

1A. Create a simple calculator application using servlet.**Steps to be followed:-**

1. Open NetBeans IDE
2. Go to File->New Project
3. Select Java Web->Web Application
4. Provide Project Name
5. Select Server(If server is not added Click on Add->Browse location for Server). Here we are using Glassfish Server.
6. Select Java Version EE 7
7. No need of selecting Framework.
8. Finish the Creation of Project
9. Go to Web Pages->Open index.html
- 10.Right Click on Source Packages->New->Servlet (CalculatorServlet.java)
- 11.Give name to your Servlet File.
- 12.Provide package name if you want to create.
- 13.Click Next->Check Add Deployment Descriptor->Click Finish
- 14.Go to Services->Servers->Start Glassfish Server.
- 15.If Server is not added then Right Click on Server->Add Server->Select Glassfish->Browse for location where Glassfish Server is placed->Finish

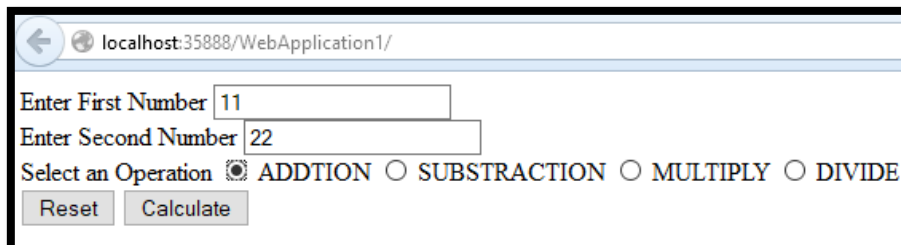
index.html

```
<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="+">
ADDITION <input type="radio" name="opr" value="-">
SUBTRACTION <input type="radio" name="opr" value="*">
MULTIPLY <input type="radio" name="opr" value="/">
DIVIDE <br><input type="reset">
<input type="submit" value="Calculate" >
</form></body></html>
```

CalculatorServlet.java

```
packagemypack;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
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;
out.println("<h1> Result = "+result);
if(opr.equals("-"))
result=n1-n2;
if(opr.equals("/"))
result=n1/n2;
out.println("</body></html>");
} }
```

Output:-



localhost:35888/WebApplication1/

Enter First Number 11

Enter Second Number 22

Select an Operation ADDITION SUBSTRUCTION MULTIPLY DIVIDE

Reset Calculate



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"

Follow the same steps for creating project as defined in Practical 1A.

index.html

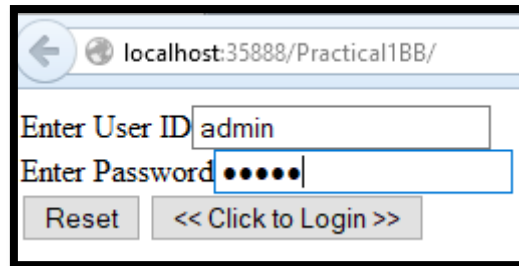
```
<html><head><title>Login Form</title></head>
<form action="LoginServlet" >
Enter User ID<input type="text" name="txtId"><br>
Enter Password<input type="password" name="txtPass"><br>
<input type="reset">
<input type="submit" value=" Click to Login "></form></html>
```

LoginServlet.java

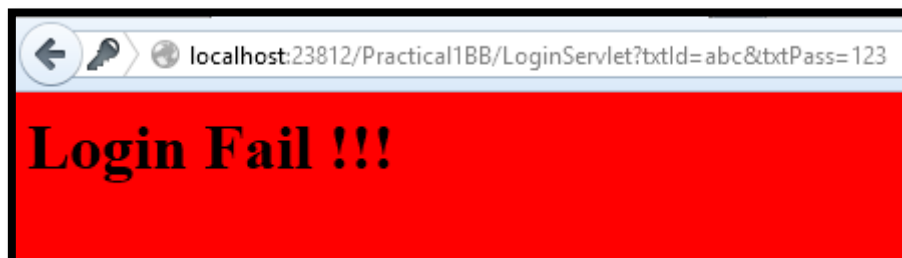
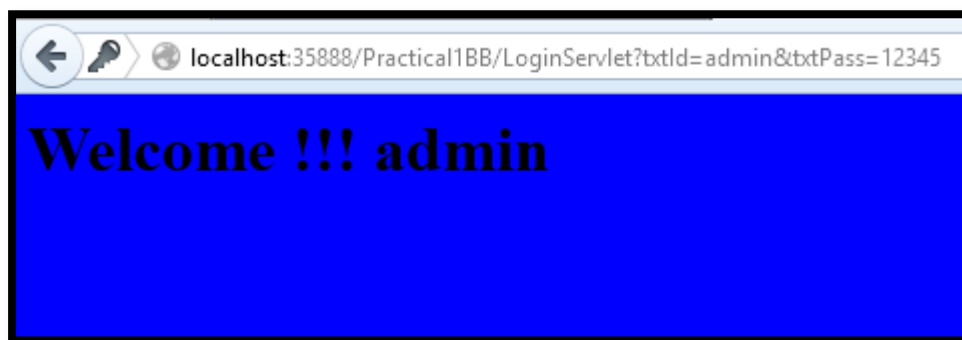
```
packagemypack;
import java.io.*;
import javax.servlet.ServletException;
import javax.servlet.http.*;
public class LoginServlet extends HttpServlet
{
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException
{
response.setContentType("text/html;charset=UTF-8");
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:-



A screenshot of a web browser window. The address bar shows "localhost:35888/Practical1BB/". The page contains a login form with two input fields: "Enter User ID" with the text "admin" and "Enter Password" with six dots. Below the fields are two buttons: "Reset" and "<< Click to Login >>".



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.

Follow the same steps for creating project as defined in Practical 1A.

1. For Creating Database, Open MySQL->Provide Password
2. Follow the steps:

```
create database sr;
use sr;
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;
```

3. To Add JAR/Folder

Go to Libraries->Right Click->Select Add/Folder->Browse for MySql Connector.

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/sr", "root","admin123");
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);}
}
}

```

Welcome to Registration page

localhost:8080/Practical_1C/RegisterServlet?txtUid=Jimmy&

Enter User Name Jimmy
Enter Password ****
Enter Email abcd@xyz.com
Enter Country Japan
Reset REGISTER

1 Inserted Succesfullyyyyy

index.html RegisterServlet.java SQL 1 [jdbc:mysql://localhost:33...]

Connection: jdbc:mysql://localhost:3306/mysql?zeroDateTimeBehavior=convertToNull [ro...]

SELECT * FROM logindb.`us...`

Max. rows: 100 | Fetched Rows: 2 |

#	username	password	email	country
1	admin	admin	admin@admin.com	India
2	Jimmy	abcd	abcd@xyz.com	Japan

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.

Follow the same steps for creating project as defined in Practical 1A.

Files Required:

1. index.html
2. LoginServlet.java
3. WelcomeServlet.java

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.*;
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>");
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>");
}}
```

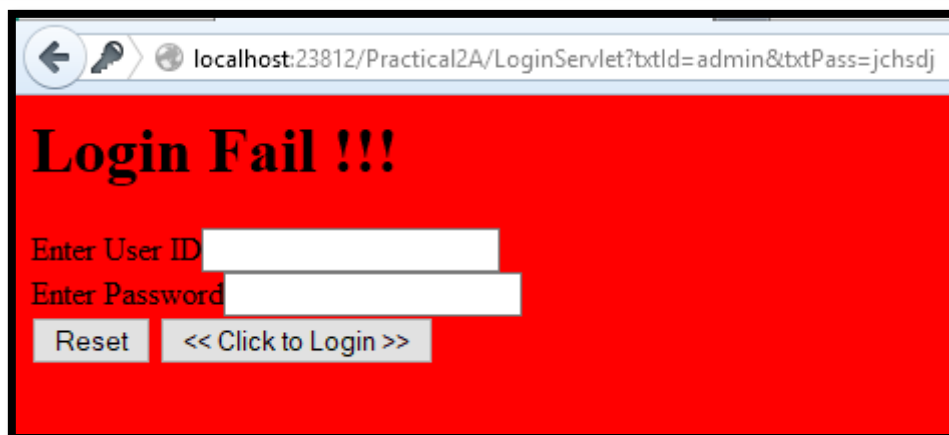
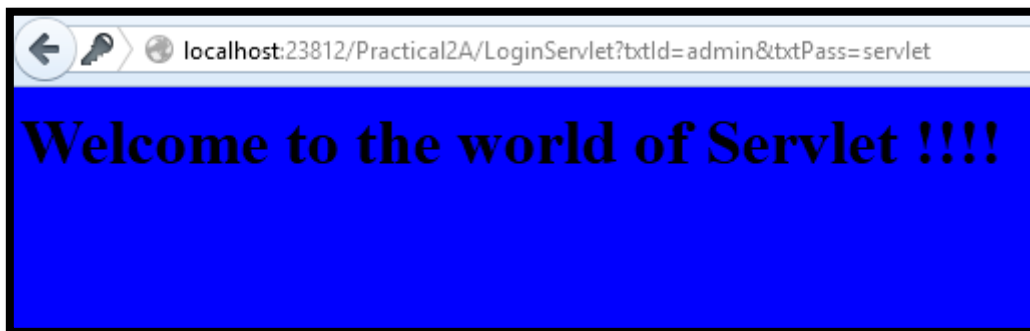
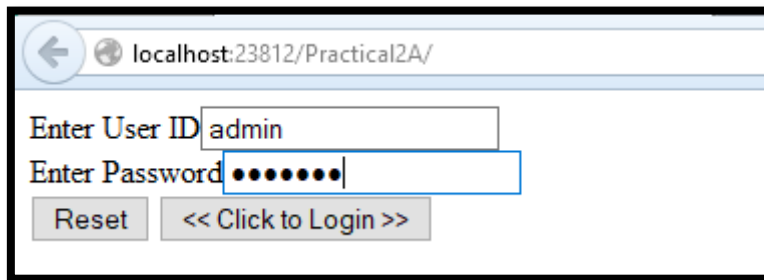
WelcomeServlet.java

```
package mypack;
import java.io.*;
import javax.servlet.*;
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>");
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>");
}}
```

Output:-



2B. Create a Servlet that uses Cookies to store the number of times a user has visited Servlet.

Follow the same steps for creating project as defined in Practical 1A.

Files Required:

1. index.html
2. Create five Servlet files(Page1.java, Page2.java, Page3.java, Page4.java, Page5.java)

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.*;
import javax.servlet.*;
import javax.servlet.http.*;
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);
```

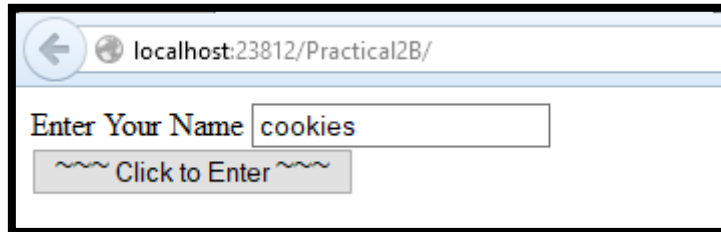
```
Cookie ck2 = new Cookie("visit","1");
response.addCookie(ck1); response.addCookie(ck2);
out.println("<h1><a href=Page2 >Click to visit Page 2 </a></h1>");
out.println("</body>");
out.println("</html>");
}
}
```

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>");
}}
```

Repeat the code from Page2.java for Page3.java, Page4.java and Page5.java with relevant changes.



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.

Follow the same steps for creating project as defined in Practical 1A.

Files Required:

1. index.html
2. Create six Servlet files(Page1.java, Page2.java, Page3.java, Page4.java, Page5.java,LogServlet.java)

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

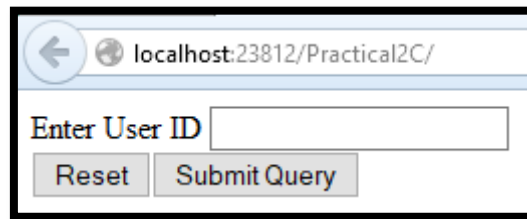
```
package mypack;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class Page2 extends HttpServlet
{
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException
{
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
out.println("<html><head><title>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.setAttribute("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>");
}}
```

Repeat the code from Page2.java in Page3.java and Page4.java with relevant changes.

LogoutServlet.java

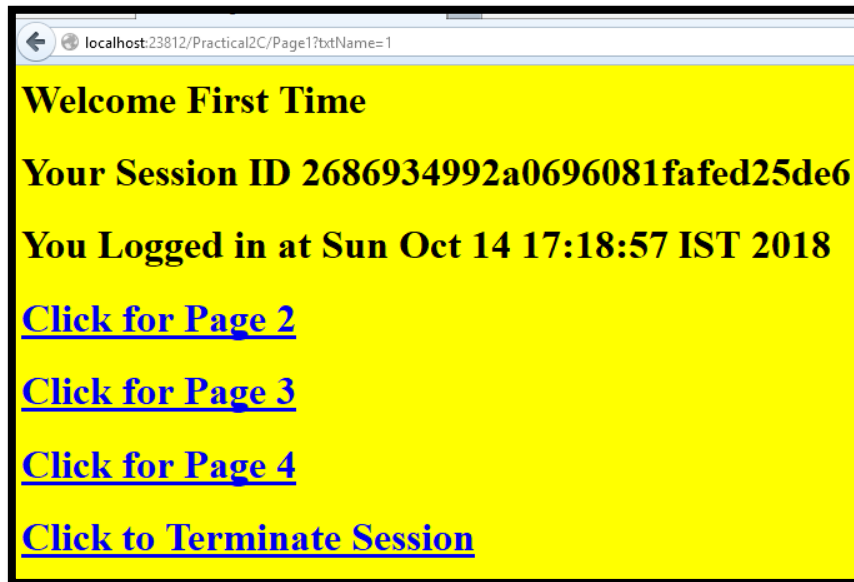
```
package mypack;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class LogoutServlet extends HttpServlet
{
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException
{
response.setContentType("text/html;charset=UTF-8");
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>");
}}
```



localhost:23812/Practical2C/

Enter User ID

Reset Submit Query



localhost:23812/Practical2C/Page1?btName=1

Welcome First Time

Your Session ID 2686934992a0696081fafed25de6

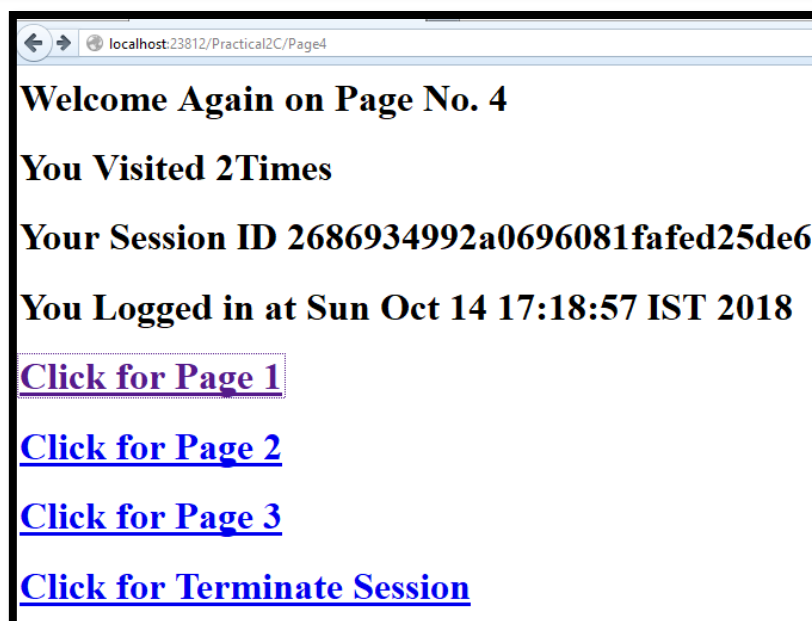
You Logged in at Sun Oct 14 17:18:57 IST 2018

[Click for Page 2](#)

[Click for Page 3](#)

[Click for Page 4](#)

[Click to Terminate Session](#)



localhost:23812/Practical2C/Page4

Welcome Again on Page No. 4

You Visited 2Times

Your Session ID 2686934992a0696081fafed25de6

You Logged in at Sun Oct 14 17:18:57 IST 2018

[Click for Page 1](#)

[Click for Page 2](#)

[Click for Page 3](#)

[Click for Terminate Session](#)

3A. Create a Servlet application to upload a file.

Follow the same steps for creating project as defined in Practical 1A.

Required Files:

1. index.html
2. FileUploadServlet.java

index.html

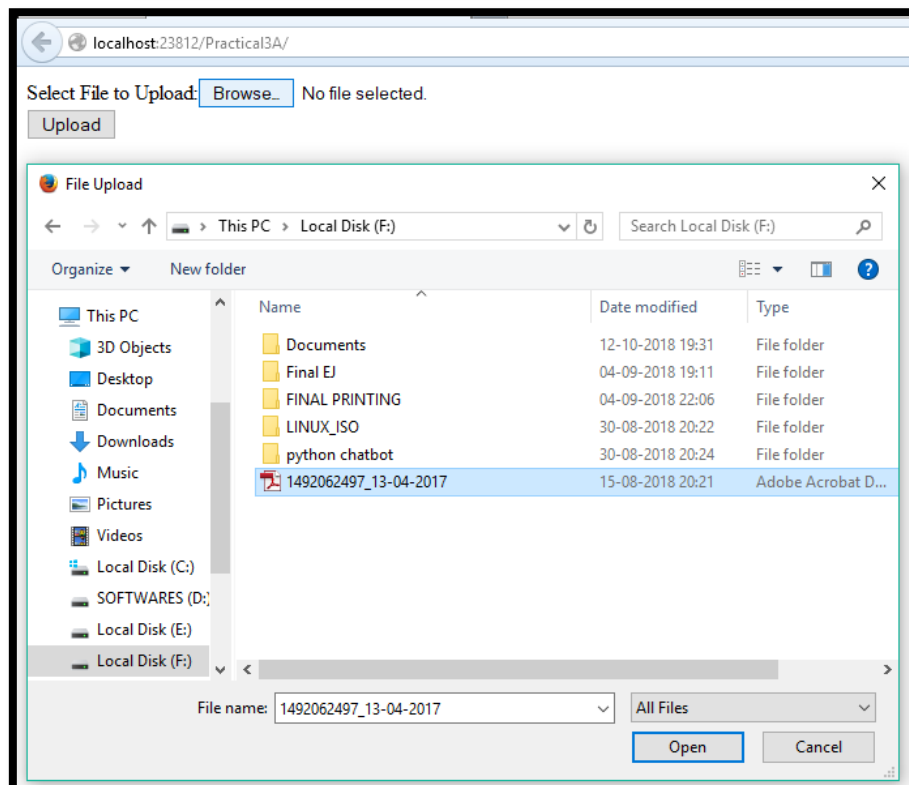
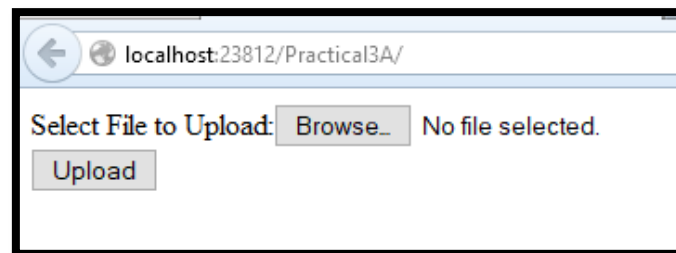
```
<html>
<body>
<form action="FileUploadServlet" method="post" enctype="multipart/form-
data">
Select File to Upload:<input type="file" name="file" id="file">
Destination <input type="text" value="/tmp" name="destination"> <br>
<input type="submit" value="Upload file" name="upload" id="upload">
</form>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

FileUploadServlet.java

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

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

Output:-



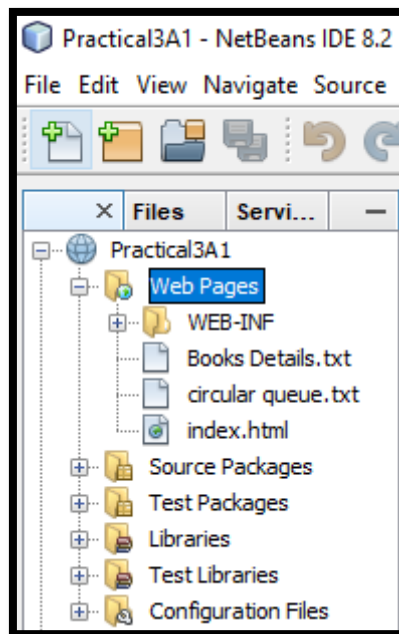
3A1. Create a Servlet application to download a file.

Follow the same steps for creating project as defined in Practical 1A.

Required Files:

1. index.html
2. FileDownloadServlet.java

Add one or more files in Web Pages Folder as shown below:-

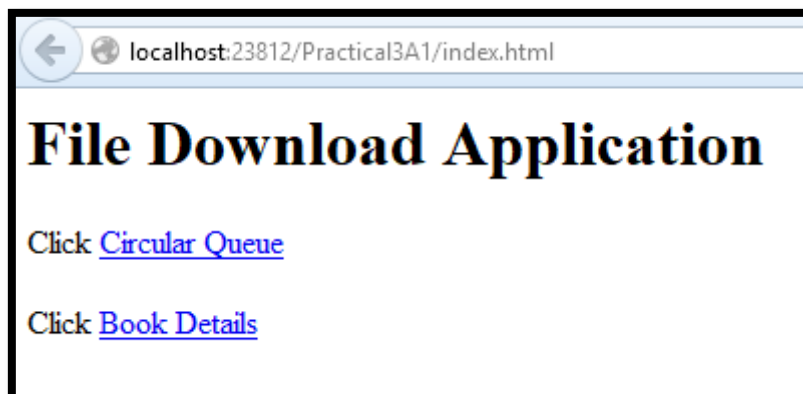
**index.html**

```
<html><head>
<title>File Download Page</title>
</head><body>
<h1>File Download Application</h1>
Click <a href="DownloadServlet?filename=Book Details.txt">
Sample Chapter</a> <br/><br/>
Click <a href="DownloadServlet?filename=circular queue.txt">Table Of
Contents</a> </body></html>
```

DownloadServlet.java

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

```
public class DownloadServlet extends HttpServlet
{
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException
{
response.setContentType("APPLICATION/OCTET-STREAM");
String filename = request.getParameter("filename");
ServletContext context = getServletContext();
InputStream is = context.getResourceAsStream("/" + filename);
//ServletOutputStream 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:-

3B. Develop Simple Servlet Question Answer Application using Database.

Follow the same steps for creating project as defined in Practical 1A.

Required Files:

1. index.html
2. QuizServlet.java
3. ShowServlet.java

Create database using MySql as follows:

```
create database qadb;
```

```
use qadb;
```

```
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??','New  
Delhi','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

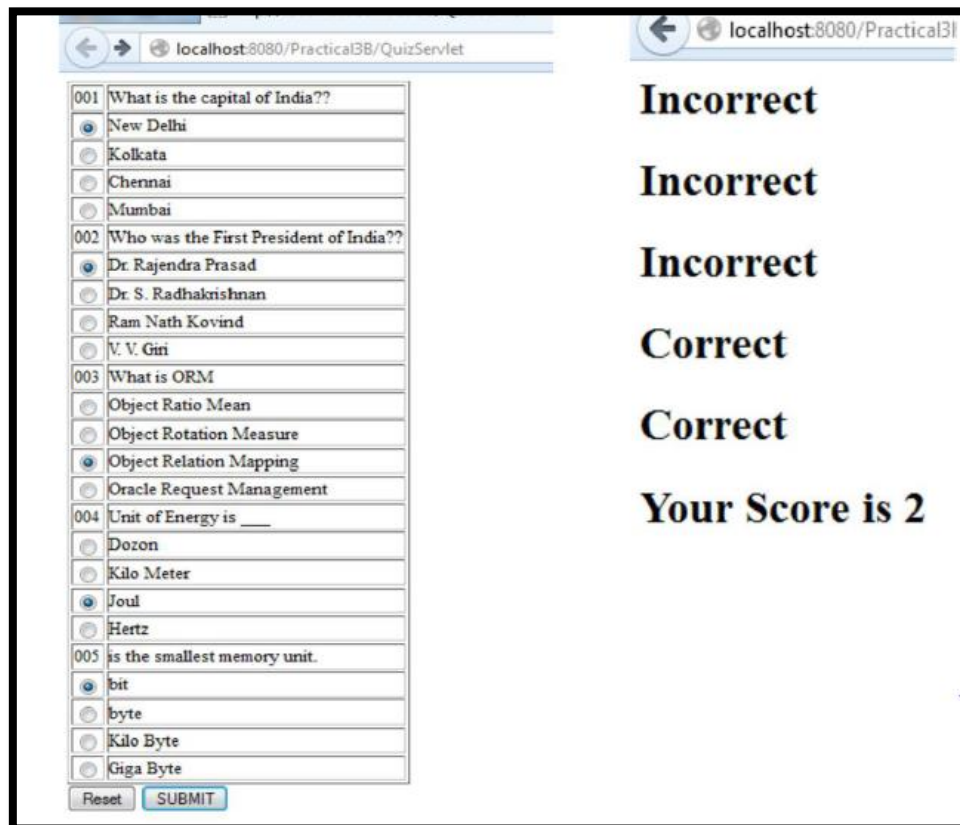
```
package mypack;
import java.io.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class QuizServlet extends HttpServlet
{
public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException
{ response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
out.println("<form action=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.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;
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","admin123");
Statement stmt = con.createStatement();
ResultSet res = stmt.executeQuery("select ans from quiz");
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:-

The screenshot shows a web browser window with the URL `localhost:8080/Practical3B/QuizServlet`. The page displays a quiz with five questions and their answers. The questions and answers are as follows:

Question ID	Question	Selected Answer	Result
001	What is the capital of India??	New Delhi	Incorrect
		Kolkata	Incorrect
		Chennai	
		Mumbai	
002	Who was the First President of India??	Dr. Rajendra Prasad	Incorrect
		Dr. S. Radhakrishnan	
		Ram Nath Kovind	
		V. V. Giri	Correct
003	What is ORM	Object Ratio Mean	
		Object Rotation Measure	
		Object Relation Mapping	Correct
		Oracle Request Management	
004	Unit of Energy is ____	Dozon	
		Kilo Meter	
		Joul	
		Hertz	
005	is the smallest memory unit.	bit	
		byte	
		Kilo Byte	
		Giga Byte	

At the bottom of the quiz, there are two buttons: "Reset" and "SUBMIT".

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

Follow the same steps for creating project as defined in Practical 1A.

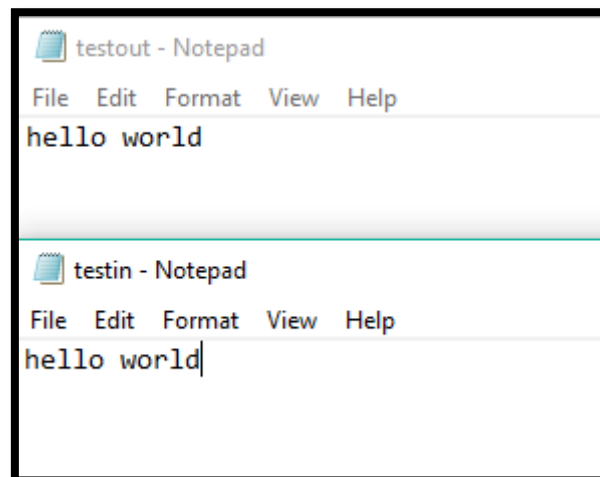
Required Files:

1. index.java(Servlet File)
2. testin.txt(Write some contents into the file)
3. testout.txt(keep it blank)

index.java

```
import java.io.*;
import java.nio.*;
import java.nio.channels.*;
public class index
{
public static void main(String args[]) throws IOException
{
FileInputStream input = new FileInputStream ("D:\\testin.txt");
// Path of Input text file
ReadableByteChannel source = input.getChannel();
FileOutputStream output = new FileOutputStream ("D:\\testout.txt");
// Path of Output text file
WritableByteChannel destination = output.getChannel();
copyData(source, destination);
source.close();
destination.close();
}
private static void copyData(ReadableByteChannel src, WritableByteChannel d
est) throws IOException
{
ByteBuffer buffer = ByteBuffer.allocateDirect(20 * 1024);
while (src.read(buffer) != -1)
{
// The buffer is used to drained
buffer.flip();
// keep sure that buffer was fully drained
while (buffer.hasRemaining())
{
```

```
dest.write(buffer);  
}  
buffer.clear(); // Now the buffer is empty, ready for the filling  
} } }
```

Output:-

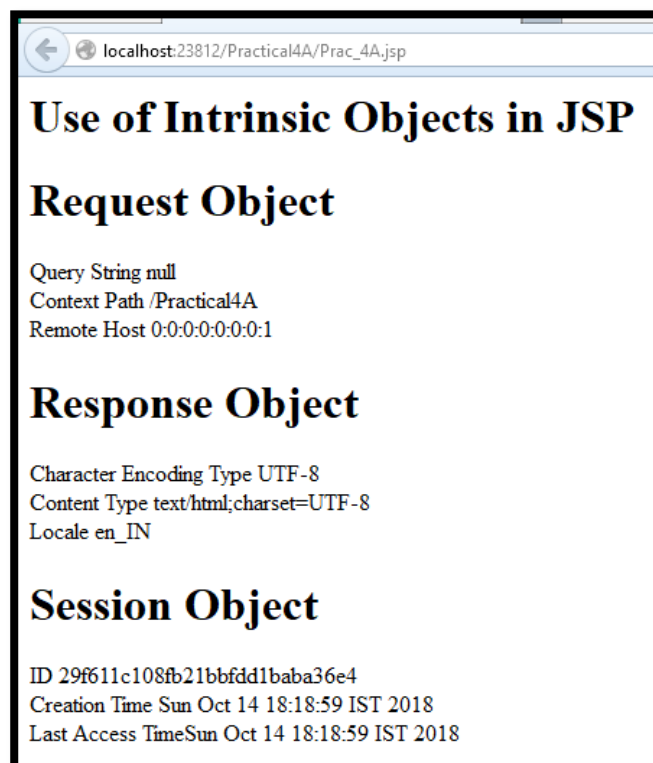
4a. Develop a simple JSP application to display values obtained from the use of intrinsic objects of various types.

Create JSP Page under Web Pages Folder.

Index.jsp

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

Output:-



4b. Develop a simple JSP application to pass values from one page to another with validations.

Follow the same steps for creating project as defined in Practical 1A.

Required Files:

1. index.html
2. Validate.jsp
3. CheckerBean.java

index.html

```
<html><head><title>User Information Page</title>
</head> <body>
<form action="Validate.jsp">
Enter Your Name<input type="text" name="name" ><br>
Enter Your Age<input type="text" name="age" ><br>
Select Hobbies<input type="checkbox" name="hob" value="Singing">Singing
<input type="checkbox" name="hob" value="Reading">Reading Books
<input type="checkbox" name="hob" value="Football">Playing Football<br>
Enter E-mail<input type="text" name="email" ><br>
Select Gender<input type="radio" name="gender" value="male">Male
<input type="radio" name="gender" value="female">Female
<input type="radio" name="gender" value="other">Other<br>
<input type="hidden" name="error" value="">
<input type="submit" value="Submit Form">
</form> </body> </html>
```

Validate.jsp

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

localhost:23812/Practical4B/

Enter Your Name

Enter Your Age

Select Hobbies Singing Reading Books Playing Football

Enter E-mail

Select Gender Male Female Other

localhost:8080/Practical4B/Validate.jsp?name=+++++&age=111&hob=Singing&hob=

Validation Page

Enter Your Name

Enter Your Age

Select Hobbies Singing Reading Books Playing Football

Enter E-mail

Select Gender Male Female Other

Enter First Name

Age Invalid

4c. Create a registration and login JSP application to register and authenticate the user based on username and password using JDBC.**Following files are required:**

1. Register.html
2. Register.jsp
3. Login.html
4. Login.jsp

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 FFFFFAAIIII LLLL !!!!"); %><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

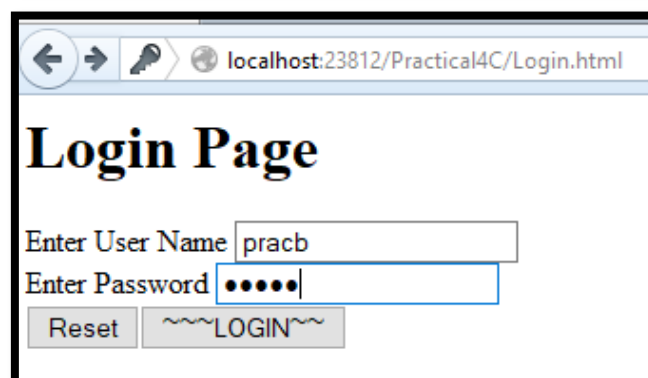
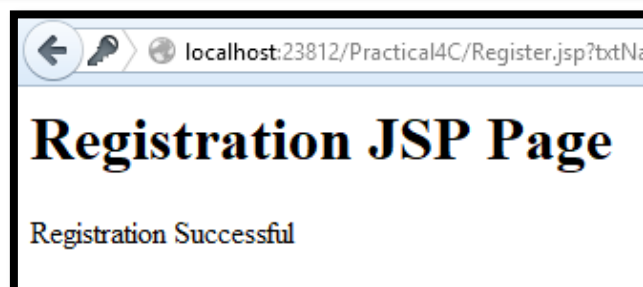
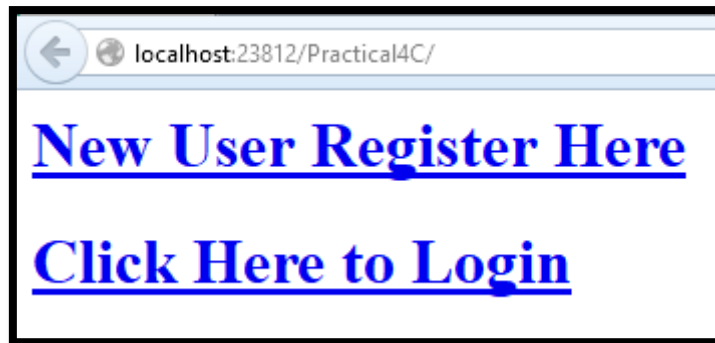
```

<%@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 SUCCESSFULL ~~~ </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:-



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.

Following files are required:

1. index.html
2. UpdateEmp.java

Create database in MySQL.

```
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');
```

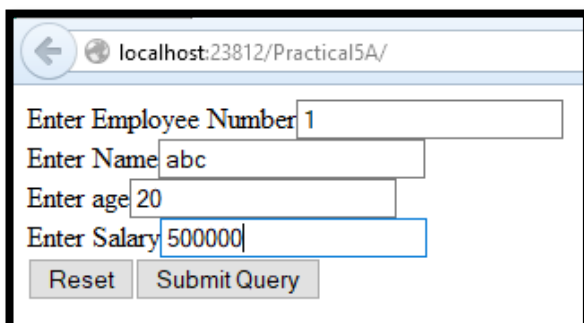
index.html

```
<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>
```



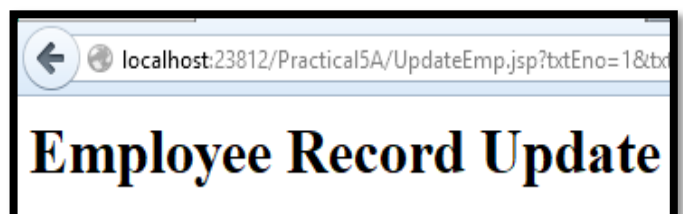
localhost:23812/Practical5A/

Enter Employee Number

Enter Name

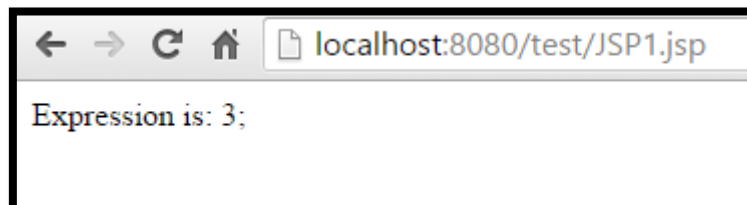
Enter age

Enter Salary



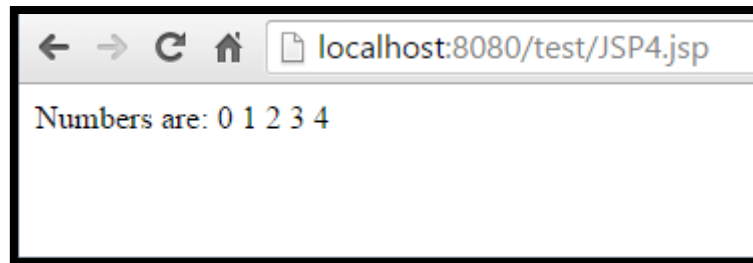
5b. Create a JSP page to demonstrate the use of Expression language.**Arithmetic Operation****index.jsp**

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Expression Language</title>
</head>
<body>
<a>Expression is:</a>
${1+2};
</body>
</html>
```

**Looping**

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Expression Language</title>
</head>
<body>
<%! int num=5; %>
<% out.println("Numbers are:");
```

```
for(int i=0;i<num;i++){  
out.println(i);  
}%>  
</body>  
</html>
```



5c. Create a JSP application to demonstrate the use of JSTL. Basic insert, update and delete example using core and sql tag libraries in JSTL.

Required Files:

Example 1: Core tags

index.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Core Tag JSP2</title>
</head>
<body>
<c:catch var="srException">
<% int num = 10/0; %>
</c:catch>
The Exception is : ${srException}
</body>
</html>
```

Output:-



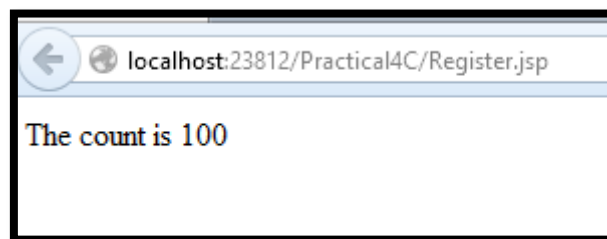
Example 2:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
```



```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Core Tag JSP5</title>
</head>
<body>
<c:set var="count" value="100"/>
<c:if test="{count == 100}">
  <c:out value="The count is 100"/>
</c:if>
</body>
</html>
```

Output:-



Example 3: SQL Tags

Step 1

Login to the database in MYSQL

Step 2

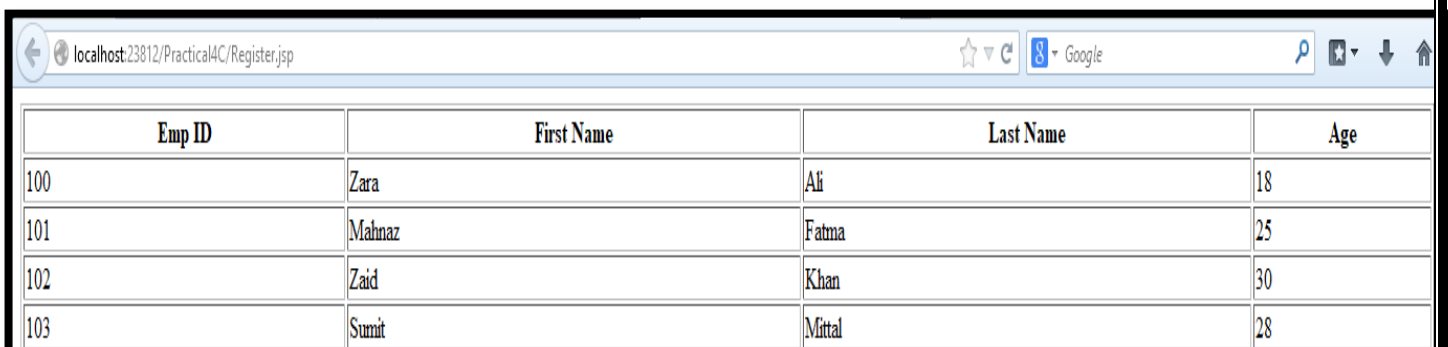
Create the **Employee** table in the **TEST** database as follows — —

```
mysql> use pracc;
mysql> create table Employees
(
  id int not null,
  age int not null,
  first varchar (255),
  last varchar (255)
);
mysql> INSERT INTO Employees VALUES (100, 18, 'Zara', 'Ali');
mysql> INSERT INTO Employees VALUES (101, 25, 'Mahnaz', 'Fatma');
mysql> INSERT INTO Employees VALUES (102, 30, 'Zaid', 'Khan');
mysql> INSERT INTO Employees VALUES (103, 28, 'Sumit', 'Mittal');
```

index.jsp

```
<%@ page import = "java.io.*,java.util.*,java.sql.*"%>
<%@ page import = "javax.servlet.http.*,javax.servlet.*" %>
<%@ 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>JSTL sql:query Tag</title>
</head> <body>
<sql:setDataSource var = "snapshot" driver = "com.mysql.jdbc.Driver"
url = "jdbc:mysql://localhost/pracc"
user = "root" password = "admin123"/>
<sql:query dataSource = "${snapshot}" var = "result">
SELECT * from Employees;
</sql:query>
<table border = "1" width = "100%">
<tr>
<th>Emp ID</th>
<th>First Name</th>
<th>Last Name</th>
<th>Age</th>
</tr>
<c:forEach var = "row" items = "${result.rows}">
<tr>
<td> <c:out value = "${row.id}"/></td>
<td> <c:out value = "${row.first}"/></td>
<td> <c:out value = "${row.last}"/></td>
<td> <c:out value = "${row.age}"/></td>
</tr>
</c:forEach>
</table> </body> </html>
```

Output:-



Emp ID	First Name	Last Name	Age
100	Zara	Ali	18
101	Mahnaz	Fatma	25
102	Zaid	Khan	30
103	Sumit	Mittal	28

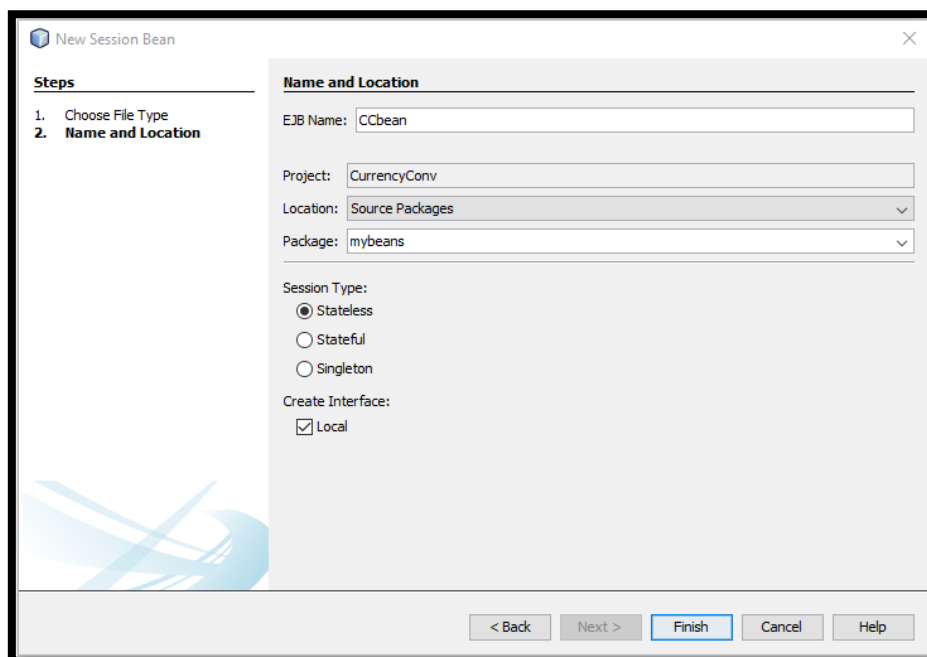
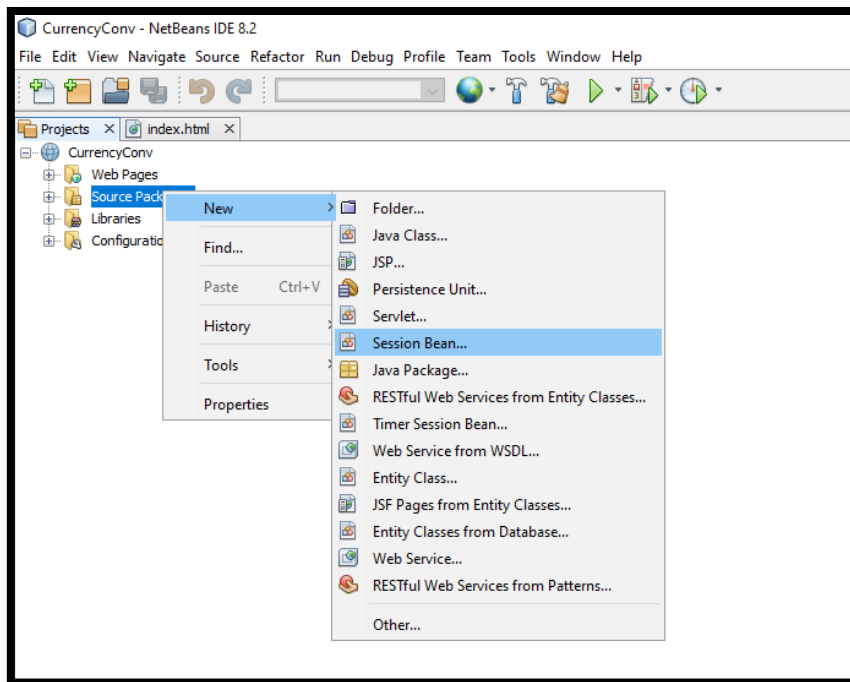
6a. Create a Currency Converter application using EJB.

Follow the same steps for creating project as defined in Practical 1A.

Required Files:

1. index.html
2. CCServlet.java(Servlet File)
3. CCBean.java(Stateless Session Bean File)

To create Stateless Session Bean (CCBean.java) follows the steps:



index.html

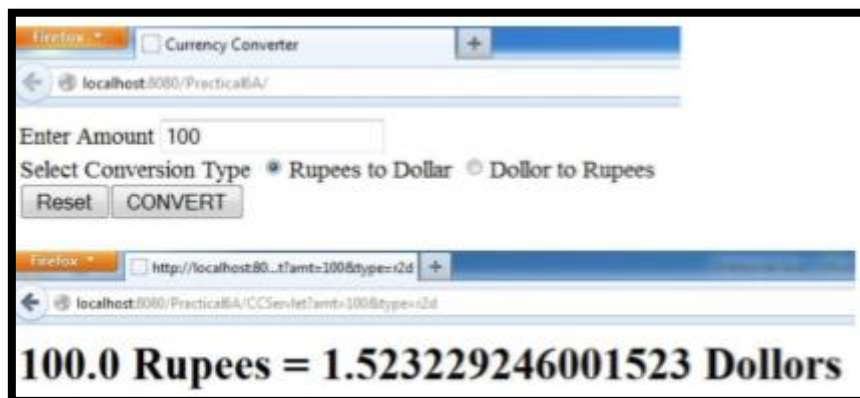
```
<html><head><title>Currency Converter</title></head> <body>
<form action="CCServlet" >
Enter Amount <input type="text" name="amt"><br>
Select Conversion Type
<input type="radio" name="type" value="r2d" checked> Rupees to Dollar
<input type="radio" name="type" value="d2r" >Dollor to Rupees<br>
<input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

```
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;
}}
```



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

Follow the same steps for creating project as defined in Practical 1A.

Required Files:

1. RoomBook.html
2. RBServlet.java
3. RRBean.java
4. Create Database using MYSQL

Database:

Create table roombook(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');

RoomBook.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.*;
```

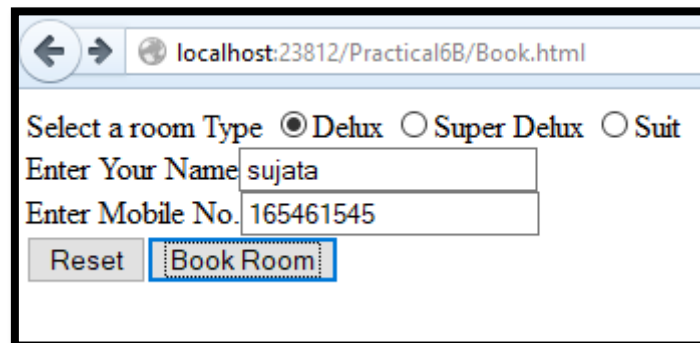
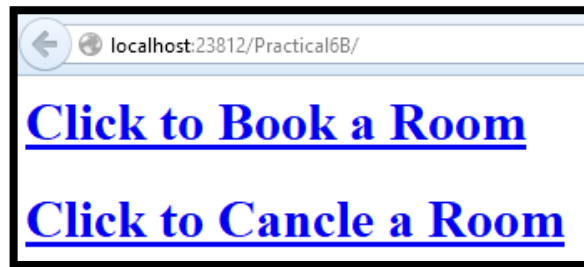
```
import javax.servlet.http.*;
import javax.ejb.EJB;
import mybeans.RRBean;
public class RBServlet extends HttpServlet
{
    @EJB RRBean obj;
    public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException
    {
        PrintWriter out=response.getWriter();
        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

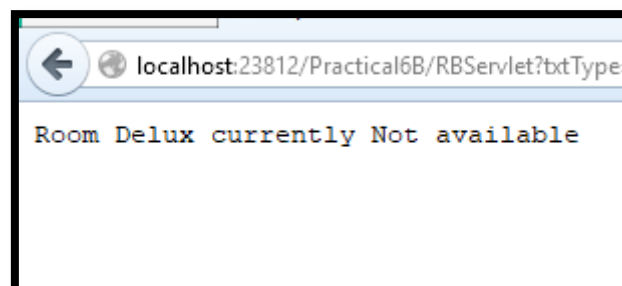
To create Stateless Session Bean (RRBean.java) refer the steps from 6A.

```
package mybeans;
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","admin123");
            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:-

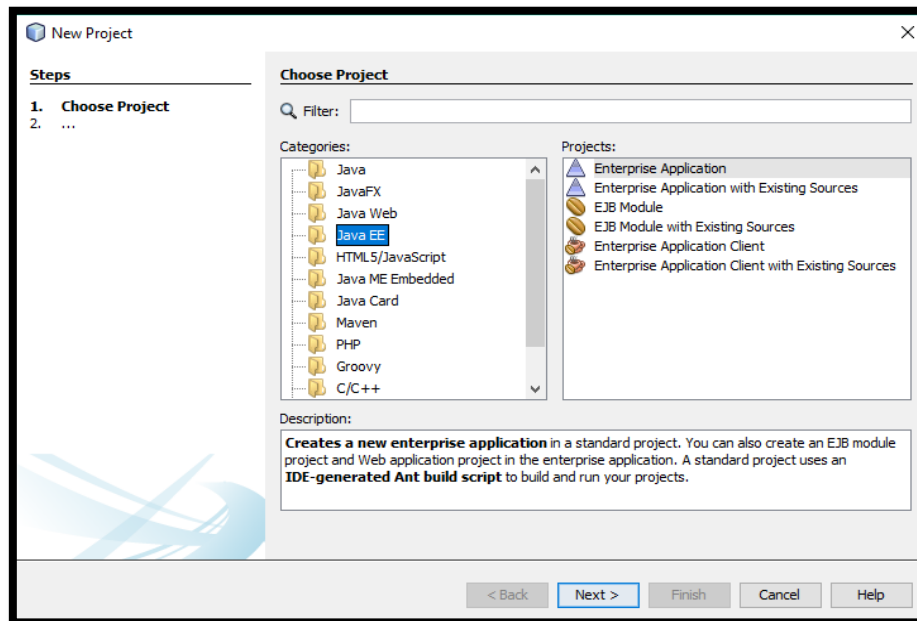
A screenshot of a web browser window. The address bar shows 'localhost:23812/Practical6B/Book.html'. The form contains the following elements:
- Radio buttons for 'Delux' (selected), 'Super Delux', and 'Suit'.
- An input field for 'Enter Your Name' with the value 'sujata'.
- An input field for 'Enter Mobile No.' with the value '165461545'.
- A 'Reset' button and a 'Book Room' button.



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

For creating Stateful Session Bean follow the given steps:

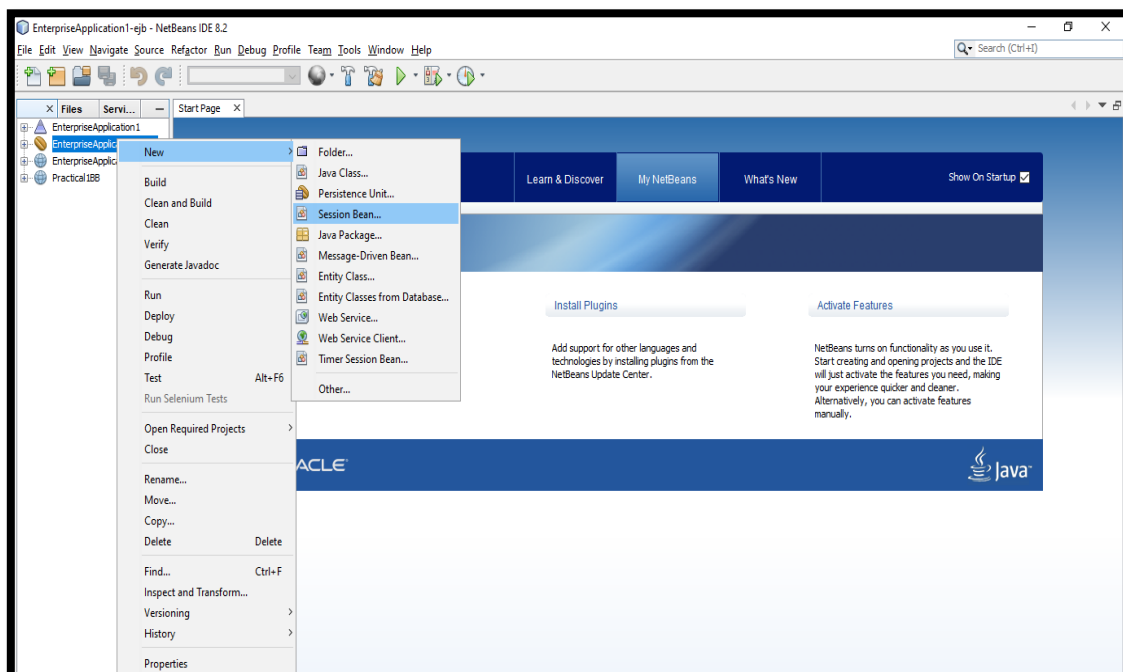
Step 1: Go to File->New Project->Java EE->Enterprise Application->Select Next



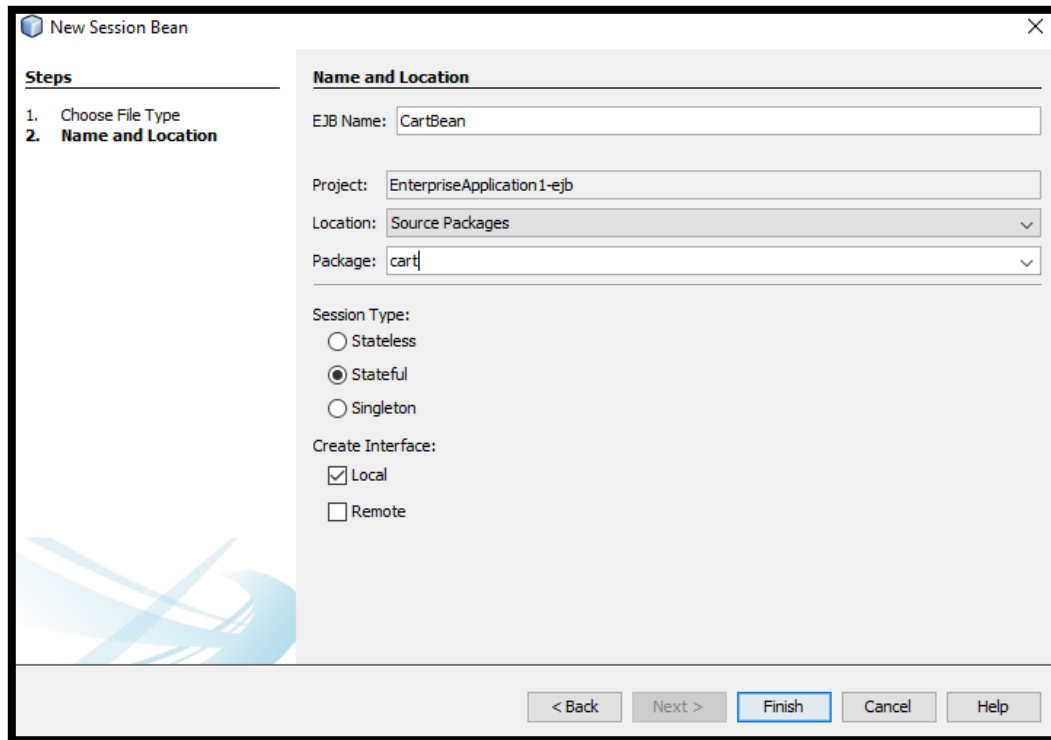
Step 2: Provide Application Name

Step 3: Provide Server

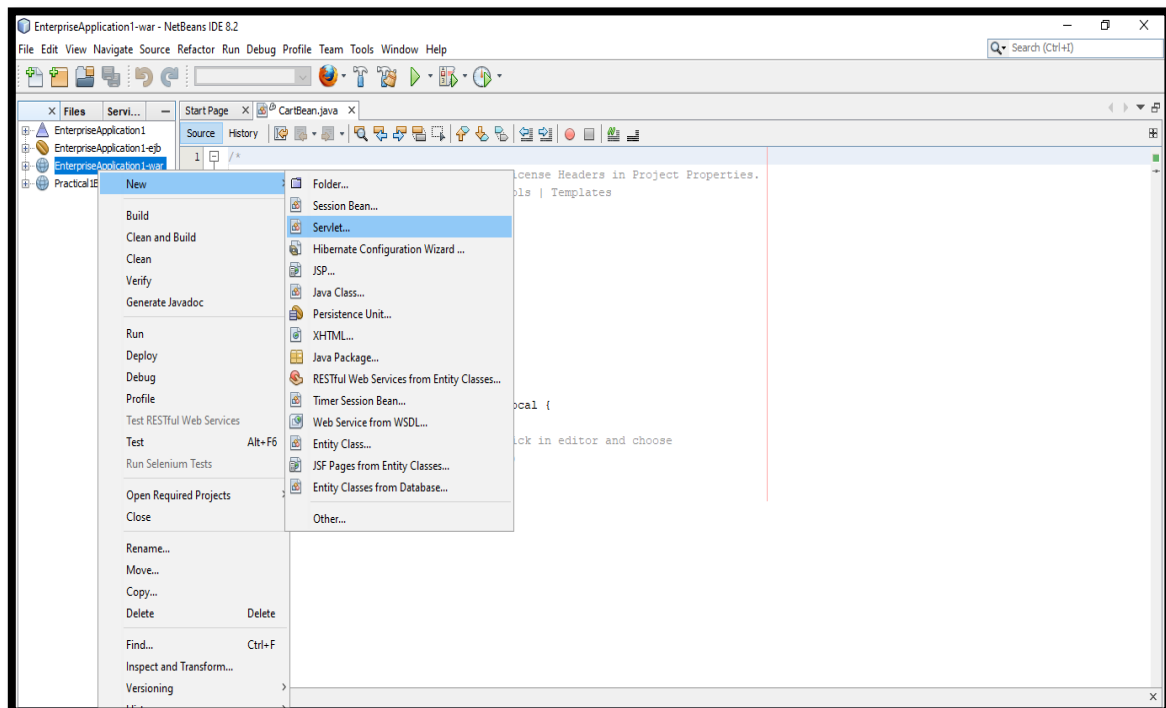
Step 4: Once Application is created follow the given steps below:



Step 5: EJB Name->Provide Package Name->Select Stateful Session Type-> Select Local Interface



Step 6: Create Servlet File by right clicking on EnterpriseApplication1-war as given below:



You have two files under EnterpriseApplication-ejb->Source Packages->cart->CartBean.java and CartBeanLocal.java

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.*;
import javax.ejb.*;
@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;
}}
```

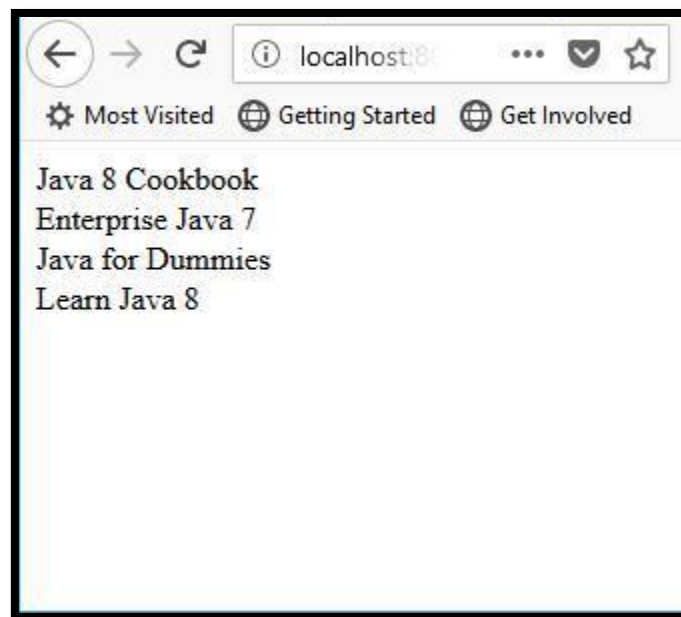
CartTestServlet.java

```
package testcart;
import cart.CartBeanLocal;
import java.io.*;
import java.util.*;
import java.util.logging.*;
```

```
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:-



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

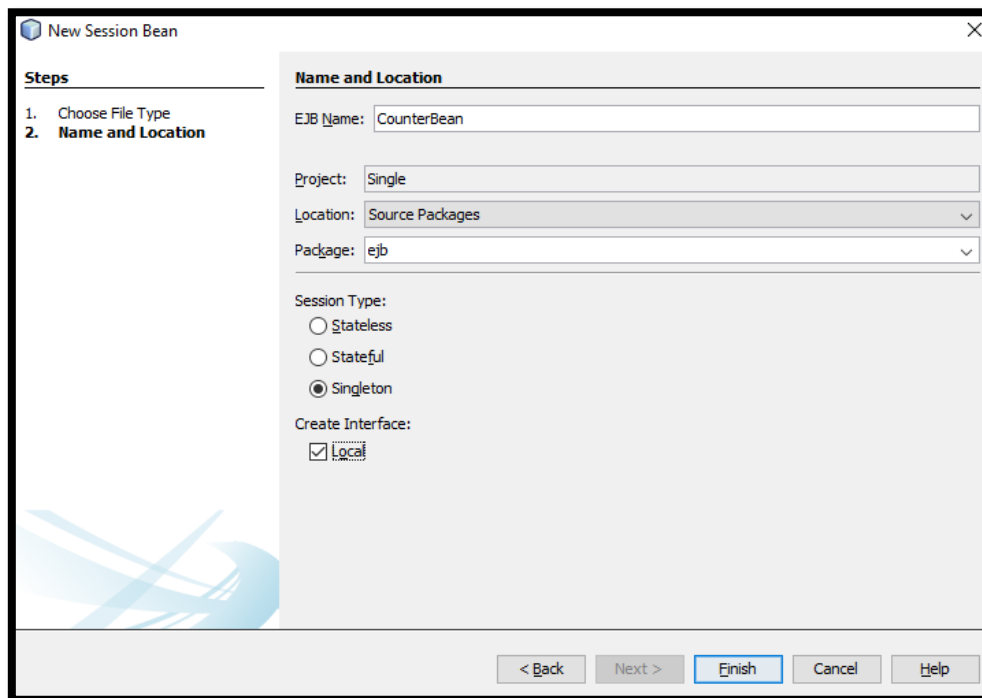
Follow the steps for creating new project as defined in Practical 1A.

Files Required:

1. CounterBean.java
2. Count.java
3. HitCountPage.xhtml

For Creating Singleton Session Bean:-

Go to Source Packages->New->Other->Session Bean->Select Singleton Session Type->Select Local Interface



CounterBean.java

```
package counter.ejb;
import javax.ejb.Singleton;
@Singleton
public class CounterBean
{
private int hits = 1;
public int getHits()
```



```
{  
    return hits++ ;  
}}
```

Count.java

```
package mypack;  
import java.io.Serializable;  
import javax.ejb.EJB;  
import javax.enterprise.context.ConversationScoped;  
import javax.inject.Named;  
import counter.ejb.CounterBean;  
@Named("count")  
@ConversationScoped  
public class Count implements Serializable  
{  
    @EJB  
    private CounterBean counterBean;  
    private int hitCount;  
    public Count()  
    {  
        this.hitCount = 0;  
    }  
    public int getHitCount()  
    {  
        hitCount = counterBean.getHits();  
        return hitCount;  
    }  
    public void setHitCount(int newHits)  
    {  
        this.hitCount = newHits;  
    }  
}}
```

HitCountPage.xhtml

```
<?xml version='1.0' encoding='UTF-8' ?>  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml"  
xmlns:h="http://xmlns.jcp.org/jsf/html">
```

```
<h:head>
<title>Page Hit Counter Using Singleton Session Bean ~~~~ </title>
</h:head>
<h:body>
<h1>Welcome to Hit Count Page</h1>
Page was hit #{count.hitCount} times
</h:body>
</html>
```



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

Follow the same steps for creating project as defined in Practical 1A.

Required Files:

1. index.jsp
2. VisitorStat.java(Stateless Session Bean File)
3. Add MYSQL connector in Libraries.
4. Database in MYSQL

Database Tables:-

```
create database bscit;
```

```
use bscit;
```

```
create table hello(fisrttime varchar(20), visitorname varchar(20), visits in(3));
```

index.jsp

```
<%@page import="javax.naming.InitialContext, ejb.VisitorStat"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<%!
private static VisitorStat visit;
public void jsplnit()
{
try
{
InitialContext ic = new InitialContext();
visit = (VisitorStat) ic.lookup("java:global/VisitorStatistics/VisitorStat");
}
catch (Exception ex)
{
System.out.println("Could not create timer bean." + ex.getMessage());
}
}
%>
<%
visit.addVisitor(request.getRemoteAddr());
%>
<html> <head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

```
<title>Welcome </title>
</head>
<body style="background-color: pink;">
<h1 style="text-align: center;">Hello World </h1>
</body>
</html>
```

VisitorStat.java

```
package ejb;
import java.sql.*;
import javax.annotation.*;
import javax.ejb.Stateless;
@Stateless
public class VisitorStat
{
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/bscit", "root",
"admin123");
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 hello (visitorname, visits) VALUES('" + host + "',1)";
stmt.executeUpdate(query);
}
catch(SQLException e)
{
try
{
stmt = conn.createStatement();
query = "UPDATE hello SET visits = visits + 1 WHERE visitorname = '" + host +
'";
stmt.executeUpdate(query);
}
catch(SQLException ex)
{
System.err.println("Cannot update: " + ex.getMessage());
}}}}}
```

7c. Develop simple Marks Entry Application to demonstrate accessing database using EJB.**Following are the files required:**

1. index.jsp
2. MarkEntry.java(Stateless Session Bean file)
3. MarkEntryServ.java(Servlet file)

Database:

create database tyit;

use tyit;

create table marks(Roll_no varchar(10), Name varchar(20), P1 varchar(20), p2 varchar(20), p3 varchar(20), p4 varchar(20), p5 varchar(20));

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> Student's Record</title>
</head>
<body>
<form method = "post" action="MarksentryServ">
<h1> Enter Students Record</h1>
Students Roll No: <input type="text" name="textno"/></br>
Students Name: <input type="text" name="txtname"/></br>
Paper 1 (SPM):<input type="text" name="textp1"/></br>
Paper 2 (AWP):<input type="text" name="textp2"/></br>
Paper 3 (Linux):<input type="text" name="textp3"/></br>
Paper 4 (IoT):<input type="text" name="textp4"/></br>
Paper 5 (EJ):<input type="text" name="textp5"/></br>
<input type="submit" value="Submit"/>
</form>
</body>
</html>
```

Stateless Session Bean File MarkEntry.java

```
package ejb;
import javax.ejb.Stateless;
import javax.ejb.LocalBean;
import java.sql.*;
@Stateless
@LocalBean
public class MarkEntry
{
    public void entry(String srollno, String sname, String p1, String p2, String p3,
    String p5)
    {
        Connection con=null;
        Statement st=null;
        ResultSet rs= null;
        try
        {
            Class.forName("com.mysql.jdbc.Driver");
            con=DriverManager.getConnection("jdbc:mysql://localhost/tyit","root","admin123");
            st=con.createStatement();
            st.executeUpdate("insert into marks values(' "+srollno+" ',' "+sname+" ',' "+
            +p1+" ',' "+p2+" ',' "+p3+" ',' "+p4+" ',' "+p5+" '");
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
    public String submit(String name)
    {
        return name.toUpperCase()+" record is submitted!!!";
    }
}
```

Servlet file**MarkentryServe.java**

```
package Servlet;
import java.io.*;
import javax.servlet.ServletException;
import javax.servlet.http.*;
import javax.servlet.annotation.WebServlet;
import javax.ejb.EJB;
import ejb.MarkEntry;
@WebServlet(name="MarkentryServ", urlPatterns={"/MarkentryServe"})
public class MarkentryServe extends HttpServlet
{
    @EJB
    Markentry bean;
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse
    response)throws ServletException, IOException
    {
        PrintWriter out=response.getWriter();
        String id, name, p1, p2,p3,p4,p5;
        id=request.getParamter("txtno");
        name=request.getParameter("txtname");
        p1= request.getParameter("txtp1");
        p2= request.getParameter("txtp2");
        p3= request.getParameter("txtp3");
        p4= request.getParameter("txtp4");
        p5= request.getParameter("txtp5");
        bean.entry(id,name,p1,p2,p3,p4,p5);
        out.println(bean.subit(name));
    }
}
```


8A. Develop a Simple Inventory Application using JPA

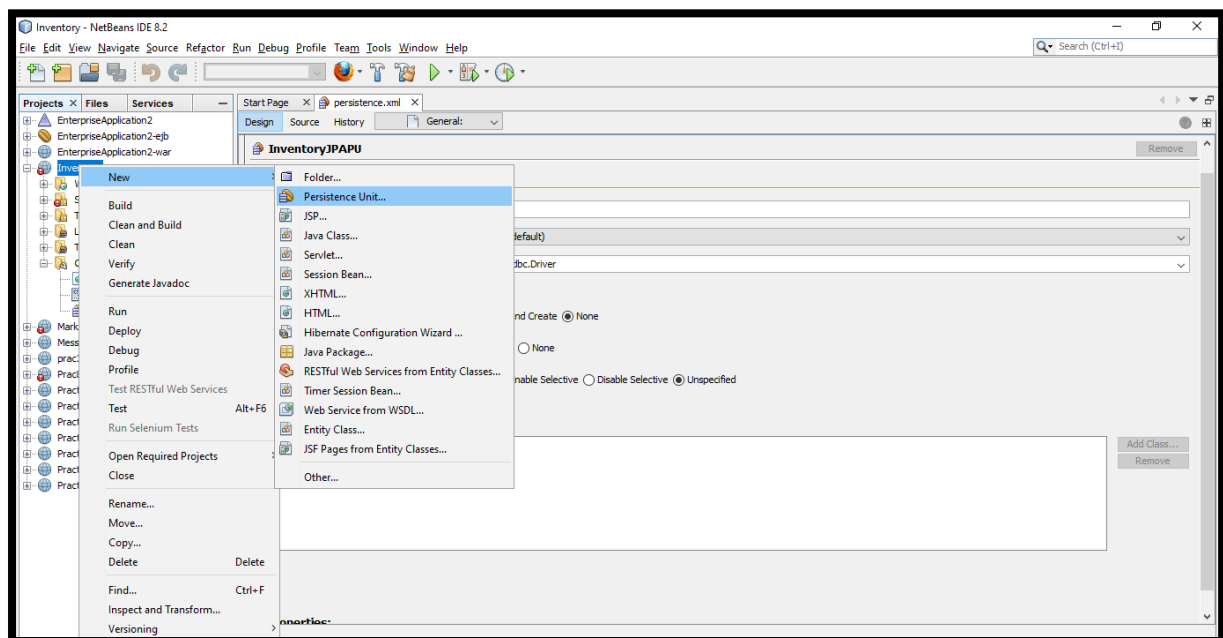
Create the project as shown in practical 1A

Following are the files required:

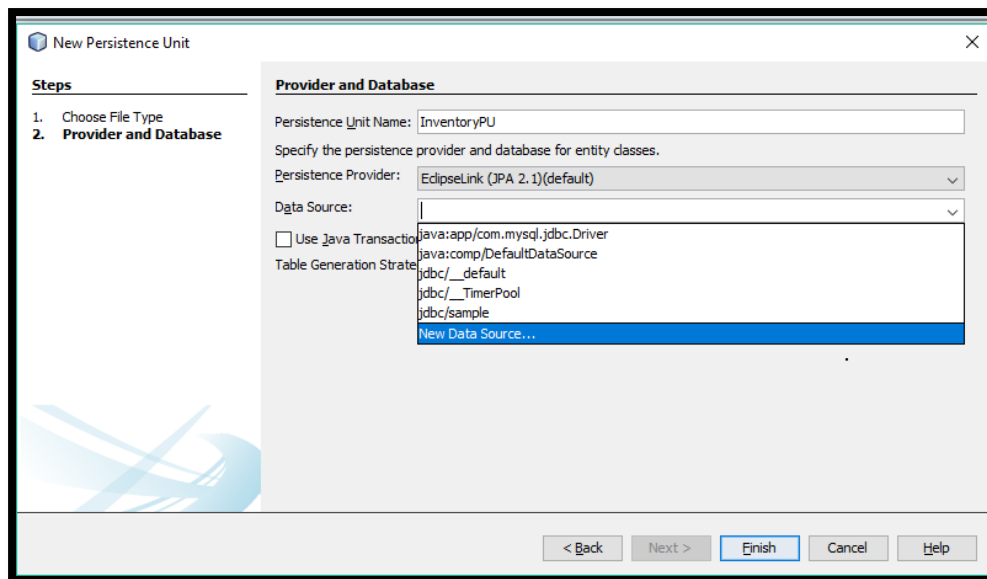
1. index.jsp
2. InventoryView.jsp
3. Inventory.java(Simple Java class or Entity Class)
4. Persistence Unit

For creating persistence unit follow the given steps:

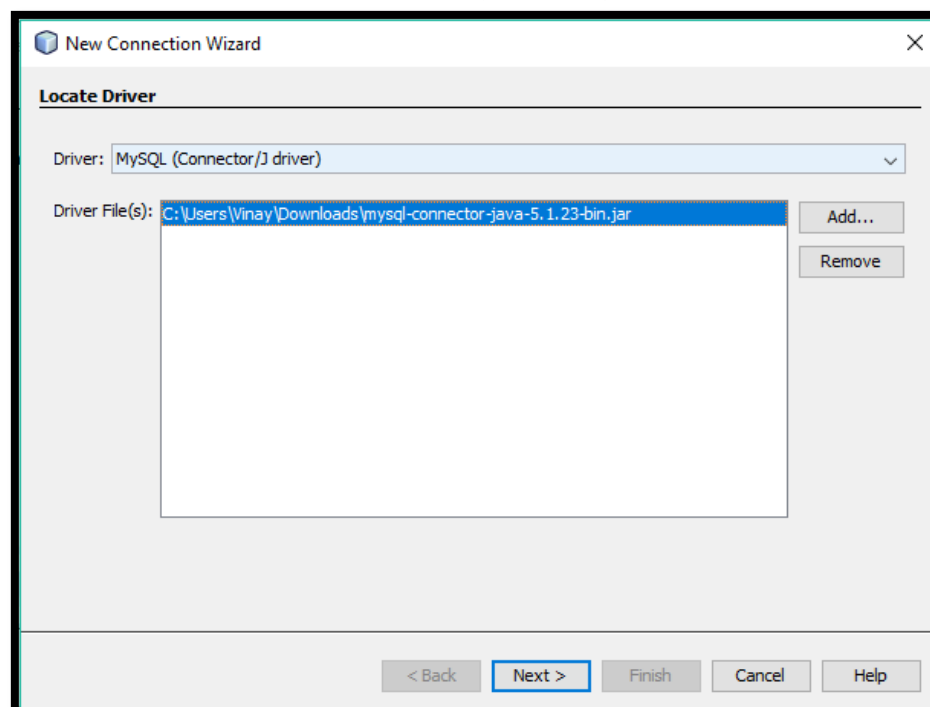
Right Click on your Project->Go to New->Other->Persistence->Persistence Unit



1. Give Persistence Name
2. Select Persistence Provider (EclipseLink is the default persistence provider for Glassfish Server)
3. Uncheck "Use Java Transaction APIs" checkbox
4. Select None for Table Generation Strategy
5. Last and most important step is to select Data Source.
6. Select New Data Source Option.



7. Provide JNDI Name and Select Database Connection. Here, if Database is already available to which you wish to connect to select it or else follow the given steps:
 - a. Select New Database Connection.
 - b. In New Connection Wizard, it will ask you for Driver Select MYSQL Connector, and Driver file(JAR file) by clicking on Add button.



- c. Provide Database Name, Username, Password and establish connection.

Table Creation:

```
create table Inventory(itemid int(3) primary key auto_increment, itemname
char(20), itemprice int(3), itemqty int(3));
```

Inventory.java

```
package myApp;
import javax.persistence.*;
@Entity
@Table(name="Inventory")
public class Inventory
{
    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    @Column(name="itemid", unique=true, updatable=false)
    Integer itemid;
    @Column(name="itemname")
    String itemname;
    @Column(name="itemprice")
    String itemprice;
    @Column(name="itemqty")
    String itemqty;
    public Inventory()
    {}
    public Integer getItemid()
    {
        return itemid;
    }
    public void setItemid(Integer itemid)
    {
        this.itemid=itemid;
    }
    public String getItemname()
    {
        return itemname;
    }
    public void setItemname(String itemname)
    {
        this.itemname=itemname;
    }
}
```

```
public String getItemprice()
{
return itemprice;
}
public void setItemprice(String itemprice)
{
this.itemprice=itemprice;
}
public String getItemqty()
{
return itemqty;
}
public void setItemqty(String itemqty)
{
this.itemqty=itemqty;
}
```

index.jsp

```
<%@page contentType="txt/html" pageEncoding=UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="txt/html; charset=UTF-8">
<title> JSP Page</title>
</head>
<body>
<h1>Enter details: </h1>
<form action="InventoryView.jsp" method="post">
Item name: <input type="text" value="iname"/>
Item price: <input type="text" value="iprice"/>
Item quantity: <input type="text" value="iqty"/>
<input type="submit" name="btnsubmit" value="Submit"/>
</form></body></html>
```

InventoryView.jsp

```
<%@page contentType="txt/html" pageEncoding=UTF-8"
import ="java.util.List, java.util.Iterator, ,myApp.Inventory,
javax.persistence.*"%>
```

```
<!DOCTYPE html>
<%!
EntityManagerFactory emf;
EntityManager em;
EntityTransaction tx;
List<Inventory> inventory;
%>
<%
emf=Persistence.createEntityManagerFactory("InventoryJPAPU");
em=emf.createEntityManager();
String submit=request.getParameter("btnsubmit");
if(submit!=null &&("Submit").equals(submit))
{
String name=request.getParameter("iname");
String price=request.getParameter("iprice");
String qty=request.getParameter("iqty");
Inventory gb=new Inventory();
gb.setItemname(name);
gb.setItemprice(price);
gb.setItemqty(qty);
tx=em.getTransaction();
tx.begin();
em.persist(gb);
tx.commit();
}
catch(Exception e)
{
if(tx!=null)
tx.rollback();
throw e;
}
response.sendRedirect("InventoryView.jsp");
}
try
inventory=em.createQuery(select * from Inventory").getResult();
}
catch (Exception e)
{
throw e;
}
}
```


8C. Create a Simple JPA application to store and retrieve Book Details.

Follow the steps for creating project as shown in practical 1A

Following are the file required:-

1. index.jsp
2. create retrieveBookDetails.java under servlet package
3. create BookWS.java under webservice
4. persistence unit.xml file

Database Creation:-

```
CREATE DATABASE bookshop;
```

```
USE bookshop;
```

```
CREATE TABLE Books
```

```
(  
    ISBN varchar(20) PRIMARY KEY,  
    BookName varchar(100),  
    BookPrice varchar(10));
```

```
INSERT INTO books VALUES ("808-97365-80", "Java EE Project Using EJB 3, JPA  
and Struts For Beginners", "650");
```

```
INSERT INTO books VALUES ("978-34-13394-1", "LAMP Programming For  
Beginners", "450");
```

```
INSERT INTO books VALUES ("21-783592-1", "Struts 2.0 For Beginners", "350");
```

```
INSERT INTO books VALUES ("1-34523-880", "Hibernate 3 for Beginners",  
"550");
```

```
INSERT INTO books VALUES ("783-983242-23", "Oracle For Professionals",  
"750");
```

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>SOAP Client - Get Book Details</title>
```

```
  </head>
```

```
<body style="background-color: pink;">
<form name="frmgetBookDetails" method="post"
action="retrieveBookDetails">
ISBN: <input type="text" name="isbn" id="isbn"/>
<input type="submit" name="btnSubmit" value="Submit"/>
    </form>
</body>
</html>
```

retrieveBookDetails.java

```
package servlet;
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.xml.ws.WebServiceRef;
import webservice.BookWS;

@WebServlet(name = "retrieveBookDetails", urlPatterns =
{"/retrieveBookDetails"})
public class retrieveBookDetails extends HttpServlet
{
    @WebServiceRef(wsdlLocation = "WEB-
INF/wsdl/localhost_8080/BookWS/BookWS.wsdl")
    private BookWS service;

    @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 retrieveBookDetails</title>");
        }
    }
}
```



```
        out.println("</head>");
        out.println("<body style='background-color: pink;'>");
        java.lang.String isbn = request.getParameter("isbn");
        out.println(getBookDetails(isbn));
        out.println("</body>");
        out.println("</html>");
    } }
    private String getBookDetails(java.lang.String isbn)
    {
        webservice.BookWS port = service.getBookWSPort();
        return port.getBookDetails(isbn);
    }
}
```

BookWS.java

```
package webservice;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import javax.ejb.Stateless;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebService;
@WebService(serviceName = "BookWS")
@Stateless()
public class BookWS
{
    @WebMethod(operationName = "getBookDetails")
    public String getBookDetails(@WebParam(name = "isbn") String isbn)
    {
        try
        {
            Class.forName("com.mysql.jdbc.Driver").newInstance();
            Connection dbcon = DriverManager.getConnection
            ("jdbc:mysql://localhost/bookshopcr", "root", "admin123");
            Statement stmt = dbcon.createStatement();
            String query = "SELECT * FROM Books WHERE isbn = '" + isbn + "'";
            ResultSet rs = stmt.executeQuery(query);
```

```
rs.next();
String bookDetails = "The name of the book is <b>" + rs.getString("BookName")
+ "</b> and its cost is <b>" + rs.getString("BookPrice") + "</b>.";
return bookDetails;
    }
catch(SQLException | ClassNotFoundException | InstantiationException |
IllegalAccessException ex)
{
System.out.println("Sorry failed to connect to the Database. " +
ex.getMessage());
    }
    return null;
}
public BookWS getBookWSPort()
{
    throw new UnsupportedOperationException("Not supported yet."); //To
change body of generated methods, choose Tools | Templates.
}
}
```

9B. Develop a Hibernate application to store Feedback of Website Visitor in MySQL Database.

Follow the steps for creating project as shown in practical 1A.

Following are the file required:-

1. GuestBookBean.java
2. hibernate.cfg.xml
3. index.html
4. fb.jsp

Add the JAR files which is required as shown below:-

Right Click on Libraries->Select Add/JAR Folder->Browse for the JAR files present.

From the hibernate-distribution 3.5.0-Beta-2->lib-> directory

- hibernate3-jar

From the hibernate distribution 3.5.0-Beta-2-> required directory

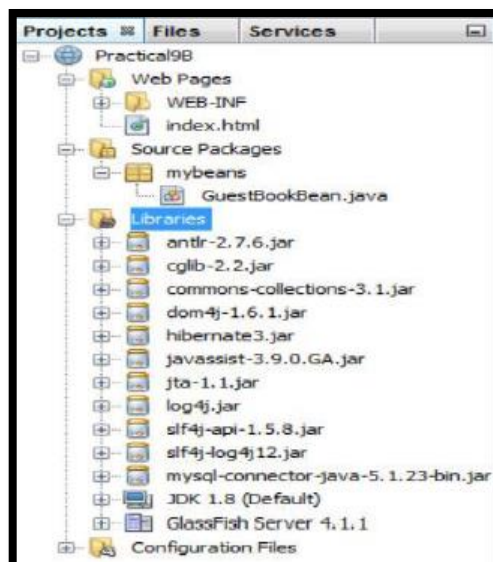
- antlr 2.7.6.jar
- jta 1.1.jar
- javassist 3.9.0 G.A.jar
- commons-collections 3.1.jar
- dom4j-1.6.1.jar
- slf4j-api-1.5.8.jar

From hibernate distribution 3.5.0 Beta-2->lib->bytecode->cglib directory

- cglib 2.2.jar

Form the hibernate annotations 3.4.0 G.A ->lib->test directory

- slf4j-log4j1.2.jar
- log4j.jar



Database Creation:-

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

GuestBookBean.java

```
package mypack;
import javax.persistence.*;
@Entity
@Table(name="guestbook")
public class GuestBookBean implements java.io.Serializable
{
@Id
@GeneratedValue
@Column(name="vno")
private Integer visitorNo;
@Column(name="vname")
private String visitorName;
@Column(name="msg")
private String msg;
@Column(name="mdate")
private String msgDate;
public GuestBookBean()
{}
public Integer getVisitorNo()
{
return visitorNo; }
public String getVisitorName()
{
return visitorName;
}
public String getMsg()
{
return msg; }
public String getMsgDate()
```

```
{
return msgDate; }
public void setVisitorNo(Integer vn)
{
visitorNo = vn ; }
public void setVisitorName(String vn)
{
visitorName=vn;
}
public void setMsg(String m)
{
msg=m;
}
public void setMsgDate(String md)
{
Msgdate=md;
}
```

For Configuring Hibernate follow the given steps:-

Source packages → New → Others → Select Category Hibernate → Hibernate Configuration Wizard

Hibernate.cfg.xml

```
<hibernate-configuration>
<session-factory>
<property
name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
<property
name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
<property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/feedbackdb?zeroDateTimeBehavior=convertToNull</property>
<property name="hibernate.connection.username">root</property>
<property name="hibernate.connection.password">root</property>
<mapping class="mypack.GuestBookBean" />
</session-factory>
</hibernate-configuration>
```

index.html

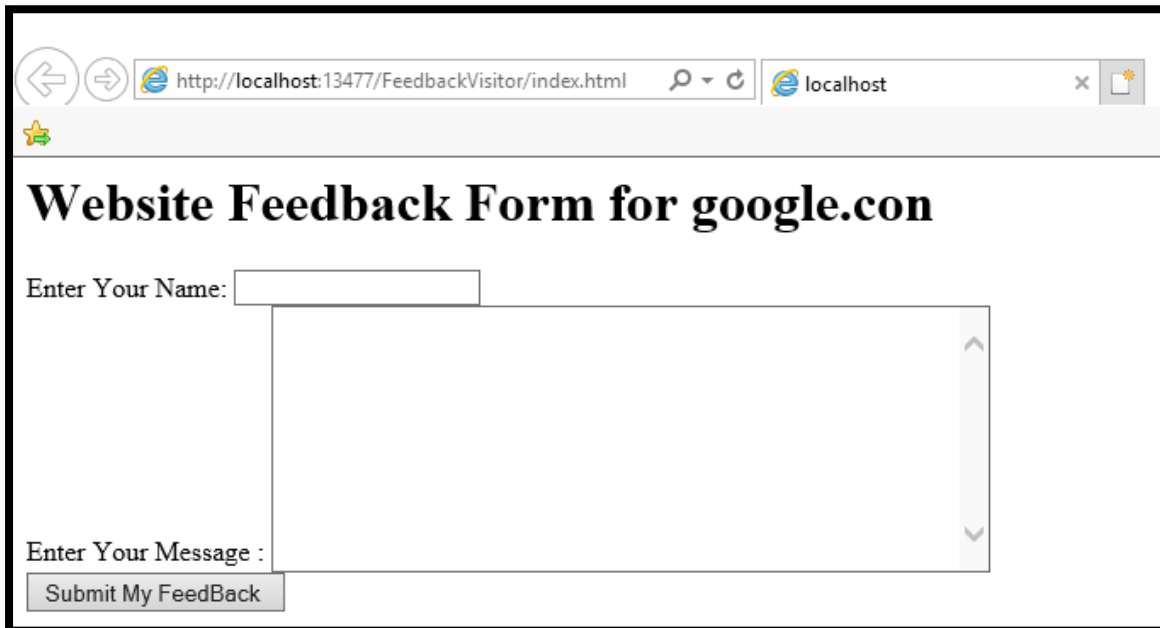
```
<h1>Website Feedback Form for google.con </h1>
<form action="fb.jsp" >
Enter Your Name: <input type="text" name="name" ><br>
Enter Your Message : <textarea rows="10" cols="50" name="message" >
</textarea><br>
<input type="submit" value="Submit My FeedBack " >
</form>
```

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;
GuestBookBean gb = new GuestBookBean();
try
{
tx = hibSession.beginTransaction();
String username = request.getParameter("name");
String usermsg = request.getParameter("message");
String nowtime = ""+new java.util.Date();
gb.setVisitorName(username);
gb.setMsg(usermsg);
gb.setMsgDate(nowtime);
hibSession.save(gb);
tx.commit();
out.println("Thank You for your valuable feedback....");
}catch(Exception e)
{
out.println(e);
}
hibSession.close();
```

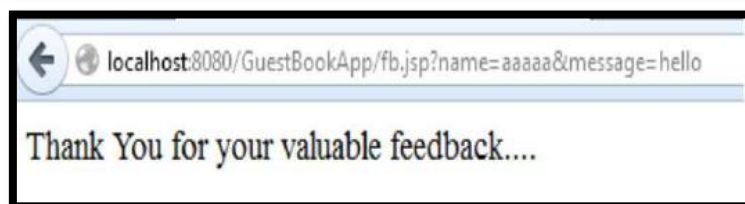
%>

Output:-



A screenshot of a web browser window. The address bar shows the URL `http://localhost:13477/FeedbackVisitor/index.html`. The page title is "Website Feedback Form for google.com". The form contains the following elements:

- A label "Enter Your Name:" followed by a text input field.
- A large text area for the message.
- A label "Enter Your Message :" followed by the text area.
- A button labeled "Submit My FeedBack".



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

Create a One-To-One relationship between a Student and Library in such a way that one student can be issued only one type of book.

Required files:-

1. Student.java(Right Click on Source Packages->New->Java Class)
2. Library.java(Right Click on Source Packages->New->Java Class)
3. persistence.xml
4. OnetoOneExample.java

Create Database

```
create table student(S_ID int(3) primary key, S_NAME varchar(20));
```

```
create table library(B_ID int(3) primary key, B_NAME varchar(20), STUD_S_ID  
varchar(20));
```

Create an entity class Student.java abc package that contains student id (s_id) and student name (s_name);

Student.java

```
package abc;  
import javax.persistence.*;  
@Entity  
public class Student  
{  
    @Id  
    @GeneratedValue(strategy=GenerationType.AUTO)  
    private int s_id;  
    private String s_name;  
    public int getS_id() {  
        return s_id;  
    }  
    public void setS_id(int s_id)  
{  
        this.s_id = s_id;  
    }  
}
```



```
    }  
    public String getS_name()  
{  
    return s_name;  
    }  
    public void setS_name(String s_name) {  
        this.s_name = s_name;  
    }  
}
```

Create another entity class Library.java under abc package that contains book id (b_id), book name (b_name) and an object of student type marked with @OneToOne annotation.

Library.java

```
package abc;  
import javax.persistence.*;  
@Entity  
public class Library  
{  
    @Id  
    @GeneratedValue(strategy=GenerationType.AUTO)  
    private int b_id;  
    private String b_name;  
    @OneToOne  
    private Student stud;  
    public Library(int b_id, String b_name, Student stud) {  
        super();  
        this.b_id = b_id;  
        this.b_name = b_name;  
        this.stud = stud;  
    }  
    public Library()  
    {  
        super();  
    }  
    public int getB_id()  
    {  
        return b_id;  
    }  
}
```

```
}
public void setB_id(int b_id)
{
    this.b_id = b_id;
}
public String getB_name()
{
    return b_name;
}
public void setB_name(String b_name)
{
    this.b_name = b_name;
}
public Student getStud()
{
    return stud;
}
public void setStud(Student stud)
{
    this.stud = stud;
} } }
```

Now, map the entity class and other databases configuration in Persistence.xml file.

Persistence.xml

```
<persistence>
<persistence-unit name="Book_issued">
<class>abc.Student</class>
<class>abc.Library</class>
<properties>
<property name="javax.persistence.jdbc.driver" value="com.mysql.jdbc.Driver
"/>
<property name="javax.persistence.jdbc.url" value="jdbc:mysql://localhost:33
06/mapping"/>
<property name="javax.persistence.jdbc.user" value="root"/>
<property name="javax.persistence.jdbc.password" value="admin123"/>
<property name="eclipselink.logging.level" value="SEVERE"/>
<property name="eclipselink.ddl-generation" value="create-or-extend-
tables"/>
```

```
</properties>  
</persistence-unit>  
</persistence>
```

Create a persistence class `OneToOneExample` under `abc` package to persist the entity object with data.

OneToOneExample.java

```
import javax.persistence.*;  
public class OneToOneExample  
{  
    public static void main(String[] args)  
    {  
        EntityManagerFactory emf = Persistence.createEntityManagerFactory( "Book_issued" );  
        EntityManager em = emf.createEntityManager( );  
        em.getTransaction( ).begin( );  
        Student st1=new Student();  
        st1.setS_id(1);  
        st1.setS_name("Vipul");  
        Student st2=new Student();  
        st2.setS_id(2);  
        st2.setS_name("Vimal");  
        em.persist(st1);  
        em.persist(st2);  
        Library lib1=new Library();  
        lib1.setB_id(101);  
        lib1.setB_name("Data Structure");  
        lib1.setStud(st1);  
        Library lib2=new Library();  
        lib2.setB_id(102);  
        lib2.setB_name("DBMS");  
        lib2.setStud(st2);  
        em.persist(lib1);  
        em.persist(lib2);  
        em.getTransaction().commit();  
        em.close();  
        emf.close();  
    }  
}
```

Output:

After the execution of the program, two tables are generated under MySQL workbench.

Student table - This table contains the student details. To fetch data, run **select * from student** query in MySQL.

S_ID	S_NAME
1	Vipul
2	Vimal

Library table - This table represents the mapping between student and library. To fetch data, run **select * from library** query in MySQL.

B_ID	B_NAME	STUD_S_ID
101	Data Structure	1 [->]
102	DBMS	2 [->]