return "mypack.Product[ id=" + id + " ]";
  }}
```

Output:



Aim: 8b. Develop a Guestbook Application Using JPA.

MySQL Query:

```
create table GuestBook(
vno int PRIMARY KEY AUTO_INCREMENT,
vname varchar(50),
msg varchar(100),
mdate varchar(50)
)
```

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>
    Sign To Guest Book
    <form action="GuestbookView.jsp" method="post">
      Visitor Name:<input name="pr" maxlength="25" size="50" />
      Message: <textarea rows="5" cols="36" name="message" ></textarea>
      <input type="submit" name="btnSubmit" value="Submit" />
    </form>
  </body>
</html>
```

GuestbookView.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import="java.util.*,javax.persistence.*,mypack.Guestbook" %>
<!DOCTYPE html>
<%!
  private EntityManagerFactory entityManagerFactory;
  private EntityManager entityManager;
  private EntityTransaction entityTransaction;
  List<Guestbook> Guestbook;
%>
<%
  entityManagerFactory = Persistence.createEntityManagerFactory("pra8bPU");
  entityManager = entityManagerFactory.createEntityManager();
  String submit = request.getParameter("btnSubmit");
```

```
  if(submit != null && ("Submit").equals(submit)) {
    try {
      String guest = request.getParameter("pr");
      String message = request.getParameter("message");
      String messageDate = new java.util.Date().toString();
      Guestbook gb = new Guestbook();
      gb.setVname(guest);
      gb.setMsg(message);
      gb.setMdate(messageDate);
      entityTransaction = entityManager.getTransaction();
      entityTransaction.begin();
      entityManager.persist(gb);
      entityTransaction.commit();
    } catch (RuntimeException e) {
      if(entityTransaction != null) entityTransaction.rollback();
      throw e;
    }
    response.sendRedirect("GuestbookView.jsp");
  }
  try {
    Guestbook = entityManager.createQuery("SELECT g from Guestbook
g").getResultList();
  } catch (RuntimeException e) {
    throw e;
  }
 entityManager.close();
%>
<html>

  <body>
    View the Guest Book <b>Click <a href="index.jsp"> here</a> to sign the
guestbook.</b>
    <hr />
    <%
      Iterator iterator = Guestbook.iterator();
      while(iterator.hasNext()){
      Guestbook obj = (Guestbook) iterator.next();
      %>
      on <%= obj.getMdate()%><br />
      <b><%= obj.getVname()%></b>
      <%= obj.getMsg()%>
      <br /><br />
      <%
      }
```

```
%>
  </body>
</html>
```

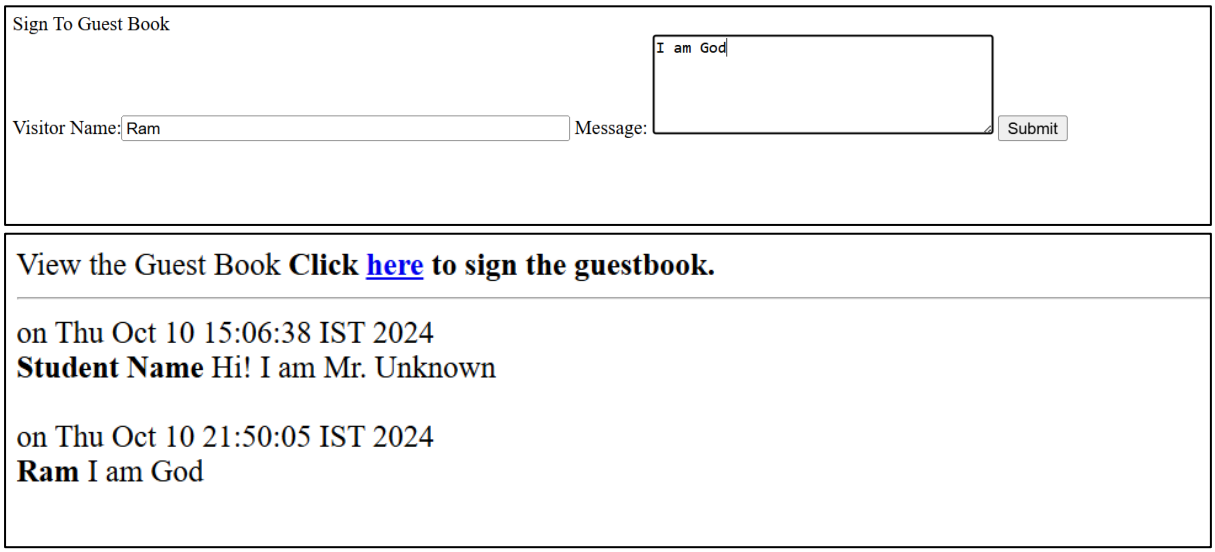
Guestbook.java

```
package mypack;
import java.io.Serializable;
import javax.persistence.*;
import javax.validation.constraints.Size;
import javax.xml.bind.annotation.XmlRootElement;
@Entity
@Table(name = "guestbook")
@XmlRootElement
@NamedQueries({
  @NamedQuery(name = "Guestbook.findAll", query = "SELECT g FROM Guestbook g")
  , @NamedQuery(name = "Guestbook.findByVno", query = "SELECT g FROM
Guestbook g WHERE g.vno = :vno")
  , @NamedQuery(name = "Guestbook.findByVname", query = "SELECT g FROM
Guestbook g WHERE g.vname = :vname")
  , @NamedQuery(name = "Guestbook.findByMsg", query = "SELECT g FROM
Guestbook g WHERE g.msg = :msg")
  , @NamedQuery(name = "Guestbook.findByMdate", query = "SELECT g FROM
Guestbook g WHERE g.mdate = :mdate")})
public class Guestbook implements Serializable {
  private static final long serialVersionUID = 1L;
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  @Basic(optional = false)
  @Column(name = "vno")
  private Integer vno;
  @Size(max = 50)
  @Column(name = "vname")
  private String vname;
  @Size(max = 100)
  @Column(name = "msg")
  private String msg;
  @Size(max = 50)
  @Column(name = "mdate")
  private String mdate;
  public Guestbook() {
  }
  public Guestbook(Integer vno) {
    this.vno = vno;
  }
```

```
    public Integer getVno() {
        return vno;
    }
    public void setVno(Integer vno) {
        this.vno = vno;
    }
    public String getVname() {
        return vname;
    }
    public void setVname(String vname) {
        this.vname = vname;
    }
    public String getMsg() {
        return msg;
    }
    public void setMsg(String msg) {
        this.msg = msg;
    }
    public String getMdate() {
        return mdate;
    }
    public void setMdate(String mdate) {
        this.mdate = mdate;
    }
    @Override
    public int hashCode() {
        int hash = 0;
        hash += (vno != null ? vno.hashCode() : 0);
        return hash;
    }
    @Override
    public boolean equals(Object object) {
        if (!(object instanceof Guestbook)) {
            return false;
        }
        Guestbook other = (Guestbook) object;
        if ((this.vno == null && other.vno != null) || (this.vno != null &&
!this.vno.equals(other.vno))) {
            return false;
        }
        return true;
    }
    @Override
    public String toString() {
```

```
    return "mypack.Guestbook[ vno=" + vno + " ]";
  }}
```

Output:



Aim: 8c. Create simple JPA application to store and retrieve Book details.

MySQL Query:

```
create table Book (BookNo int PRIMARY KEY AUTO_INCREMENT, BookName CHAR(50),
AuthorName CHAR(100), Date CHAR(50));
```

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 style="background-color: pink;">
Book description <br>
<form action="BookView.jsp" method="post">
Book name : <input name="pr" maxlength="25" size="50" />
Author Name: : <input type ="text" name="message" />
<input type="submit" name="btnSubmit" value="Submit" />
</form>
</body>
</html>
```

BookView.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@page import ="java.util.*,javax.persistence.*,mypack.Book" %>
<!DOCTYPE html>
<%!
private EntityManagerFactory entityManagerFactory;
private EntityManager entityManager;
private EntityTransaction  entityTransaction;
List<Book>Book;
%>
<%
entityManagerFactory=Persistence.createEntityManagerFactory("pra8CPU");
entityManager=entityManagerFactory.createEntityManager();
String submit =request.getParameter("btnSubmit");
if(submit !=null && ("submit").equals("submit")){
try
{
  String bn=request.getParameter("pr");
  String an =request.getParameter("message");
```

```
    String messageDate =new java.util.Date().toString();
    Book gb=new Book() ;
    gb.setBookName(bn);
    gb.setAuthorName(an);
    gb.setDate(messageDate);
    entityTransaction=entityManager.getTransaction();
    entityTransaction.begin();
    entityManager.persist(gb);
    entityTransaction.commit();
}catch(RuntimeException e)
{
   if(entityTransaction!=null)entityTransaction.rollback();
   throw e;
}
response.sendRedirect("BookView.jsp");
}
try{
Book =entityManager.createQuery("SELECT g from Book g").getResultList();
}catch(RuntimeException e)
{
   throw e;
}
entityManager.close();
%>
<html>
   <body>
     View the Book <b>Click <a href="index.jsp"> here</a> to sign the Book.</b>
<hr/>
      <%
         Iterator iterator=Book.iterator();
         while (iterator.hasNext())
         {
            Book obj=(Book) iterator.next();
     %>
     On<%=obj.getDate()%><br>
     <b><%=obj.getBookName()%></b>
     <%=obj.getAuthorName()%>
     <br>
     <br>
     <%
        }
     %>
   </body>
</html>
```

Book.java

```
package mypack;
import java.io.Serializable;
import javax.persistence.*;
import javax.validation.constraints.Size;
import javax.xml.bind.annotation.XmlRootElement;
@Entity
@Table(name = "book")
@XmlRootElement
@NamedQueries({
  @NamedQuery(name = "Book.findAll", query = "SELECT b FROM Book b")
  , @NamedQuery(name = "Book.findByBookNo", query = "SELECT b FROM Book b
WHERE b.bookNo = :bookNo")
  , @NamedQuery(name = "Book.findByBookName", query = "SELECT b FROM Book b
WHERE b.bookName = :bookName")
  , @NamedQuery(name = "Book.findByAuthorName", query = "SELECT b FROM Book b
WHERE b.authorName = :authorName")
  , @NamedQuery(name = "Book.findByDate", query = "SELECT b FROM Book b WHERE
b.date = :date")})
public class Book implements Serializable {
  private static final long serialVersionUID = 1L;
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  @Basic(optional = false)
  @Column(name = "BookNo")
  private Integer bookNo;
  @Size(max = 50)
  @Column(name = "BookName")
  private String bookName;
  @Size(max = 100)
  @Column(name = "AuthorName")
  private String authorName;
  @Size(max = 50)
  @Column(name = "Date")
  private String date;
  public Book() {
  }
  public Book(Integer bookNo) {
    this.bookNo = bookNo;
  }
  public Integer getBookNo() {
    return bookNo;
  }
```

```
    public void setBookNo(Integer bookNo) {
        this.bookNo = bookNo;
    }
    public String getBookName() {
        return bookName;
    }
    public void setBookName(String bookName) {
        this.bookName = bookName;
    }
    public String getAuthorName() {
        return authorName;
    }
    public void setAuthorName(String authorName) {
        this.authorName = authorName;
    }
    public String getDate() {
        return date;
    }
    public void setDate(String date) {
        this.date = date;
    }
    @Override
    public int hashCode() {
        int hash = 0;
        hash += (bookNo != null ? bookNo.hashCode() : 0);
        return hash;
    }
    @Override
    public boolean equals(Object object) {
        // TODO: Warning - this method won't work in the case the id fields are not set
        if (!(object instanceof Book)) {
            return false;
        }
        Book other = (Book) object;
        if ((this.bookNo == null && other.bookNo != null) || (this.bookNo != null &&
!this.bookNo.equals(other.bookNo))) {
            return false;
        }
        return true;
    }
    @Override
    public String toString() {
        return "mypack.Book[ bookNo=" + bookNo + " ]";
    }}
```

Output:

Book description
Book name : Enterprise Java(EJ) Author Name: : ganesh Submit

View the Book **Click here to sign the Book.**

OnSat Oct 12 12:43:05 IST 2024
Enterprise Java(EJ) Sandeep

OnSat Oct 12 12:44:44 IST 2024
Enterprise Java(EJ) ganesh

Database:

SELECT * FROM book LIMIT ... ×

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

#	BookNo	BookName	AuthorName	Date
1	1	Enterprise Java(EJ)	Sandeep	Sat Oct 12 12:43:05 IST 2024
2	2	Enterprise Java(EJ)	ganesh	Sat Oct 12 12:44:44 IST 2024

Practical No. 9

Aim: 9a. Develop a JPA Application to demonstrate use of ORM associations.

persistStudent.java

```
package com.jpa.persist;
import com.jpa.student.Student;
import javax.persistence.*;
public class persistStudent {
   public static void main(String args[]){
      EntityManagerFactory emf=Persistence.createEntityManagerFactory("pra9aPU");
      EntityManager em = emf.createEntityManager();
      em.getTransaction().begin();
      Student s1 = new Student();
      s1.setSId(101);
      s1.setSName("sandeep");
      s1.setSAge(20);
      Student s2 = new Student();
      s1.setSId(102);
      s1.setSName("Suraj");
      s1.setSAge(20);
      em.persist(s1);
      em.persist(s2);
      em.getTransaction().commit();
      emf.close();
      em.close();
   }
}
```

Student.java

```
package com.jpa.student;
import java.io.Serializable;
import javax.persistence.*;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Size;
import javax.xml.bind.annotation.XmlRootElement;
@Entity
@Table(name = "student")
@XmlRootElement
@NamedQueries({
   @NamedQuery(name = "Student.findAll", query = "SELECT s FROM Student s")
   , @NamedQuery(name = "Student.findBySId", query = "SELECT s FROM Student s
WHERE s.sId = :sId")
```

```
  , @NamedQuery(name = "Student.findBySName", query = "SELECT s FROM Student s
WHERE s.sName = :sName")
  , @NamedQuery(name = "Student.findBySAge", query = "SELECT s FROM Student s
WHERE s.sAge = :sAge")})
public class Student implements Serializable {
  private static final long serialVersionUID = 1L;
  @Id
  @Basic(optional = false)
  @NotNull
  @Column(name = "s_id")
  private Integer sId;
  @Size(max = 100)
  @Column(name = "s_name")
  private String sName;
  @Column(name = "s_age")
  private Integer sAge;
  public Student() {
  }
  public Student(Integer sId) {
    this.sId = sId;
  }
  public Integer getSId() {
    return sId;
  }
  public void setSId(Integer sId) {
    this.sId = sId;
  }
  public String getSName() {
    return sName;
  }
  public void setSName(String sName) {
    this.sName = sName;
  }
  public Integer getSAge() {
    return sAge;
  }
  public void setSAge(Integer sAge) {
    this.sAge = sAge;
  }
  @Override
  public int hashCode() {
    int hash = 0;
    hash += (sId != null ? sId.hashCode() : 0);
    return hash;
```

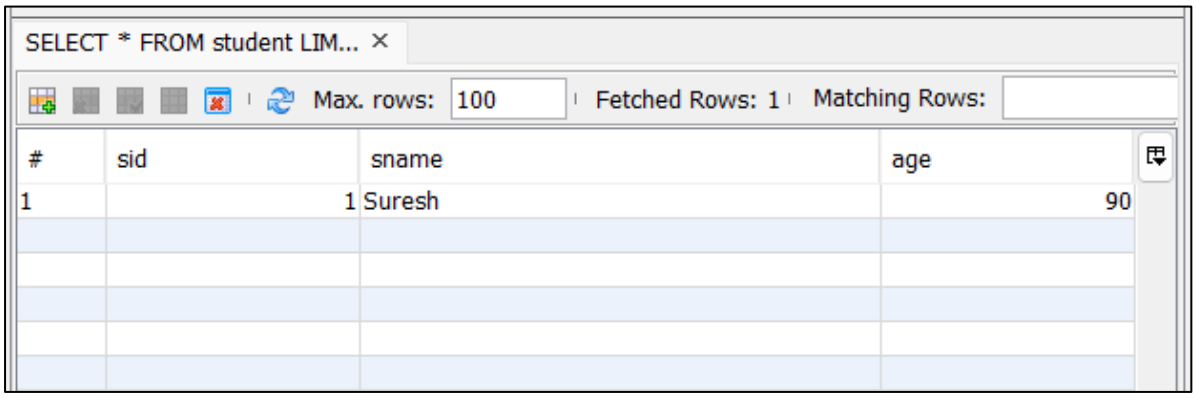
```
    }
    @Override
    public boolean equals(Object object) {
        if (!(object instanceof Student)) {
            return false;
        }
        Student other = (Student) object;
        if ((this.sId == null && other.sId != null) || (this.sId != null &&
!this.sId.equals(other.sId))) {
            return false;
        }
        return true;
    }
    @Override
    public String toString() {
        return "com.jpa.student.Student[ sId=" + sId + " ]";
    }
}
```

StudentEntity.java

```
package com.jpa.student;
import java.io.Serializable;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
@Entity
public class StudentEntity implements Serializable {
    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;
    public Long getId() {
        return id;
    }
    public void setId(Long id) {
        this.id = id;
    }
    @Override
    public int hashCode() {
        int hash = 0;
        hash += (id != null ? id.hashCode() : 0);
        return hash;
    }
```

```
@Override
public boolean equals(Object object) {
    if (!(object instanceof StudentEntity)) {
        return false;
    }
    StudentEntity other = (StudentEntity) object;
    if ((this.id == null && other.id != null) || (this.id != null && !this.id.equals(other.id))) {
        return false;
    }
    return true;
}
@Override
public String toString() {
    return "com.jpa.student.StudentEntity[ id=" + id + " ]";
}}
```

Output:



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

MySQL queries:

```
create table GuestBook(
vno int PRIMARY KEY AUTO_INCREMENT,
vname varchar(50),
msg varchar(100),
mdate varchar(50)
)
```

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>Guest Book</title>
  </head>
  <body>
    <h1>Website Feedback Form for google.con </h1>
<form action="GuestBookView.jsp" >
 Enter Your Name: <input type="text" name="guest" ><br>
 Enter Your Message: <textarea rows="10" cols="50" name="message" ></textarea><br>
<input type="submit" name="btnSubmit" value="Submit" >
</form>
  </body>
</html>
```

GuestBookView.jsp

```
<%@page
import="java.util.Iterator,org.hibernate.Transaction,org.hibernate.service.ServiceRegistr
yBuilder,org.hibernate.cfg.Configuration,org.hibernate.service.ServiceRegistry,java.util.
List,myApp.GuestBook,org.hibernate.SessionFactory"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<%!
  SessionFactory sessionFactory;
  ServiceRegistry serviceRegistry;
  org.hibernate.Session hibernateSession;
  List<GuestBook> guestbook;
%>
```

```
<%
   Configuration configuration = new Configuration();
   configuration.configure();
   serviceRegistry = new
ServiceRegistryBuilder().applySettings(configuration.getProperties()).buildServiceRegis
try();
   sessionFactory = configuration.buildSessionFactory(serviceRegistry);
   hibernateSession = sessionFactory.openSession();
   Transaction transaction = null;
   String submit = request.getParameter("btnSubmit");
   if(submit != null && ("Submit").equals(submit)) {
      GuestBook gb = new GuestBook();
      try {
         transaction = hibernateSession.beginTransaction();
         String guest = request.getParameter("guest");
         String message = request.getParameter("message");
         String messageDate = new java.util.Date().toString();
         gb.setVisitorName(guest);
         gb.setMessage(message);
         gb.setMessageDate(messageDate);

         hibernateSession.save(gb);
         transaction.commit();
      } catch (RuntimeException e) {
         if(transaction != null) transaction.rollback();
         throw e;
      }
      response.sendRedirect("GuestBookView.jsp");
   }

   try {
      hibernateSession.beginTransaction();
      guestbook = hibernateSession.createQuery("from GuestBook").list();
   } catch (RuntimeException e) {
      throw e;
   }
  hibernateSession.close();
%>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>Guest Book</title>
   </head>
   <body>
```

```
<table style="alignment-adjust: central; width: 100%; border: 0px;">
  <tr>
    <td>
      <table style="width: 100%; border: 0px;">
        <tr>
          <td style="width: 60%; vertical-align: middle; text-align: left; padding-
right:0px; padding-left:0px; padding-bottom:0px; font:24px/30px Georgia; width:228px;
color:#786e4e; padding-top:0px; height:37px;">
            Thanks for your feedback
          </td>
          <td style="vertical-align: bottom; text-align: right; font:12px/16px Georgia,
serif; color:#786e4e;">
            <b>Click <a href="index.jsp"> here</a> to sign for Feedback.</b>
          </td>
        </tr>
      </table>
    </td>
  </tr>
  <tr style="text-align: left; vertical-align: top;">
    <td style="height: 20px;"><hr /></td>
  </tr>
  <tr>
    <td>
      <table style="text-align: left; width: 100%; border: 0px;">
        <%
          Iterator iterator = guestbook.iterator();
          while (iterator.hasNext()) {
            GuestBook objGb = (GuestBook) iterator.next();
        %>
        <tr>
          <td style="font:12px/16px Georgia; color:#786e4e;">
            On <%=objGb.getMessageDate()%>,<br />
            <b><%=objGb.getVisitorName()%>:</b>
            <%=objGb.getMessage()%>
            <br /><br />
          </td>
        </tr>
        <%
          }
        %>
      </table>
    </td>
  </tr>
</table>
```

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

hibernate.cfg.xml

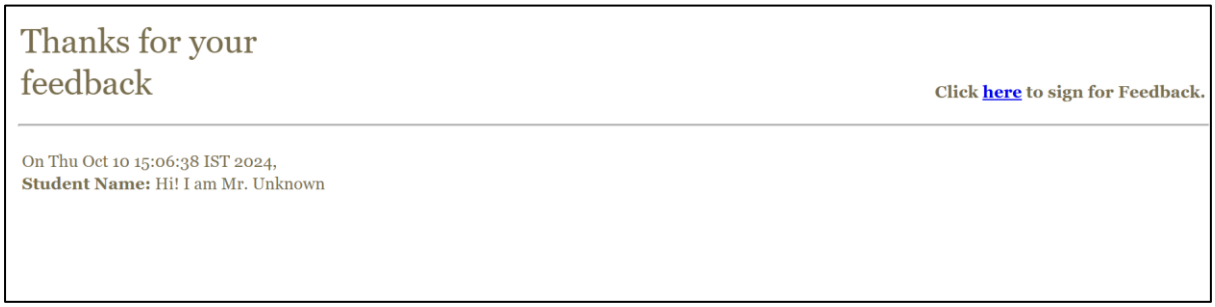
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD
3.0//EN" "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
 <session-factory>
  <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
  <property
name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
  <property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/mysql</property>
  <property name="hibernate.connection.username">root</property>
  <property name="hibernate.connection.password">root</property>
  <mapping class="myApp.GuestBook"/>
 </session-factory>
</hibernate-configuration>
```

GuestBook.java

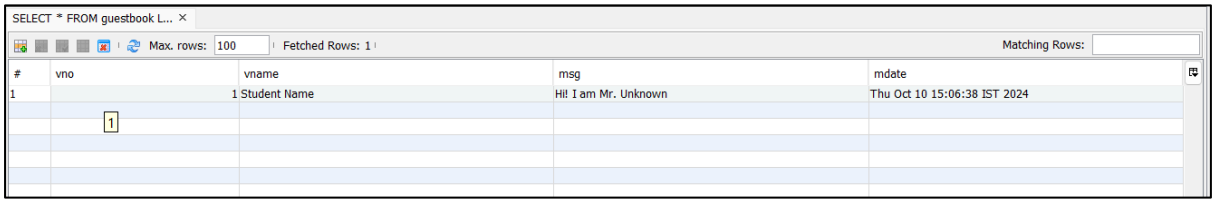
```
package myApp;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.Table
@Entity
@Table(name="GuestBook")
public class GuestBook implements java.io.Serializable {
  @Id
  @GeneratedValue
  @Column(name="vno")
  private Integer visitorNo;
  @Column(name="vname")
  private String visitorName;
  @Column(name="msg")
  private String message;
  @Column(name="mdate")
  private String messageDate;
  public GuestBook() {
  }
  public GuestBook(String visitorName, String message, String messageDate) {
    this.visitorName = visitorName;
```

```
        this.message = message;
        this.messageDate = messageDate;
    }
    public Integer getVisitorNo() {
        return visitorNo;
    }
    public void setVisitorNo(Integer visitorNo) {
        this.visitorNo = visitorNo;
    }
    public String getVisitorName() {
        return visitorName;
    }
    public void setVisitorName(String visitorName) {
        this.visitorName = visitorName;
    }
    public String getMessage() {
        return message;
    }
    public void setMessage(String message) {
        this.message = message;
    }
    public String getMessageDate() {
        return messageDate;
    }
    public void setMessageDate(String messageDate) {
        this.messageDate = messageDate;
    }
}
```

Output:



Database:



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

MySQL queries:

```
create table emptable(
eno int PRIMARY KEY AUTO_INCREMENT,
ename varchar(50),
eadd varchar(100),
edate varchar(50)
)
```

Index.html

```
<html>
   <head>
      <title>TODO supply a title</title>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
   </head>
   <body>
<form action="fb.jsp">
Employee   name : <input type ="text" name="name" /> <br>
 Address  : <textarea cols="50" rows="10" name="address"> </textarea><br>
 <input type="submit" name="btnSubmit" value="Submit" />
</form>
   </body>
</html>
```

fb.jsp

```
<%@page import="org.hibernate.*,org.hibernate.cfg.*,mypack.*" %>
<%!SessionFactory sf;
org.hibernate.Session hibSession;
%>
<%
sf=new Configuration().configure().buildSessionFactory();
hibSession=sf.openSession();
Transaction tx=null;
empbook gb=new empbook();
try{
  tx= hibSession.beginTransaction();
   String empname =request.getParameter("name");
    String empaddress  =request.getParameter("address");
```

```
    String nowtime  =""+new java.util.Date();
    gb.setEmpName(empname);
    gb.seteAddress(empaddress);
    gb.seteDate(nowtime);
    hibSession.save(gb);
    tx.commit();
    out.println("thanks for feedback ");
}
catch(Exception e)
{out.println(e);}
hibSession.close();
%>
```

hibernate.cfg.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD
3.0//EN" "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
 <session-factory>
  <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
  <property
name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
  <property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/mysql?zeroDateTimeBe
havior=convertToNull</property>
  <property name="hibernate.connection.username">root</property>
  <property name="hibernate.connection.password">root</property>
      <mapping class ="mypack.empbook"/>
 </session-factory>
</hibernate-configuration>
```

empbook.java

```
package mypack;
import javax.persistence.*;
@Entity
@Table(name="emptable")
public class empbook implements java.io.Serializable {
  @Id
  @GeneratedValue
  @Column(name="eno")
  private Integer empNo;
  @Column(name="ename")
  private String empName;
  @Column(name="eadd")
```

```
private String eAddress;
@Column(name="edate")
private String eDate;
public empbook(){}
public Integer getEmpNo() {
  return empNo;
}
public void setEmpNo(Integer empNo) {
  this.empNo = empNo;
}
public String getEmpName() {
  return empName;
}
public void setEmpName(String empName) {
  this.empName = empName;

public String geteAddress() {
  return eAddress;
}
public void seteAddress(String eAddress) {
  this.eAddress = eAddress;
}
public String geteDate() {
  return eDate;
}
public void seteDate(String eDate) {
  this.eDate = eDate;
}}
```

Output:

Employee name : Ganesh

Thane

Address :

Submit

thanks for feedback

Database:

SELECT * FROM emptable LI... ×

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

#	eno	ename	eadd	edate
1	1	Sandeep Vishwakarma	thane	Thu Oct 03 17:00:41 IST 2024
2	2	Ganesh	Thane	Thu Oct 10 15:17:25 IST 2024

Aim: 10a. Develop an application to demonstrate Hibernate One- To -One Mapping Using Annotation.

MySQL queries:

```
CREATE TABLE empdb (
   id INT(11) NOT NULL AUTO_INCREMENT,
   firstName CHAR(20) DEFAULT NULL,
   lastName CHAR(20) DEFAULT NULL,
   PRIMARY KEY (id)
);
```

hibernate.cg.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN" "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
 <session-factory>
   <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
   <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
   <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/mysql?zeroDateTimeBehavior=convertToNull</property>
   <property name="hibernate.connection.username">root</property>
   <property name="hibernate.connection.password">root</property>
   <mapping resource="employee.hbm.xml"/>
 </session-factory>
</hibernate-configuration>
```

employee.hbm.xml

```
<!DOCTYPE hibernate-mapping PUBLIC
 "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
 "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
   <class name="Employee" table="empdb">
      <id name="id">
         <generator class="assigned" />
      </id>
      <property name="firstName"/>
      <property name="lastName"/>
   </class>
</hibernate-mapping>
```

Employee.java

```
public class Employee {
   private int id;
   private String firstName;
   private String lastName;
   public int getId() {
      return id;
   }
   public void setId(int id) {
      this.id = id;
   }
   public String getFirstName() {
      return firstName;
   }
   public void setFirstName(String firstName) {
      this.firstName = firstName;

   public String getLastName() {
      return lastName;
   }
   public void setLastName(String lastName) {
      this.lastName = lastName;
   }}
```

StoreData.java

```
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;
import org.hibernate.cfg.Configuration;
public class StoreData {
   public static void main(String[] args) {
       Configuration cfg = new Configuration();
      cfg.configure("hibernate.cfg.xml");
      StandardServiceRegistryBuilder ssr = new
StandardServiceRegistryBuilder().applySettings(cfg.getProperties());
      SessionFactory factory = cfg.buildSessionFactory(ssr.build());
      Session session = factory.openSession();
      Transaction t = session.beginTransaction();
      Employee e1 = new Employee();
      e1.setId(1);
      e1.setFirstName("Ram");
      e1.setLastName("Patel");
      session.save(e1);
```

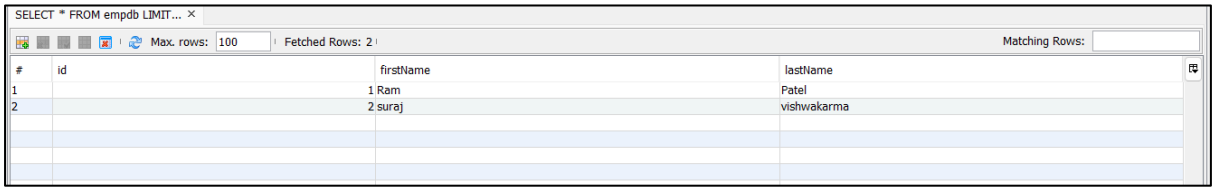
```
t.commit();
System.out.println("Successfully saved");
session.close();
factory.close();}}
```

Output:

Java Main file run (StoreData.java):

```
INFO: HHH000399: Using default transaction strategy (direct JDBC transactions)
Oct 10, 2024 3:33:16 PM org.hibernate.hql.internal.ast.ASTQueryTranslatorFactory
INFO: HHH000397: Using ASTQueryTranslatorFactory
Oct 10, 2024 3:33:17 PM org.hibernate.validator.internal.util.Version <clinit>
INFO: HV000001: Hibernate Validator 5.1.2.Final
Successfully saved
```

Database:



Aim: 10b. Develop Hibernate application to enter and retrieve course details with ORM Mapping.

MySQL queries:

```
CREATE TABLE course (
   id INT(11) NOT NULL AUTO_INCREMENT,
   Cname CHAR(25) DEFAULT NULL,
   fees INT(11) DEFAULT NULL,
   PRIMARY KEY (id)
);
```

hibernate.cfg.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN" "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
 <session-factory>
   <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
   <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
   <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/mysql?zeroDateTimeBehavior=convertToNull</property>
   <property name="hibernate.connection.username">root</property>
   <property name="hibernate.connection.password">root</property>
   <mapping resource="employee.hbm.xml"/>
 </session-factory>
</hibernate-configuration>
```

employee.hbm.xml

```
<!DOCTYPE hibernate-mapping PUBLIC
 "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
 "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
    <class name="Course" table="course">
   <meta attribute="class-description">
     This class contains the course detail.
   </meta>
   <id name="id" type="int" column="id">
     <generator class="native"/>
   </id>
   <property name="CName" column="Cname" type="string"/>
   <property name="fees" column="fees" type="int"/>
 </class>
</hibernate-mapping>
```

Course.java

```
public class Course {
    private int id;
   private String CName;
   private int fees;
   public Course() {}
   public Course(String Cname,int fees)
      {
      this.CName=Cname;
      this.fees=fees;
      }
   public int getId() {
      return id;
   }
   public void setId(int id) {
      this.id = id;
   }
   public String getCName() {
      return CName;
   }
   public void setCName(String CName) {
      this.CName = CName;
   }
   public int getFees() {
      return fees;
   }
   public void setFees(int fees) {
      this.fees = fees;
   }
}
```

ManageCourse.java

```
import org.hibernate.HibernateException;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
import java.util.Iterator;
import java.util.List;
public class ManageCourse {
   private static SessionFactory factory;
   public static void main(String[] args) {
      try {
         factory = new Configuration().configure().buildSessionFactory();
```

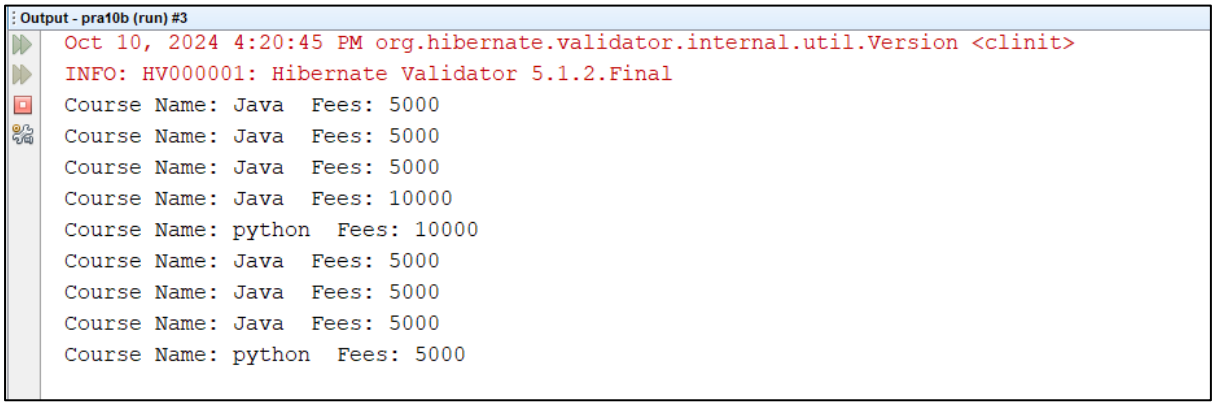
```
    } catch (Throwable ex) {
      System.err.println("Failed to create sessionFactory object." + ex);
      throw new ExceptionInInitializerError(ex);
    }

    ManageCourse mc = new ManageCourse();
    Integer courseID1 = mc.addCourse("Java", 10000);
    Integer courseID2 = mc.addCourse("Net", 10000);
    mc.listCourses();
    mc.updateCourse(courseID1, 5000);
    mc.deleteCourse(courseID2);
    mc.listCourses();
  }
  public Integer addCourse(String CName, int fees) {
    Session session = factory.openSession();
    org.hibernate.Transaction tx = null;
    Integer courseID = null;
    try {
      tx = session.beginTransaction();
      Course course = new Course(CName, fees);
      courseID = (Integer) session.save(course);
      tx.commit();
    } catch (HibernateException e) {
      if (tx != null) tx.rollback();
      e.printStackTrace();
    } finally {
      session.close();
    }
    return courseID;
  }
  public void listCourses() {
    Session session = factory.openSession();
    org.hibernate.Transaction tx = null;
    try {
      tx = session.beginTransaction();
      List courses = session.createQuery("FROM Course").list();
      for (Iterator iterator = courses.iterator(); iterator.hasNext(); ) {
        Course course = (Course) iterator.next();
        System.out.print("Course Name: " + course.getCName());
        System.out.println("  Fees: " + course.getFees());
      }
      tx.commit();
    } catch (HibernateException e) {
      if (tx != null) tx.rollback();
```

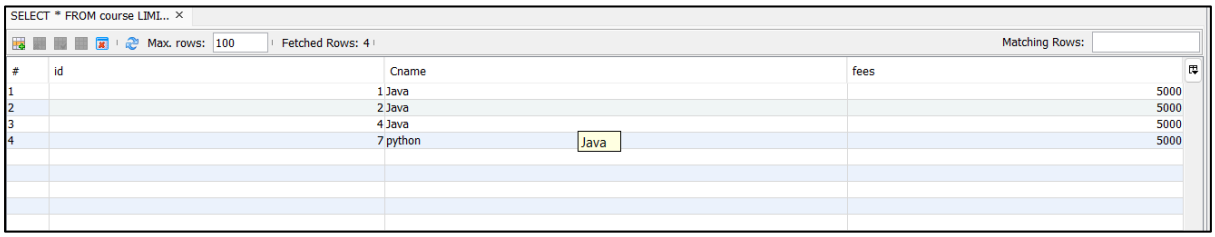
```
            e.printStackTrace();
        } finally {
            session.close();
        }
    }
    public void updateCourse(Integer courseID, int fees) {
        Session session = factory.openSession();
        org.hibernate.Transaction tx = null;
        try {
            tx = session.beginTransaction();
            Course course = (Course) session.get(Course.class, courseID);
            course.setFees(fees);
            session.update(course);
            tx.commit();
        } catch (HibernateException e) {
            if (tx != null) tx.rollback();
            e.printStackTrace();
        } finally {
            session.close();
        }
    }
    public void deleteCourse(Integer courseID) {
        Session session = factory.openSession();
        org.hibernate.Transaction tx = null;
        try {
            tx = session.beginTransaction();
            Course course = (Course) session.get(Course.class, courseID);
            session.delete(course);
            tx.commit();
        } catch (HibernateException e) {
            if (tx != null) tx.rollback();
            e.printStackTrace();
        } finally {
            session.close();
        }
    }
}
```

Output:

Java Main file run (ManageCourse.java):



Database:



Aim: 10c. Develop a five page web application site using any two or three Java EE Technologies.

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>
    <style>
      div.ex{
        text-align: right;
        width:300px;
        padding:10px;
        border: 5px solid grey;
        margin: 0px;
      }
    </style>
  </head>
    <body>
    <h1>Registration Form</h1>
   <div class="ex">
      <form action="RegistrationController" method="post">
        <table style="with: 50%">
          <tr><td>Full Name</td>
            <td><input type="text" name="fullname" /></td></tr>
          <tr><td>Address</td>
            <td><input type="text" name="address" /></td></tr>
          <tr><td>Age</td>
            <td><input type="text" name="age" /></td></tr>
          <tr><td>Qualification</td>
            <td><input type="text" name="qual" /></td></tr>
        <tr><td>Percentage</td>
            <td><input type="text" name="percent" /></td></tr>
          <tr><td>Year Passed</td>
            <td><input type="text" name="yop" /></td></tr>
        </table>
        <input type="submit" value="register" />
      </form>
    </div>
</body>
```

```
</html>
```

home.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>
    <style>
      table#nat{
        width: 50%;
        background-color: #c48ec5;
      }
    </style>
  </head>
  <body>
<% String name = request.getParameter ("fullname");
    String Addr = request.getParameter("address");
    String age = request.getParameter ("age");
    String Qual = request.getParameter ("qual");
    String Persent = request.getParameter ("percent");
    String Year = request.getParameter("yop"); %>
<table id ="nat">
<tr>
    <td>Full Name</td>
    <td><%= name %></td> </tr>
<tr>
    <td>Address</td>
    <td><%= Addr %></td> </tr>
<tr>
    <td>Age</td>
    <td><%= age %></td> </tr>
<tr>
    <td>Qualification</td>
    <td><%= Qual %></td> </tr>
<tr>
    <td>Percentage</td>
    <td><%= Persent %></td> </tr>
<tr>
    <td>Year of Passout</td>
    <td><%= Year %></td>
</tr>
</table>
```

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

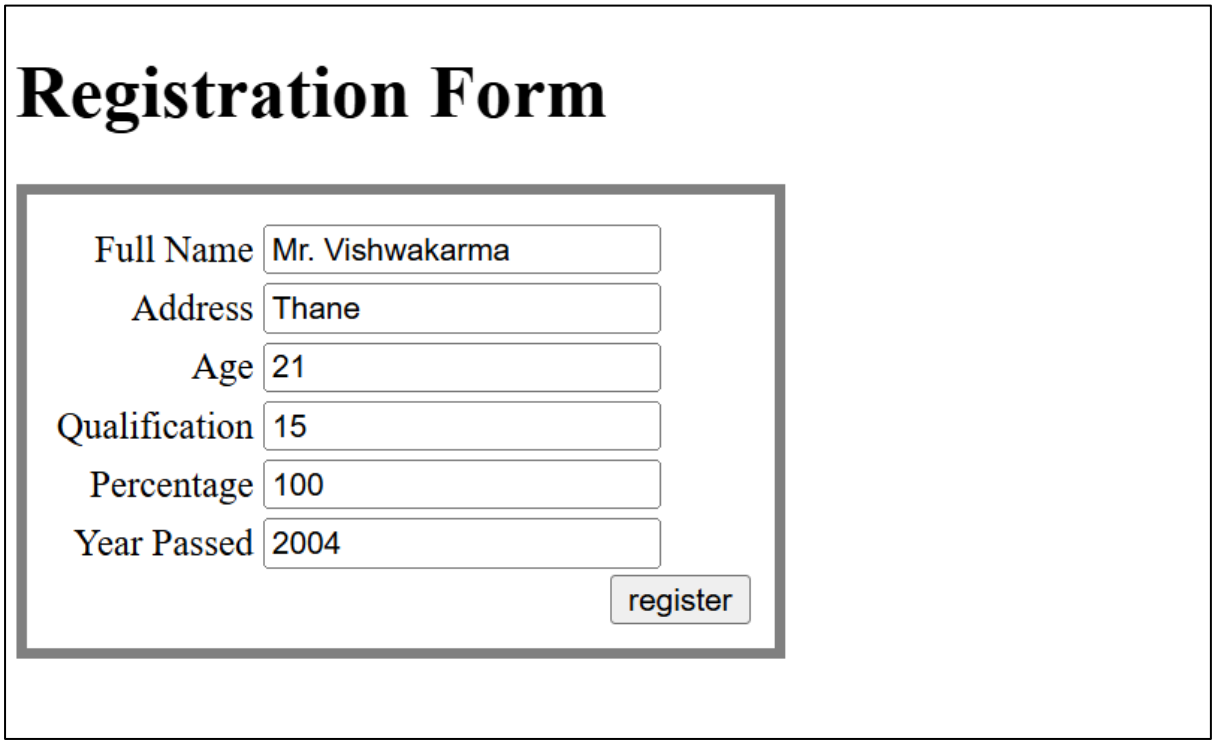
RegistrationController.java

```
package mypack;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.RequestDispatcher;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet(name = "RegistrationController", urlPatterns = {"/RegistrationController"})
public class RegistrationController extends HttpServlet {
   protected void doPost(HttpServletRequest request, HttpServletResponse response)
         throws ServletException, IOException {
      response.setContentType("text/html;charset=UTF-8");
         PrintWriter out = response.getWriter();
         String name = request.getParameter("fullname");
         String Addr = request.getParameter("address");
      String age = request.getParameter ("age");
      String Qual = request.getParameter ("qual");
      String Persent = request.getParameter ("percent");
      String Year = request.getParameter("yop");
      if(name.isEmpty()||
Addr.isEmpty()||Qual.isEmpty()||Persent.isEmpty()||Year.isEmpty())
      {
         RequestDispatcher rd = request.getRequestDispatcher("index.jsp");
         out.println("<font color=red>Please fill all the fields</font>");
         rd.include(request, response);
      }
      else{
          RequestDispatcher rd = request.getRequestDispatcher("home.jsp");
          rd.forward(request, response);
      }

   }

}
```

Output:



Full Name	Mr. Vishwakarma
Address	Thane
Age	21
Qualification	15
Percentage	100
Year of Passout	2004