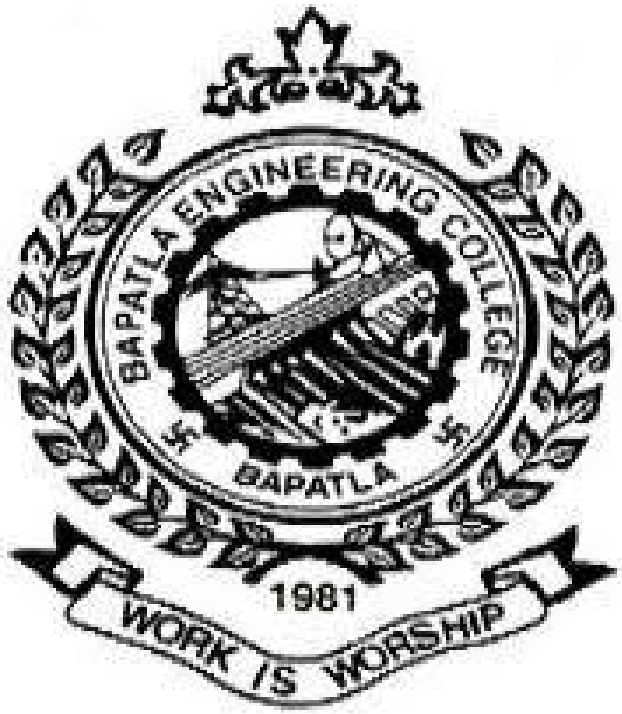
Enterprise programming – II

LAB RECORD

LAB CODE: 14ITL703



DEPARTMENT OF INFORMATION TECHNOLOGY

Bapatla Engineering College :: Bapatla

(Autonomous)

(Affiliated to Acharya Nagarjuna University)

BAPATLA – 522101, A.P

BAPATLA ENGINEERING COLLEGE

DEPARTMENT OF INFORMATION TECHNOLOGY

ENTERPRISE PROGRAMMING -II

CERTIFICATE

This is to certify that the experiments recorded in this book is the bonafide work of _ bearing Regd. No. _ _ _ _ _ _ _ _ _ _ _ _ a student of **4/4 IT-** B.Tech (Information Technology) carried out in the subject **Enterprise programming - II** Lab in the Bapatla Engineering College, Bapatla during the year ---------- of experiments recorded are **10**.

Prof **N. Sivaram Prasad**

LECTURER-IN-CHARGE HEAD OF THE DEPARTMENT

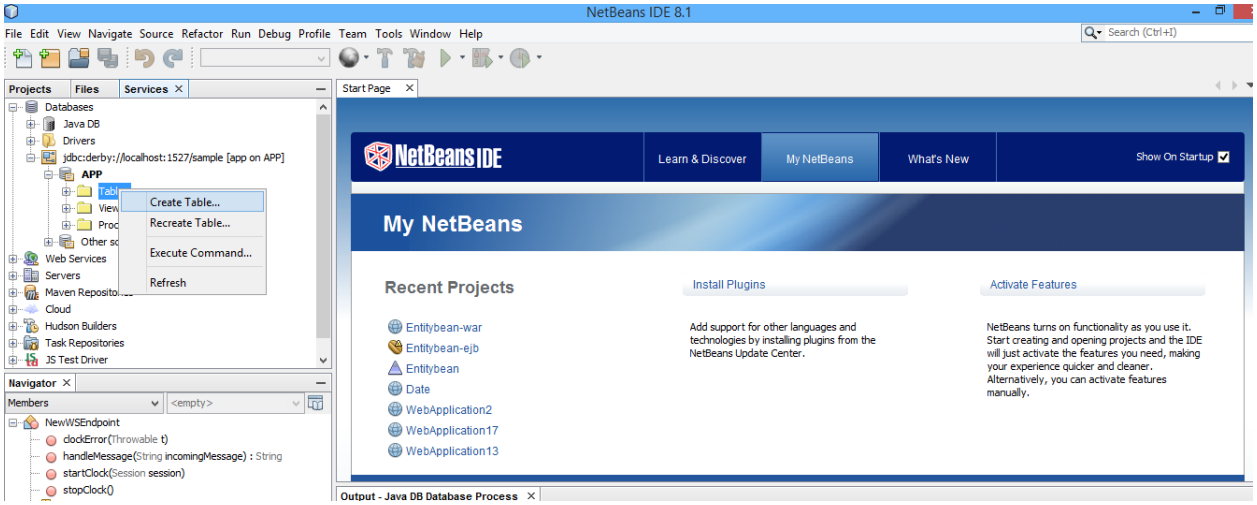
Date:

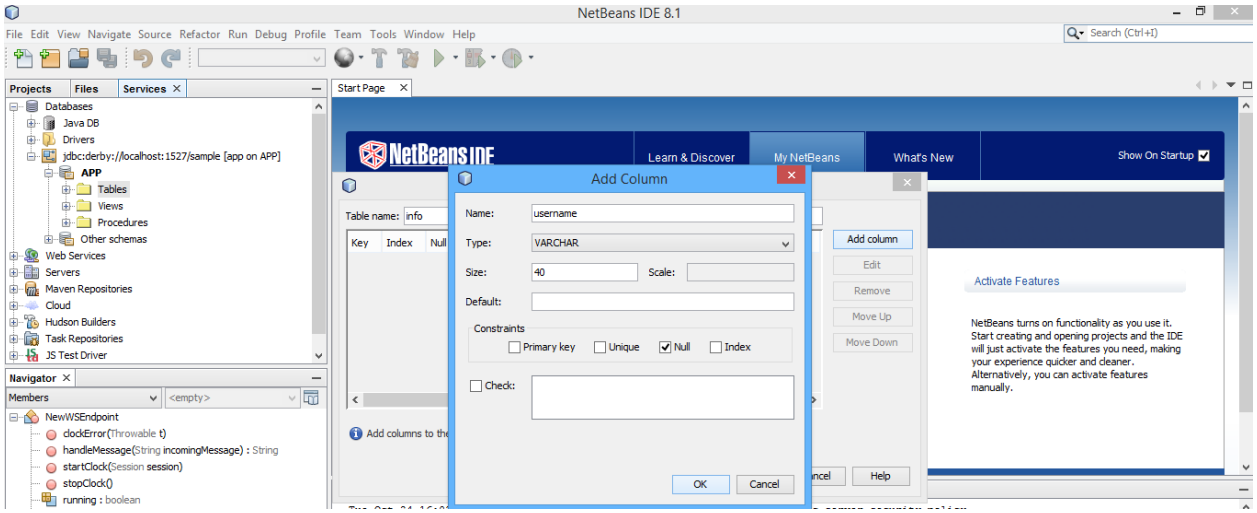
INDEX

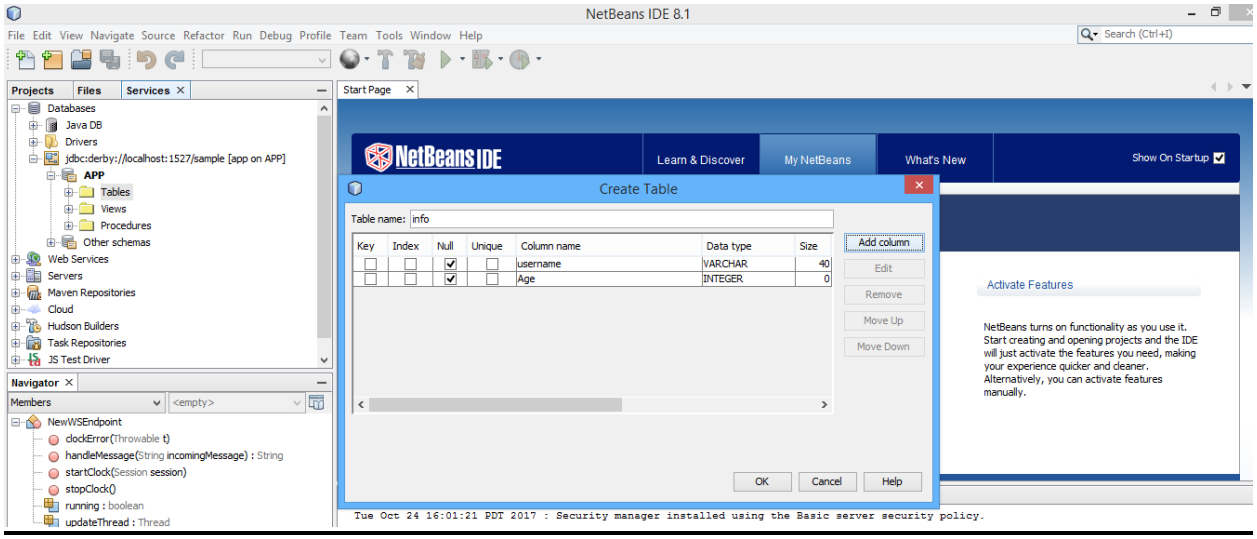
1.Write a JDBC application to implement DDL and DML commands.

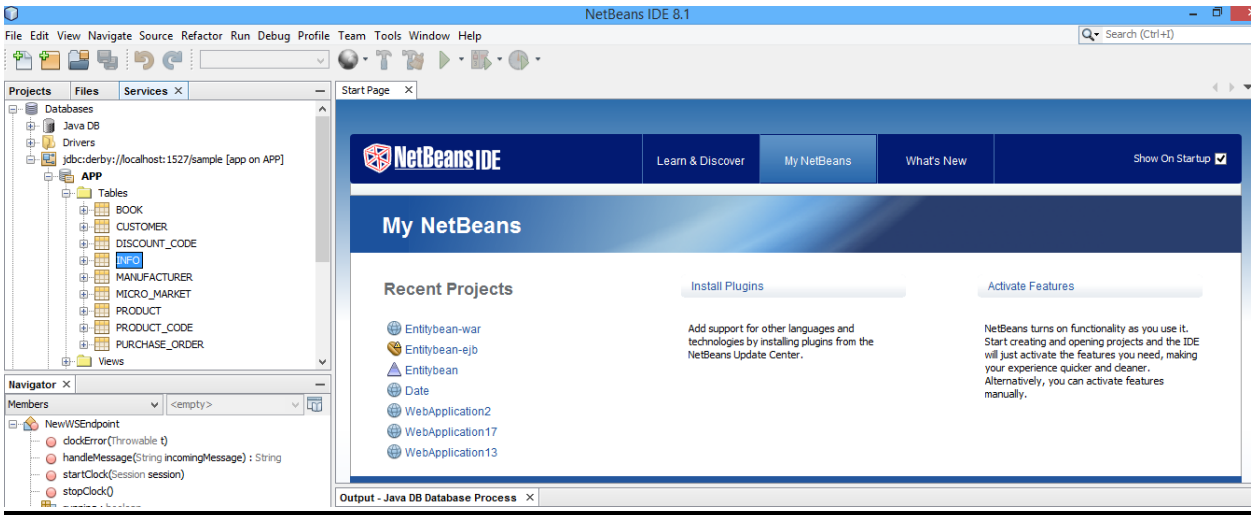
Aim:

To demonstrate JDBC connections.









Source code:

Index.html:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>JSP Page</title>
</head>
<body>
<h1>Hello!</h1>
<form action="register" method="post">
  username:<input type="text" name="t1"/></br>
  password:<input type="password" name="t2"/></br>
  emailid:<input type="text" name="t3"/></br>
  country:<input type="text" name="t4"/></br>
<center><input type="submit" value="register"/></center>
</form></body></html>
```

- Right click on project →New → Servlet

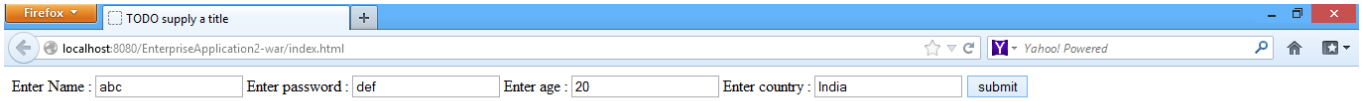
Register.java:

```
import java.io.IOException;

import java.io.PrintWriter;

import static java.lang.System.out;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class register extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException, ClassNotFoundException,
SQLException {

response.setContentType("text/html;charset=UTF-8");

 String n=request.getParameter("t1");

 String p=request.getParameter("t2");

 String e=request.getParameter("t3");

 String c=request.getParameter("t4");

 try {

PrintWriter out = response.getWriter();

Class.forName("org.apache.derby.jdbc.ClientDriver");
```

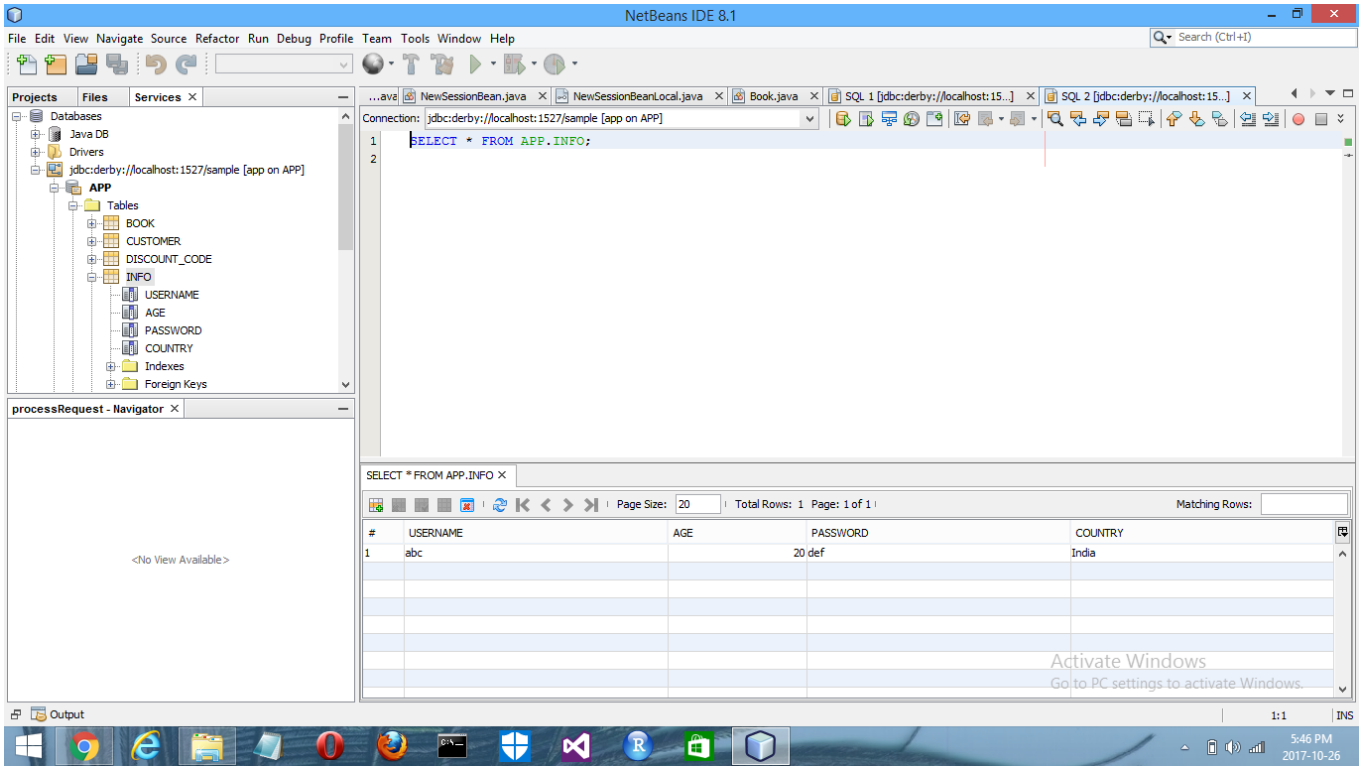
```
Connection
con=DriverManager.getConnection("jdbc:derby://localhost:1527/users","it438","834ti");
PreparedStatementps=con.prepareStatement("insert into IT438.USERS(?,?,?,?)");
ps.setString(1,n);
ps.setString(2,p);
ps.setString(3,e);
ps.setString(4,c);
inti=ps.executeUpdate();
 if(i>0)
out.println("inserted successfully");
 else
out.println("not regestered");
out.println("<!DOCTYPE html>");
out.println("<html>");
out.println("<head>");
out.println("<title>Servlet register</title>");
out.println("</head>");
out.println("<body>");
out.println("<h1>Servlet register at " + request.getContextPath() + "</h1>");
out.println("</body>");
out.println("</html>");
 } finally {
out.close();
        }
    }
 @Override
 protected void doGet(HttpServletRequest request, HttpServletResponse response)
```

```
throws ServletException, IOException {
try {
processRequest(request, response);
        } catch (ClassNotFoundException ex) {
Logger.getLogger(register.class.getName()).log(Level.SEVERE, null, ex);
        } catch (SQLException ex) {
Logger.getLogger(register.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        try {
processRequest(request, response);
        } catch (ClassNotFoundException ex) {
Logger.getLogger(register.class.getName()).log(Level.SEVERE, null, ex);
        } catch (SQLException ex) {
Logger.getLogger(register.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
@Override
    public String getServletInfo() {
        return "Short description";
    }
}
```

Output:



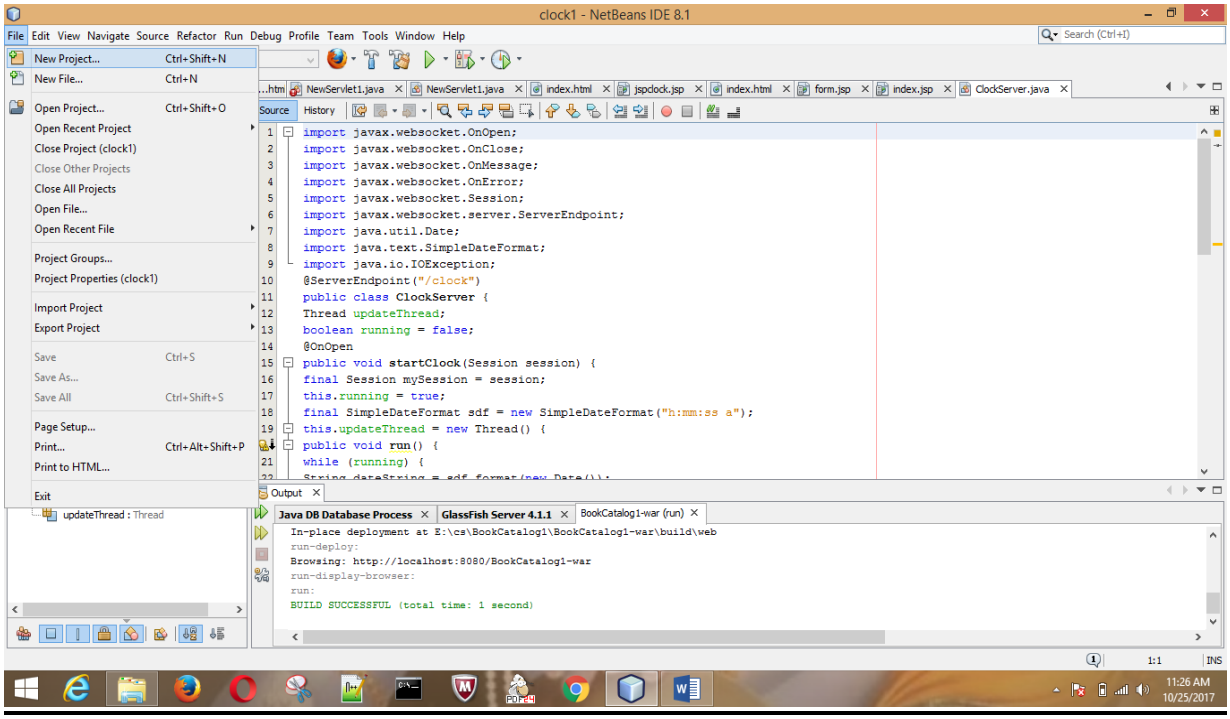
record added

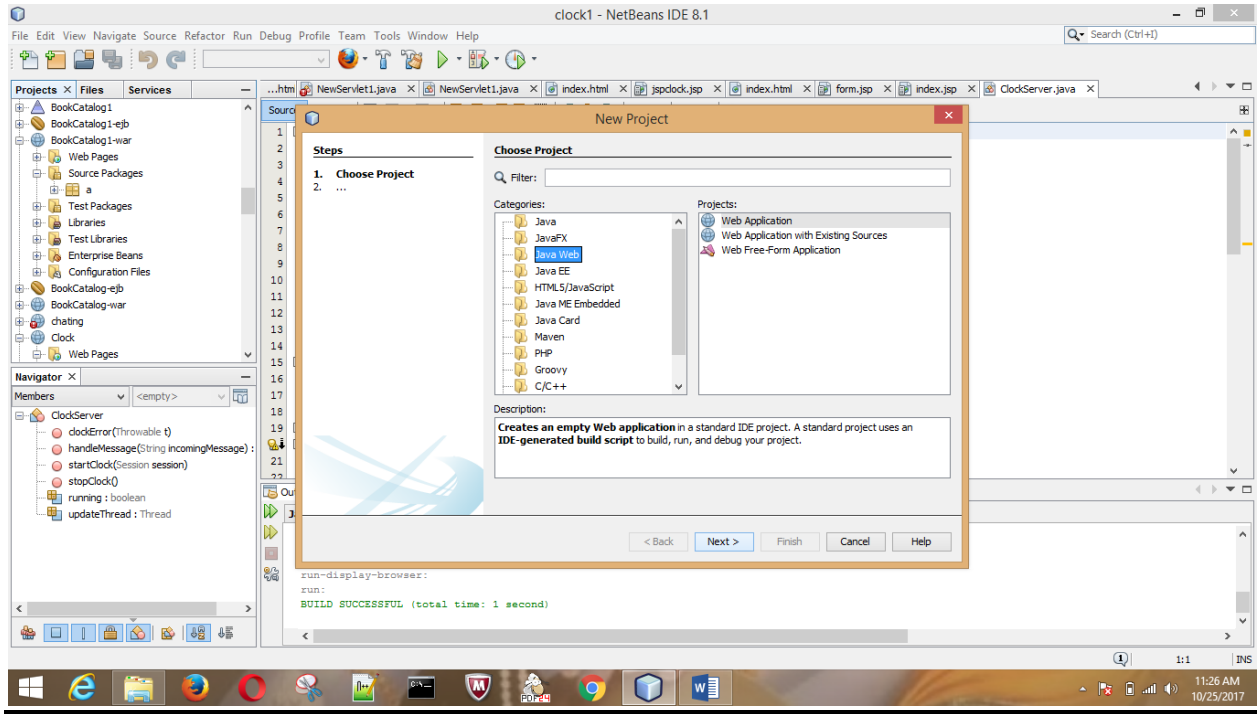

NetBeans IDE 8.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects Files Services
Databases
Java DB
Drivers
jdbc:derby://localhost:1527/sample [app on APP]
APP
Tables
BOOK
CUSTOMER
DISCOUNT_CODE
INFO
USERNAME
AGE
PASSWORD
COUNTRY
Indexes
Foreign Keys
processRequest - Navigator
<No View Available>
NewSessionBean.java
NewSessionBeanLocal.java
Book.java
SQL 1 [jdbc:derby://localhost:15...]
SQL 2 [jdbc:derby://localhost:15...]
Connection: jdbc:derby://localhost:1527/sample [app on APP]
1 SELECT * FROM APP.INFO;
2
SELECT * FROM APP.INFO
Page Size: 20 Total Rows: 1 Page: 1 of 1
Matching Rows:
USERNAME AGE PASSWORD COUNTRY
1 abc 20 def India
Activate Windows
Go to PC settings to activate Windows.
Output
1:1 INS
5:46 PM
2017-10-26

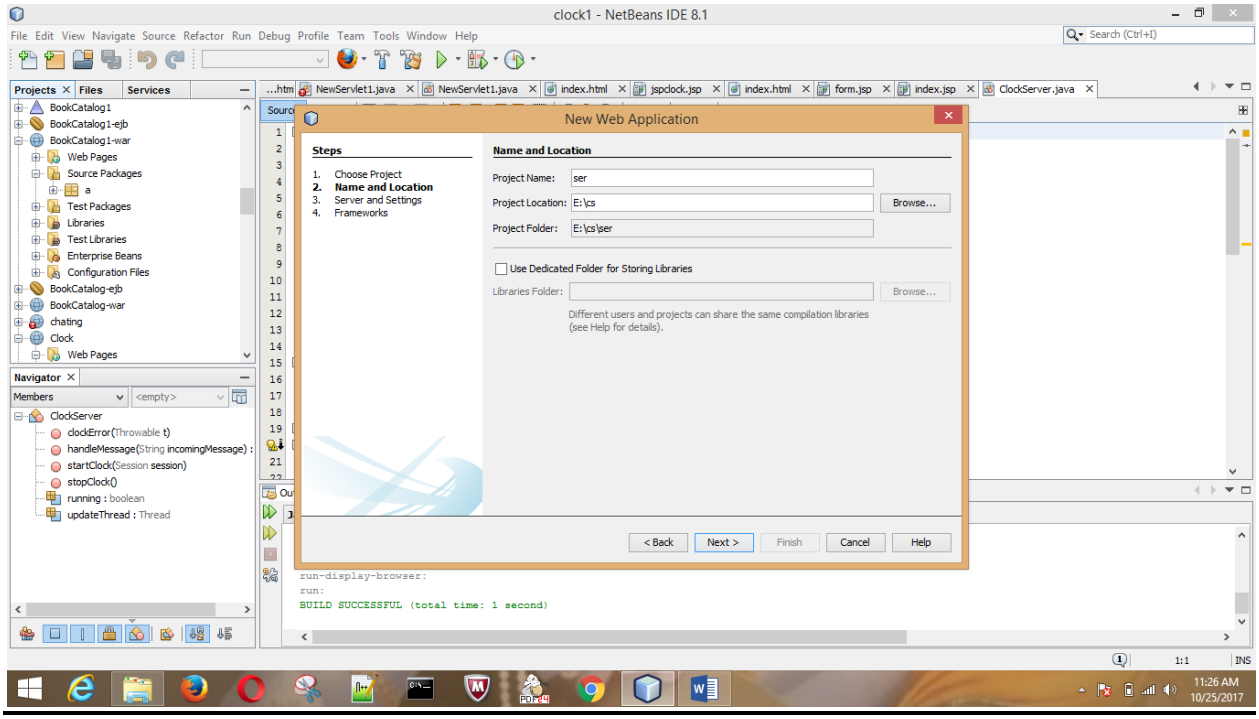
2. Write a program to demonstrate temperature using servlet.

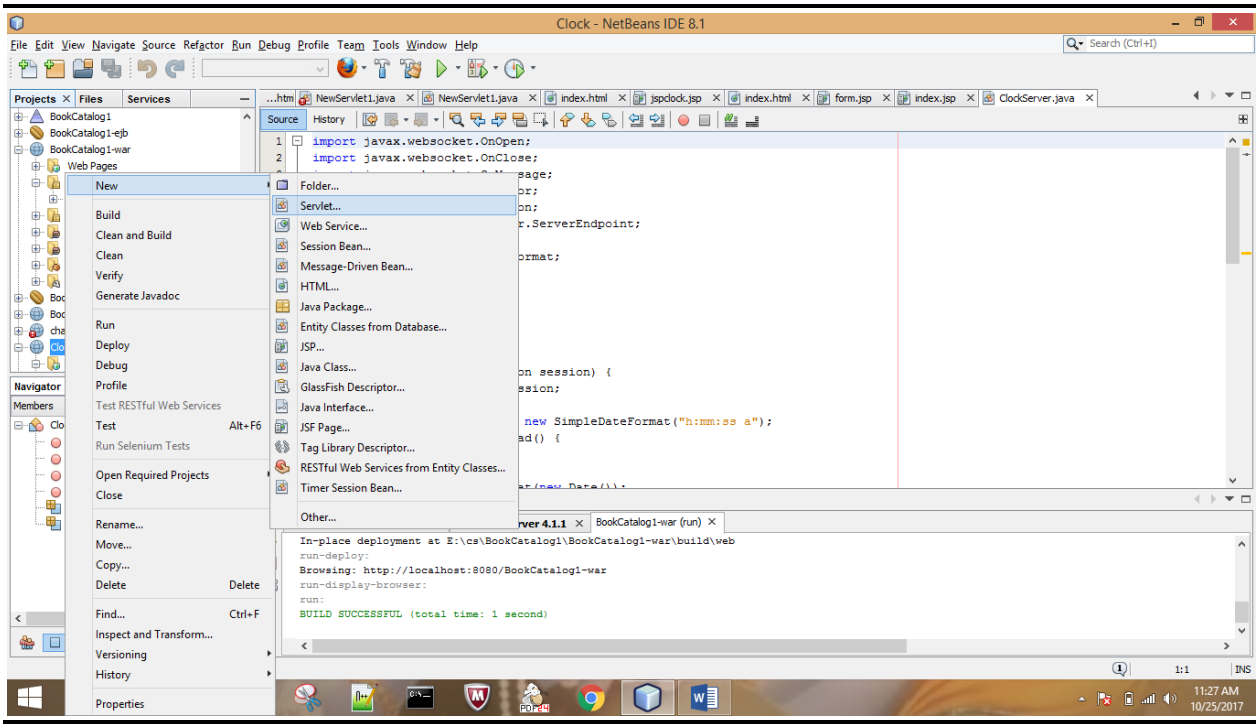
Aim:

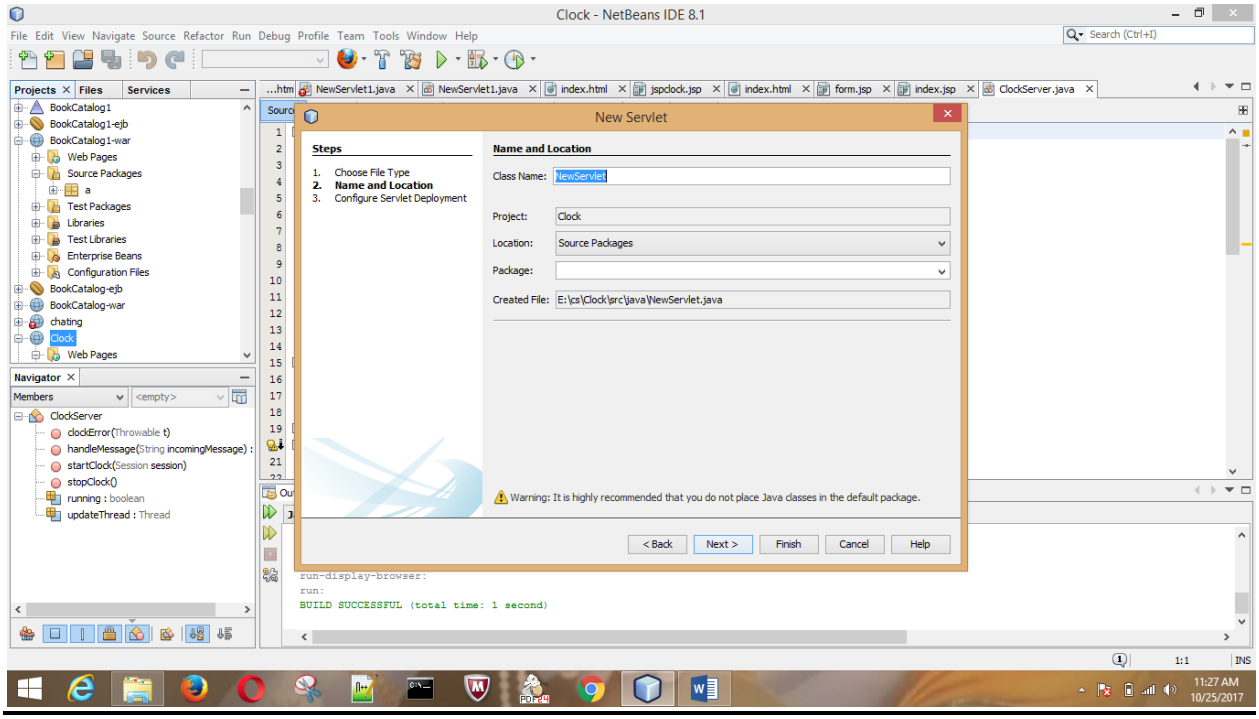
To demonstrate temperature calculation using servlet and jsp.

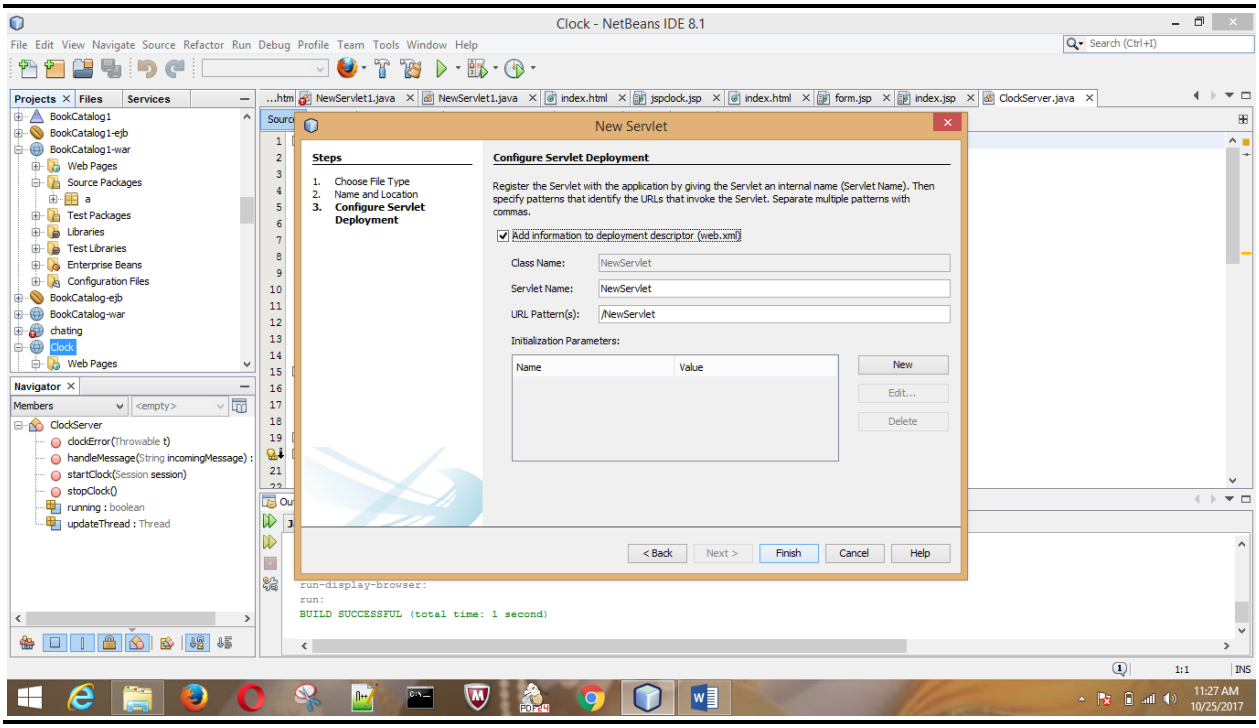





clock1 - NetBeans IDE 8.1
New Web Application
Steps
1. Choose Project
2. Name and Location
3. Server and Settings
4. Frameworks
Name and Location
Project Name: ser
Project Location: E:\cs
Project Folder: E:\cs\ser
Use Dedicated Folder for Storing Libraries
Libraries Folder:
Different users and projects can share the same compilation libraries (see Help for details).
< Back
Next >
Finish
Cancel
Help
BUILD SUCCESSFUL (total time: 1 second)


Clock - NetBeans IDE 8.1
New
Folder...
Servlet...
Web Service...
Session Bean...
Message-Driven Bean...
HTML...
Java Package...
Entity Classes from Database...
JSP...
Java Class...
GlassFish Descriptor...
Java Interface...
JSF Page...
Tag Library Descriptor...
RESTful Web Services from Entity Classes...
Timer Session Bean...
Other...
Build
Clean and Build
Clean
Verify
Generate Javadoc
Run
Deploy
Debug
Profile
Test RESTful Web Services
Test Alt+F6
Run Selenium Tests
Open Required Projects
Close
Rename...
Move...
Copy...
Delete Delete
Find... Ctrl+F
Inspect and Transform...
Versioning
History
Properties
In-place deployment at E:\cs\BookCatalog1\BookCatalog1-war\build\web
run-deploy:
Browsing: http://localhost:8080/BookCatalog1-war
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 1 second)





Source code:

NewServlet.java:

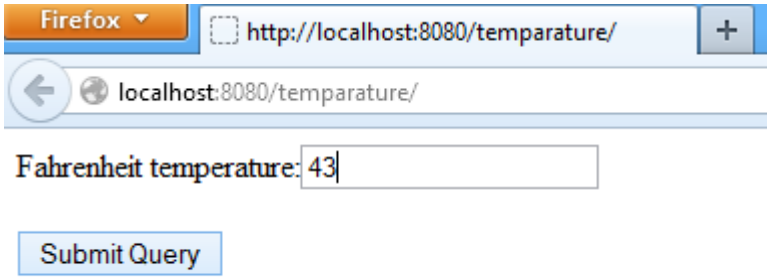
```
import java.io.IOException;

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;
public class NewServlet extends HttpServlet
{
public  voiddoGet(HttpServletRequest request, HttpServletResponse response)  throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
PrintWriter out = response.getWriter();
String f=request.getParameter("t1");
inti=Integer.parseInt(f);
double d=(i-32)*5/9;
try {
out.println("<h1>celciustemparature is  "+d  + "</h1>");
} finally {
out.close();
}
}
```

Temp.html:

```
<html>
<head>
<title></title></head>
<body>
<form action="NewServlet" method="get">
 Fahrenheit temperature:<input type="text" name="t1"><br/>
<input type="submit"/>
</form>
</body></html>
```

Output:

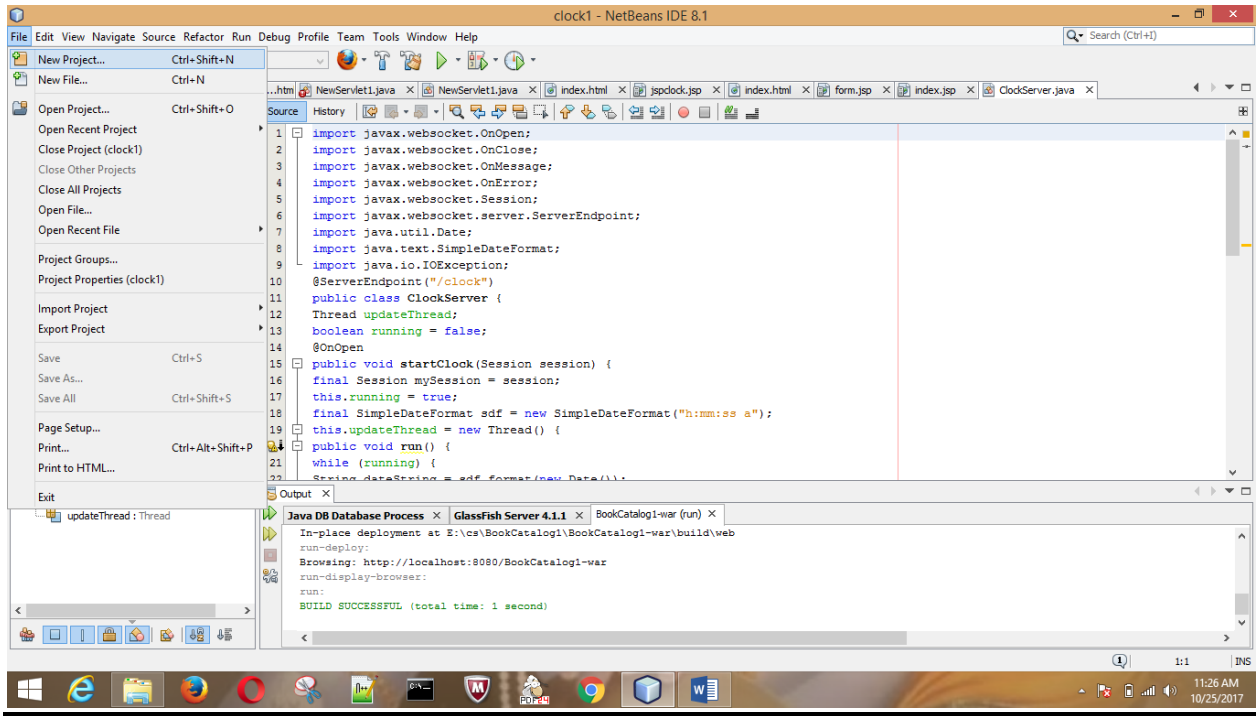

Firefox
http://localhost:8080/temparature/
localhost:8080/temparature/
Fahrenheit temperature: 43
Submit Query

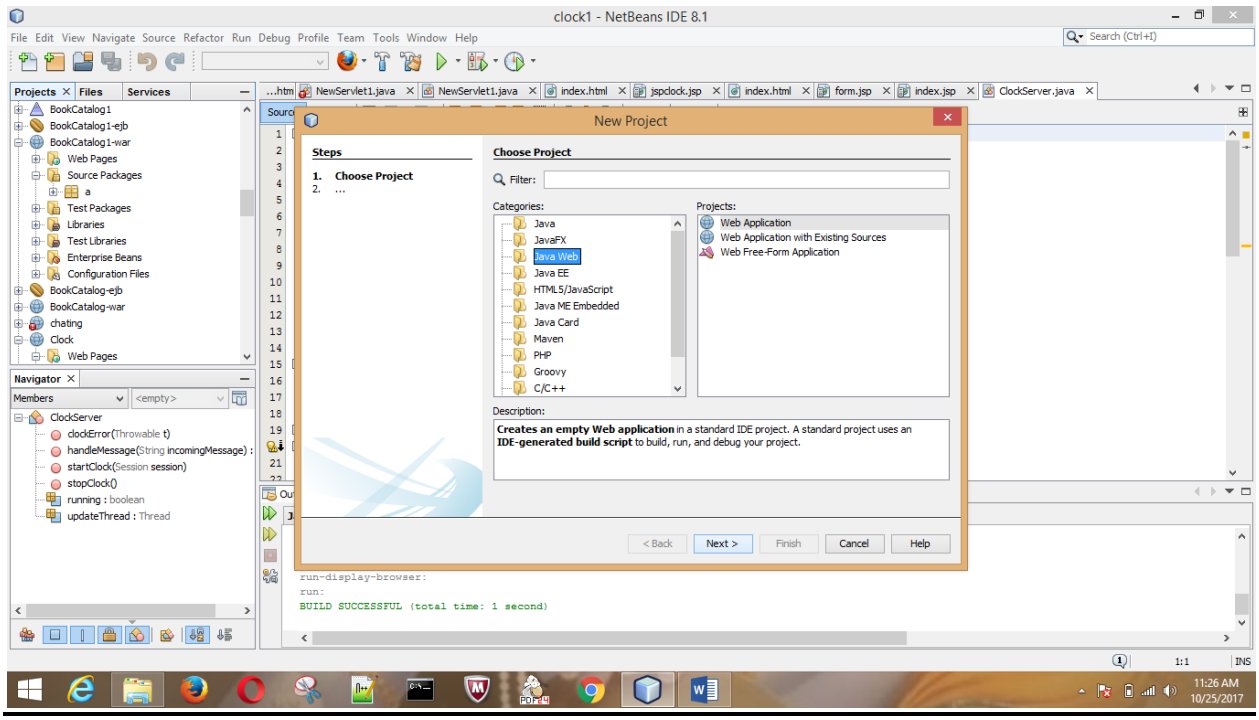

Firefox
Servlet NewServlet
localhost:8080/temparature/NewServlet?t1=43
celcius temparature is 6.0

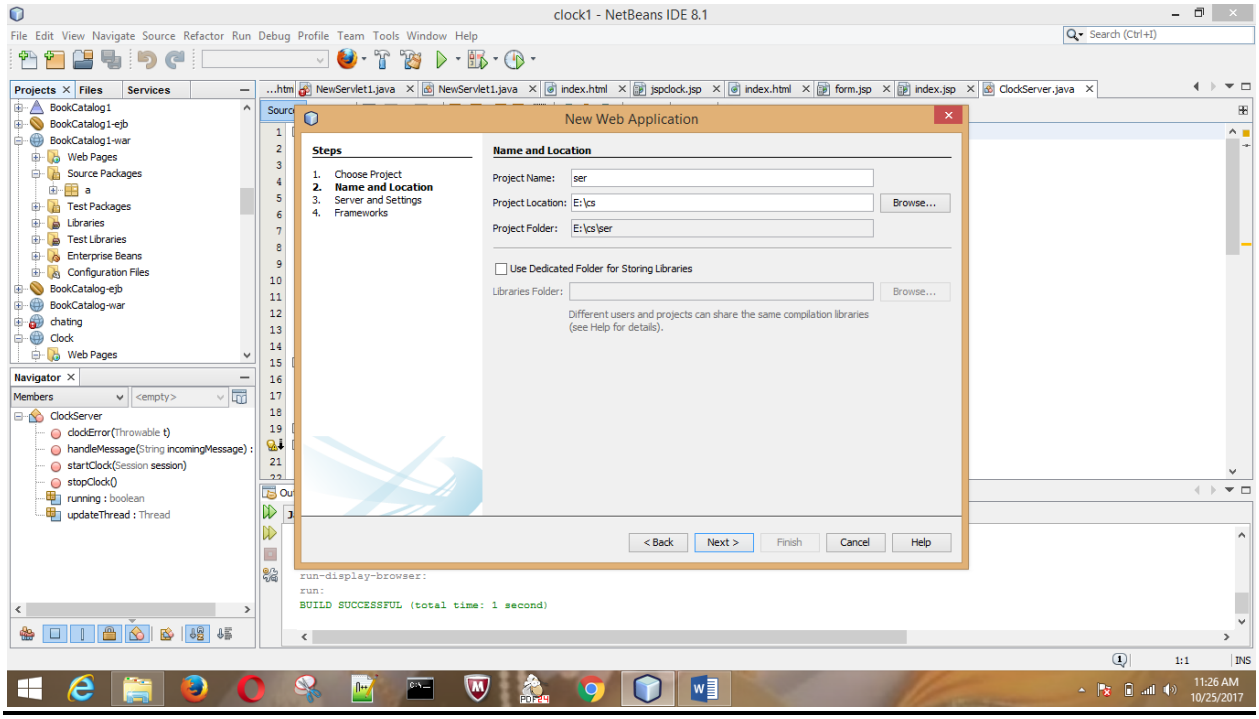
3.a) Write a program to demonstrate Cookies using Servlet.

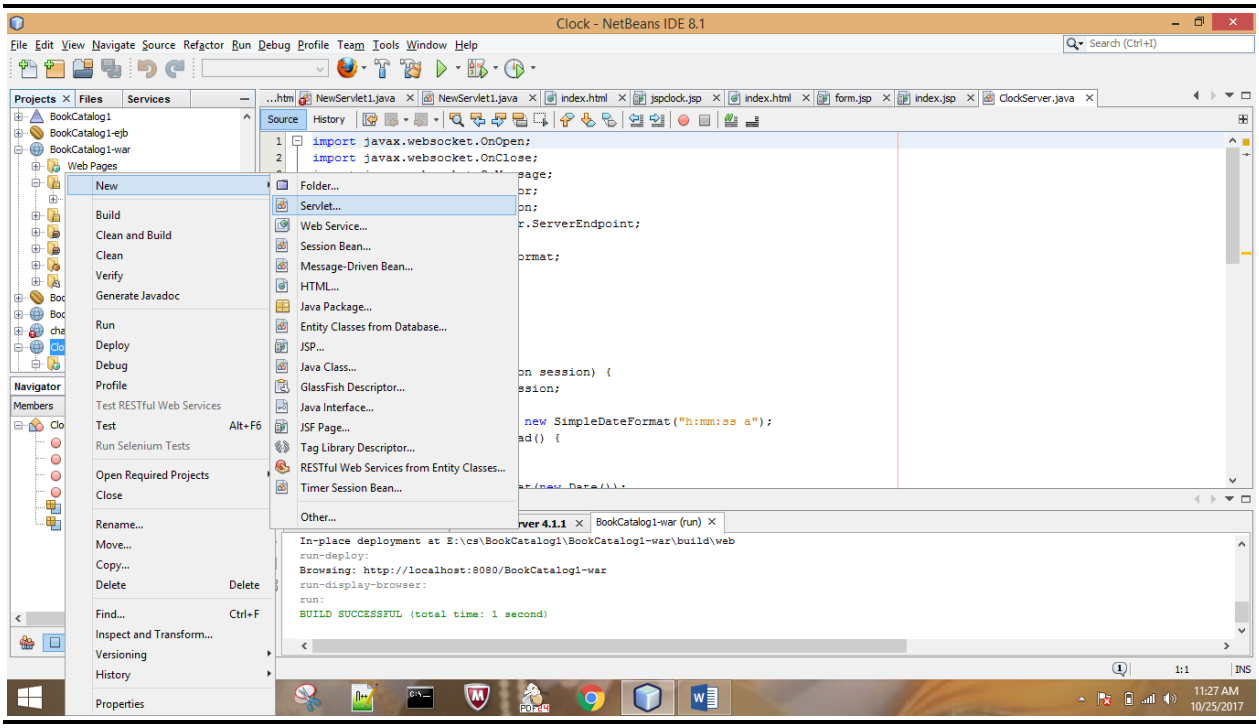
Aim:

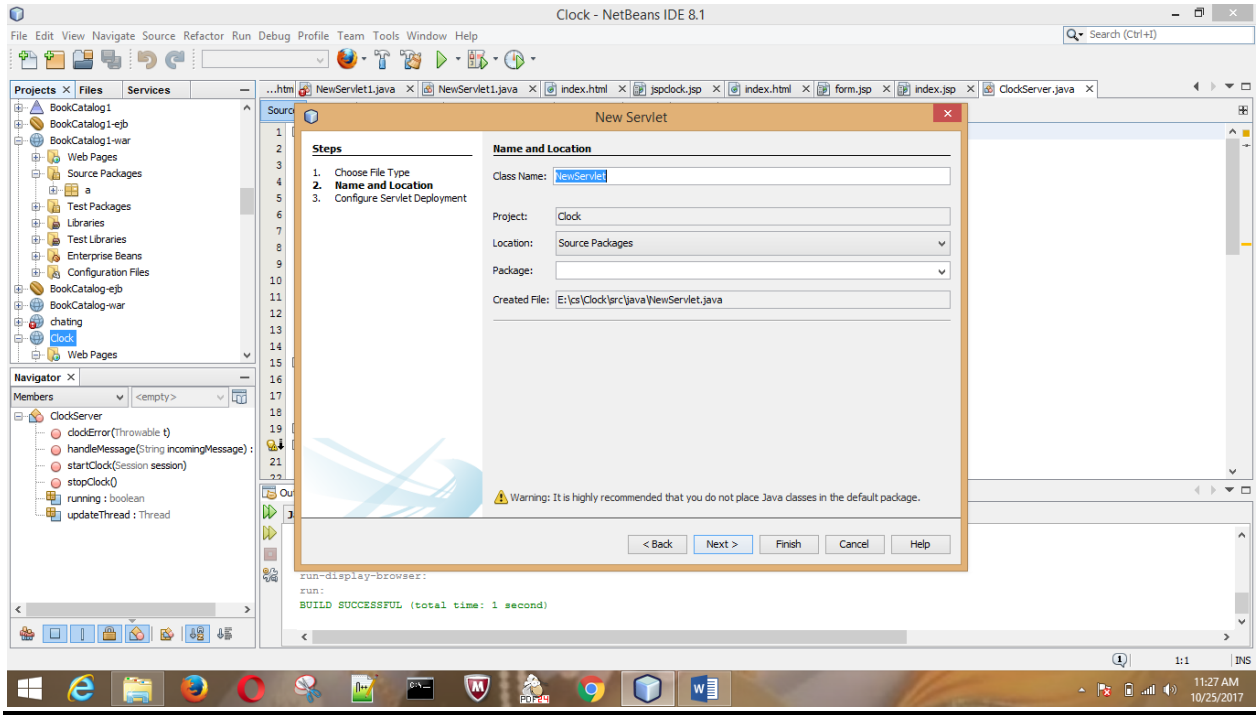
To demonstrate cookie using servlet and jsp.

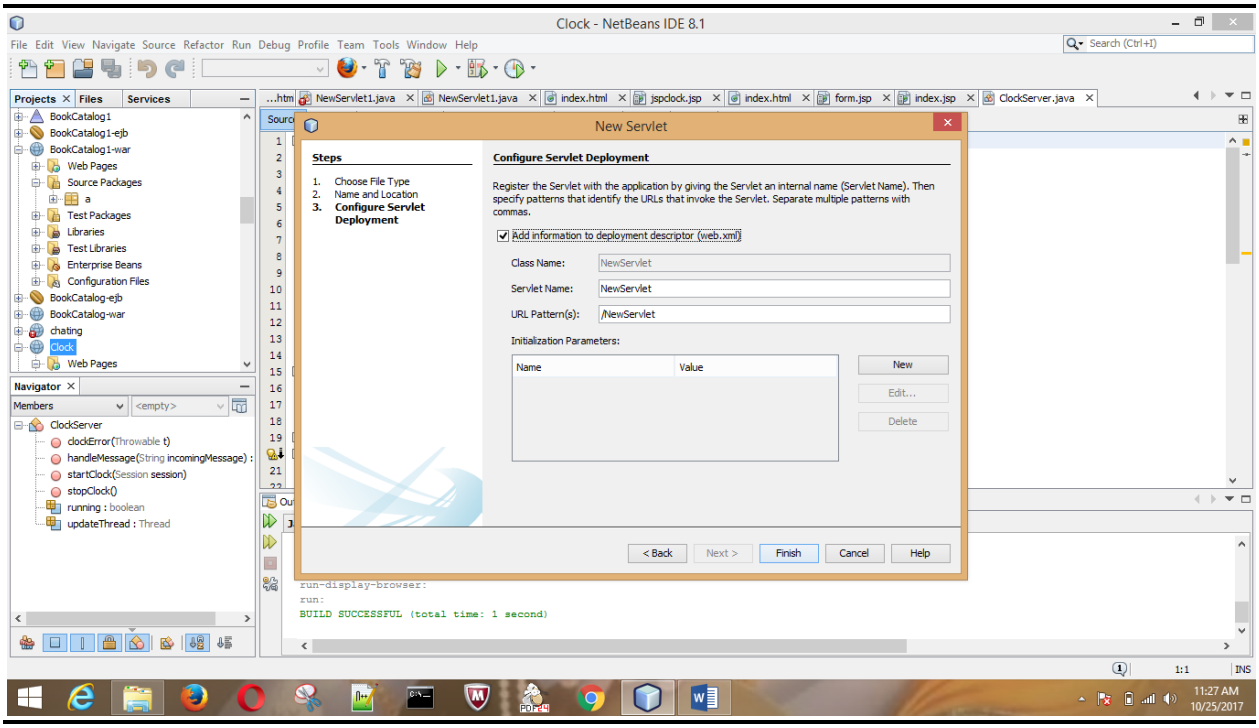





clock1 - NetBeans IDE 8.1
New Web Application
Steps
1. Choose Project
2. Name and Location
3. Server and Settings
4. Frameworks
Name and Location
Project Name: ser
Project Location: E:\cs
Project Folder: E:\cs\ser
Browse...
Use Dedicated Folder for Storing Libraries
Libraries Folder:
Different users and projects can share the same compilation libraries (see Help for details).
< Back
Next >
Finish
Cancel
Help
BUILD SUCCESSFUL (total time: 1 second)


Clock - NetBeans IDE 8.1
New
Folder...
Servlet...
Web Service...
Session Bean...
Message-Driven Bean...
HTML...
Java Package...
Entity Classes from Database...
JSP...
Java Class...
GlassFish Descriptor...
Java Interface...
JSF Page...
Tag Library Descriptor...
RESTful Web Services from Entity Classes...
Timer Session Bean...
Other...
Build
Clean and Build
Clean
Verify
Generate Javadoc
Run
Deploy
Debug
Profile
Test RESTful Web Services
Test Alt+F6
Run Selenium Tests
Open Required Projects
Close
Rename...
Move...
Copy...
Delete Delete
Find... Ctrl+F
Inspect and Transform...
Versioning
History
Properties
In-place deployment at E:\cs\BookCatalog1\BookCatalog1-war\build\web
run-deploy:
Browsing: http://localhost:8080/BookCatalog1-war
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 1 second)





Source code:

Index.html:

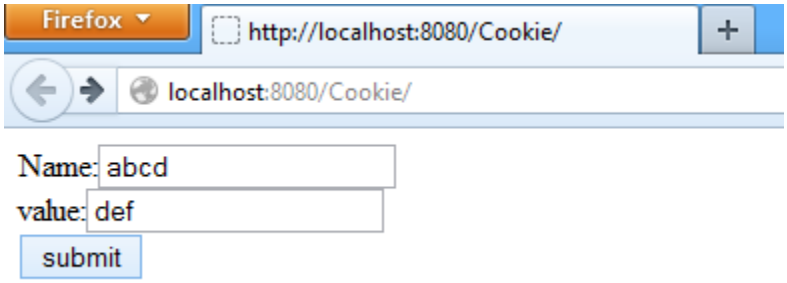
```
<html><head><title> Cookie</title></head>

<body bgcolor="red"><form action="Cookie" method=POST>

Name:  <input type="text" length=20 name="cookiename"><br  />

Value: <input type="text" length=20 name="cookievalue"><br />

<input type="submit"></form></body></html>
```

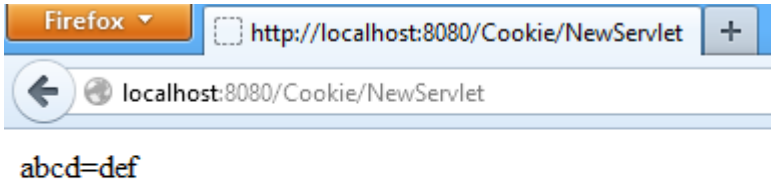
Right click on project → New → Servlet

Cookie.java:

```
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class NewServlet extends HttpServlet {
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
try (PrintWriter out = response.getWriter()) {
String Name=request.getParameter("cookieName");
if(Name!=null&&Name.length()>0){
String Value=request.getParameter("cookieValue");
Cookie e=new Cookie(Name,Value);
response.addCookie(e);
}Cookie Cookies[];
Cookies = request.getCookies();
for (Cookie e : Cookies) {
String a=e.getName();
String b=e.getValue();
out.println(a+"="+b);
}}
}
```

Output:

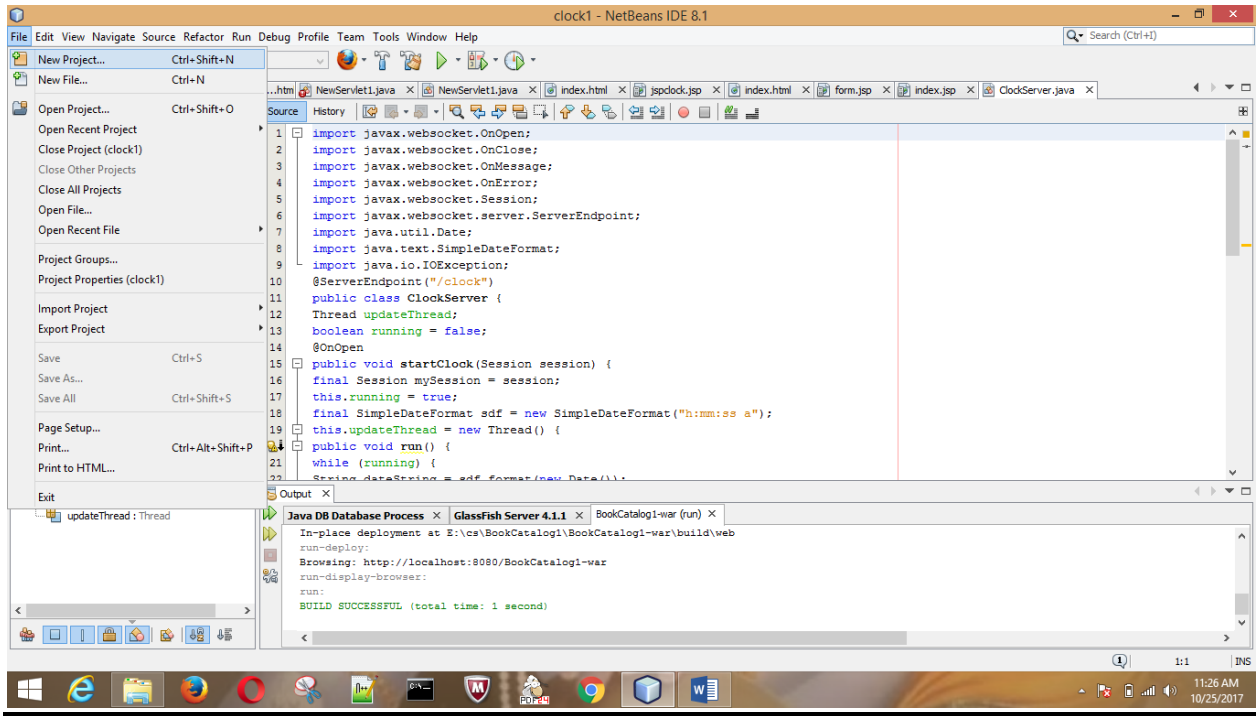

Firefox
http://localhost:8080/Cookie/
localhost:8080/Cookie/
Name: abcd
value: def
submit

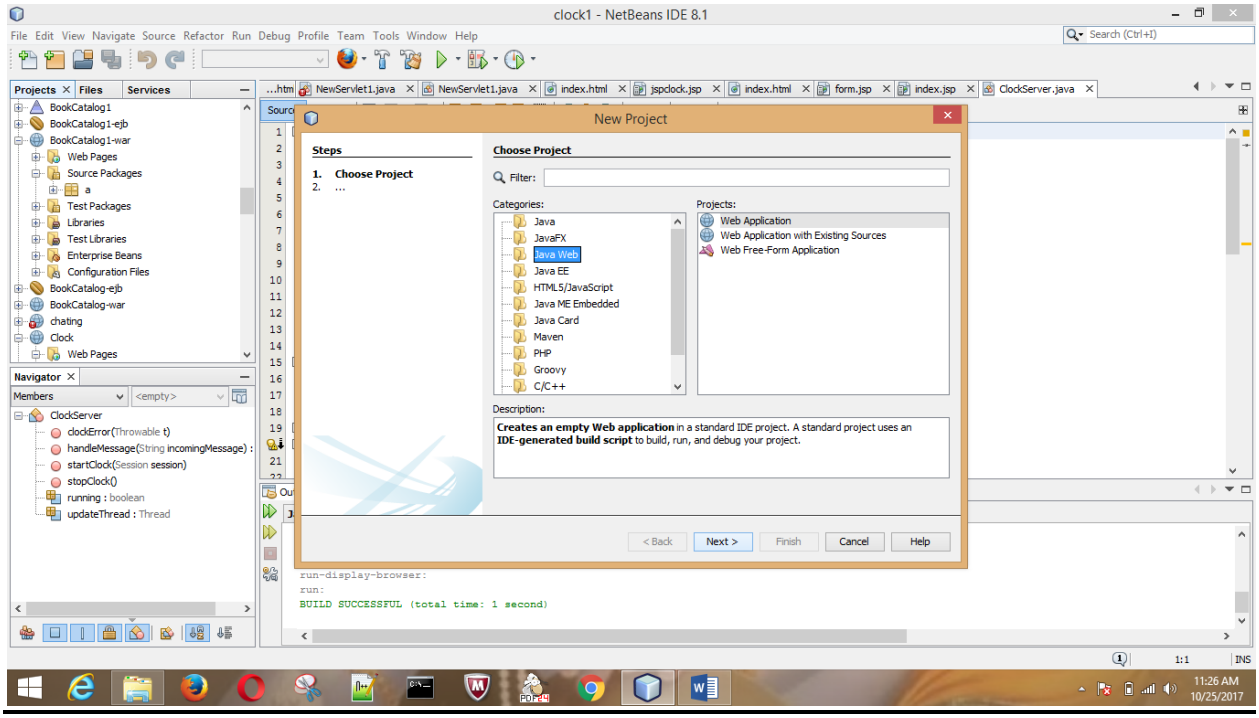

Firefox
http://localhost:8080/Cookie/NewServlet
localhost:8080/Cookie/NewServlet
abcd=def

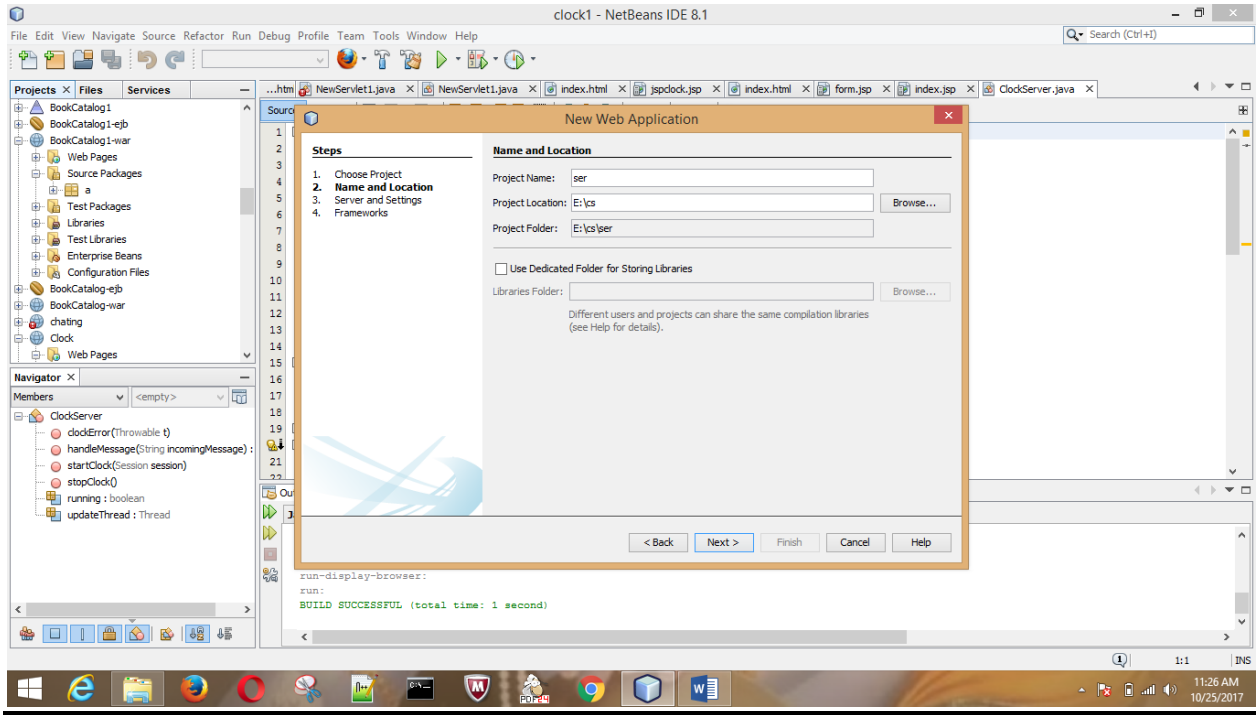
3.b) Write a program to demonstrate session using servlet.

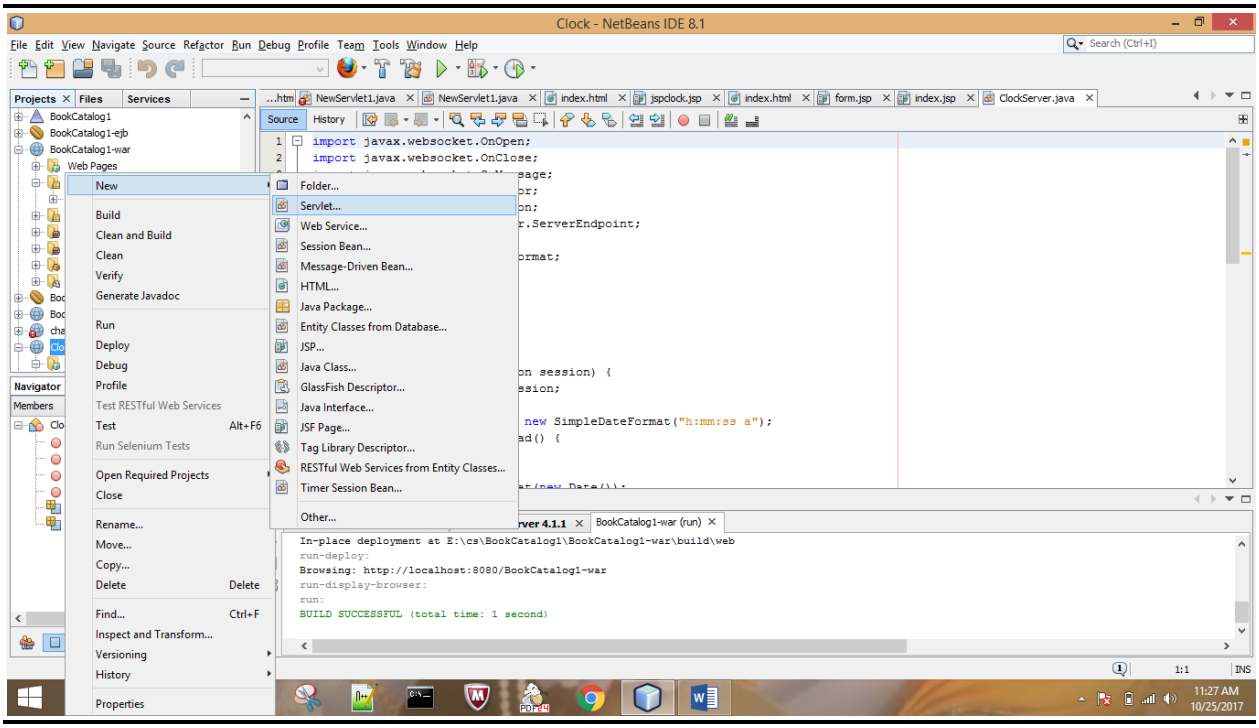
Aim:

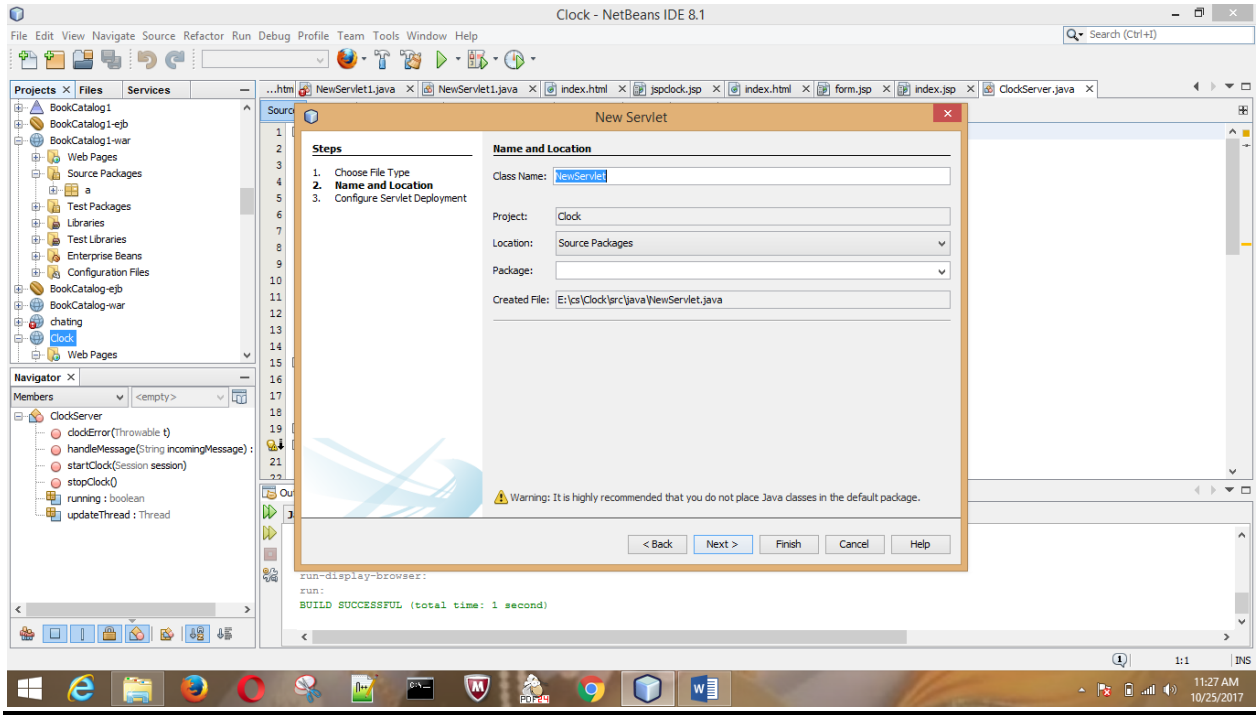
To demonstrate session using servlets.

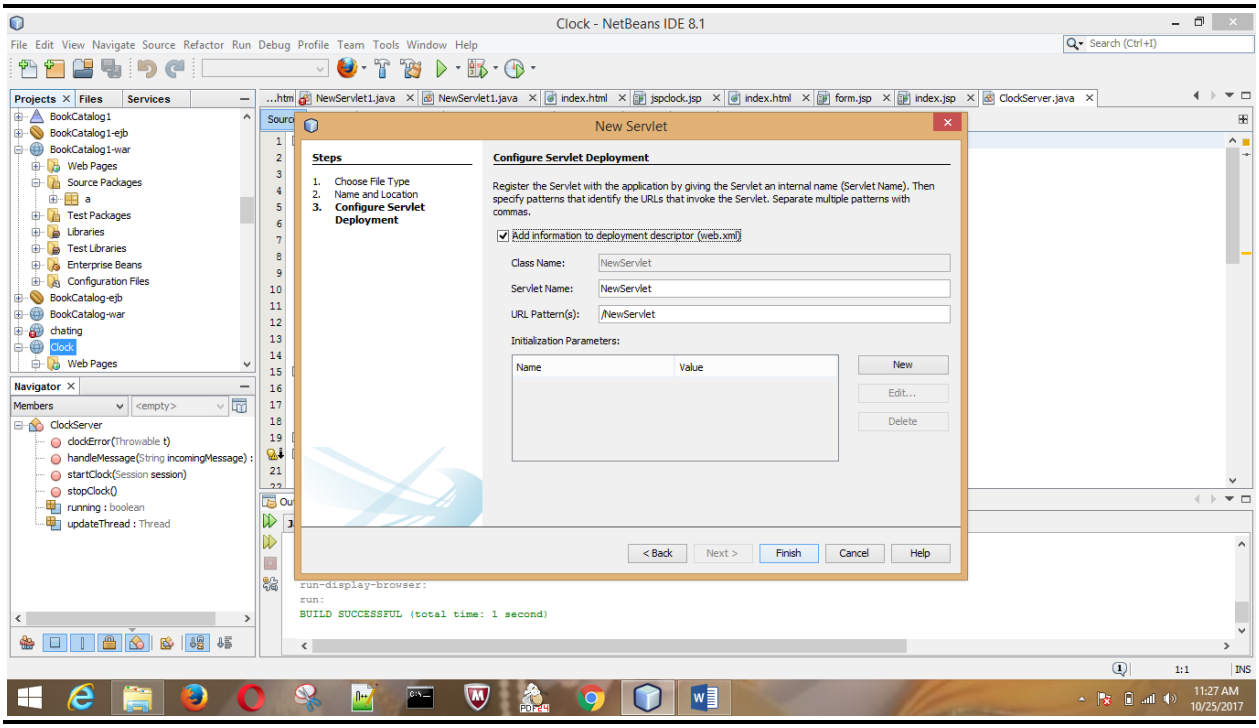





clock1 - NetBeans IDE 8.1
New Web Application
Steps
1. Choose Project
2. Name and Location
3. Server and Settings
4. Frameworks
Name and Location
Project Name: ser
Project Location: E:\cs
Project Folder: E:\cs\ser
Browse...
Use Dedicated Folder for Storing Libraries
Libraries Folder:
Different users and projects can share the same compilation libraries (see Help for details).
< Back
Next >
Finish
Cancel
Help
BUILD SUCCESSFUL (total time: 1 second)


Clock - NetBeans IDE 8.1
New
Folder...
Servlet...
Web Service...
Session Bean...
Message-Driven Bean...
HTML...
Java Package...
Entity Classes from Database...
JSP...
Java Class...
GlassFish Descriptor...
Java Interface...
JSF Page...
Tag Library Descriptor...
RESTful Web Services from Entity Classes...
Timer Session Bean...
Other...
Build
Clean and Build
Clean
Verify
Generate Javadoc
Run
Deploy
Debug
Profile
Test RESTful Web Services
Test Alt+F6
Run Selenium Tests
Open Required Projects
Close
Rename...
Move...
Copy...
Delete Delete
Find... Ctrl+F
Inspect and Transform...
Versioning
History
Properties
In-place deployment at E:\cs\BookCatalog1\BookCatalog1-war\build\web
run-deploy:
Browsing: http://localhost:8080/BookCatalog1-war
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 1 second)





Source code:

Index.html:

```
<html>
<head>
<title>Session</title>
</head>
<body>
<h1> Anexample for session attributes </h1>
```

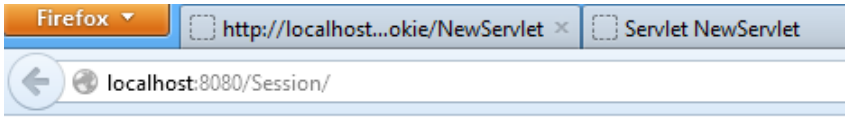
```
<form action="Session" method=GET>
Name of Session Attribute:<input type=text size=20 name=dataname /><br>
Value of Session Attribute:<input type=text size=20 name=datavalue /><br>
<input type=submit />
</form>
</body>
</html>
```

Session.java:

```
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Date;
import java.util.Enumeration;
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 NewServlet extends HttpServlet
{
protected void  doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException
{
response.setContentType("text/html;charset=UTF-8");
try (PrintWriter out = response.getWriter()) {
HttpSession session=request.getSession(true);
Date created=new Date(session.getCreationTime());
Date accessed=new Date(session.getLastAccessedTime());
```

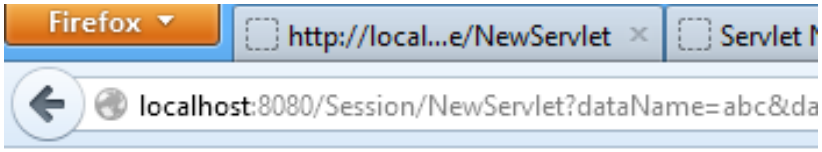
```
out.println("ID"+session.getId()+"<br/>");
out.println("created"+created+"<br/>");
outprintln("LastAccessed"+accessed+"<br/>");
String dataname=request.getParameter("dataname");
if(dataname!=null&&dataname.length()>0)
{
String datavalue=request.getParameter("datavalue");
session.setAttribute(dataname,datavalue);
}
Enumeration e=session.getAttributeNames();
while(e.hasMoreElements()){
String name=(String)e.nextElement();
String value=session.getAttribute(name).toString();
out.println("<br/>"+name+"="+value);
}
}
}
```

Output:



an example for session attributes

Name of session attribute: abc
Value of session attribute: def
submit

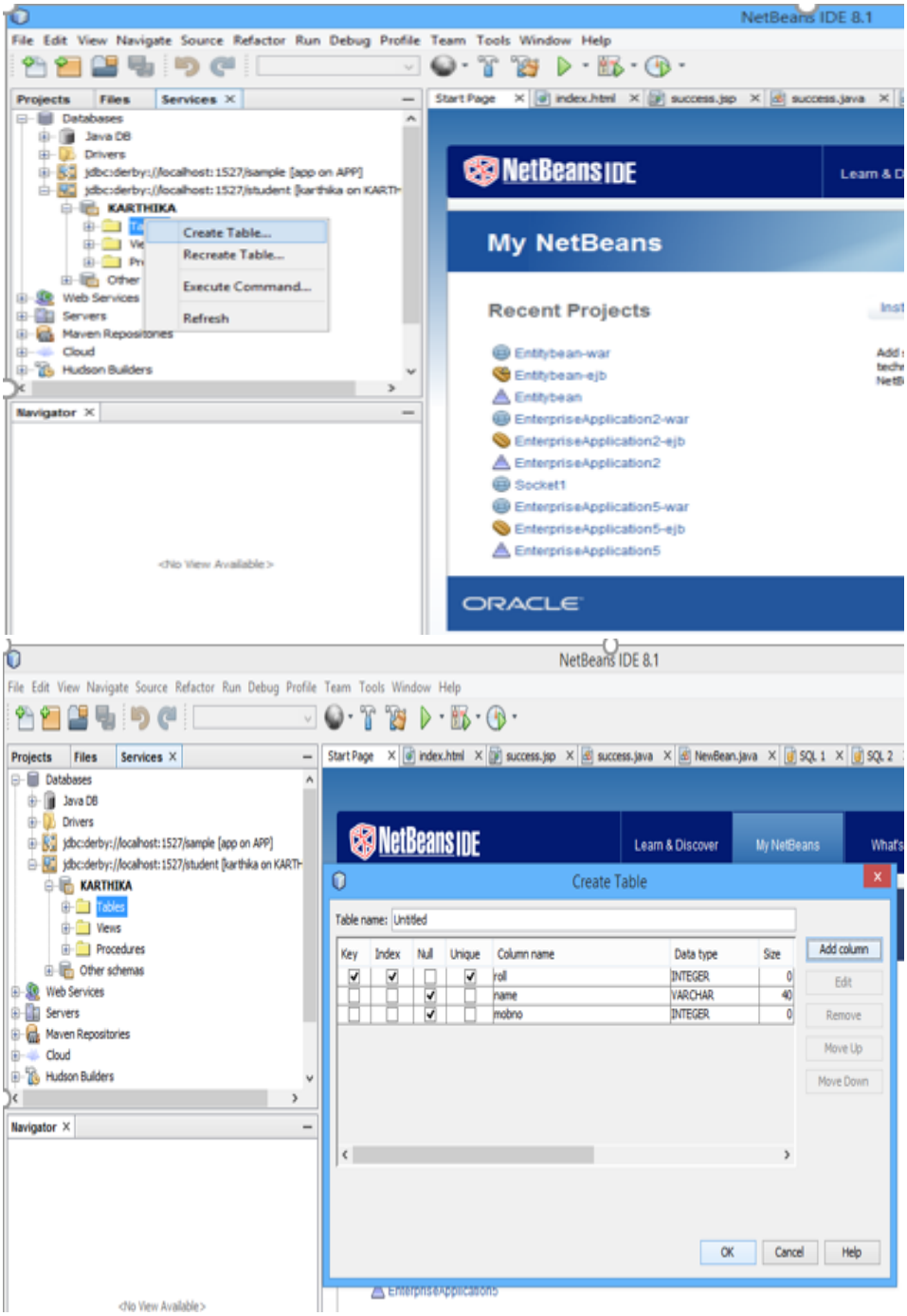


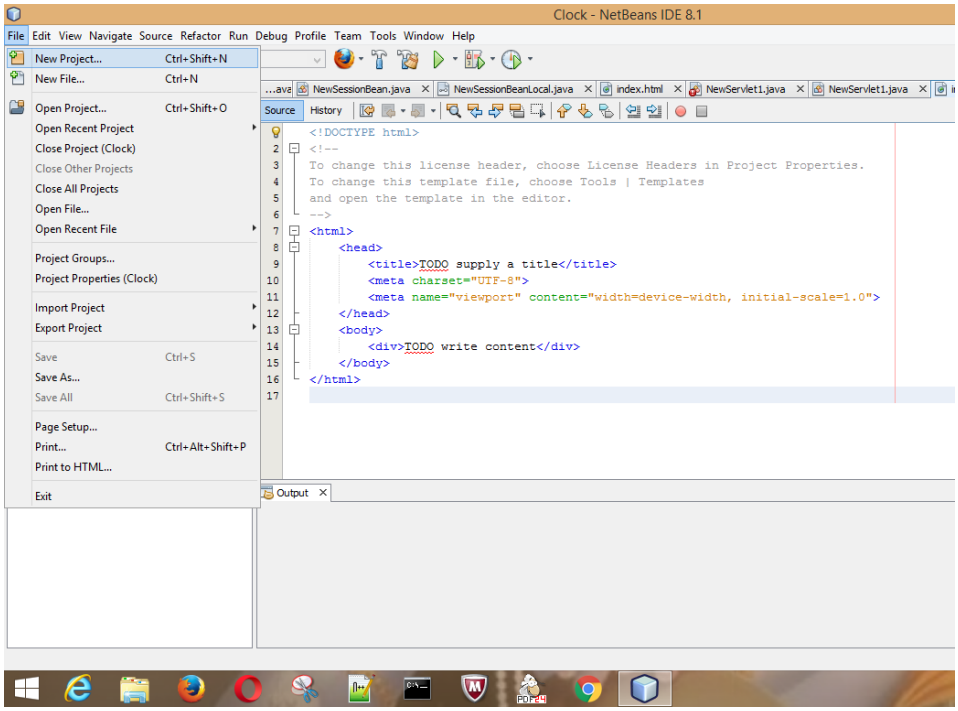
ID63d8534c0987d4d48f38ae2e99b9
createdThu Oct 19 13:05:22 PDT 2017
LastAccessedThu Oct 19 13:05:22 PDT 2017

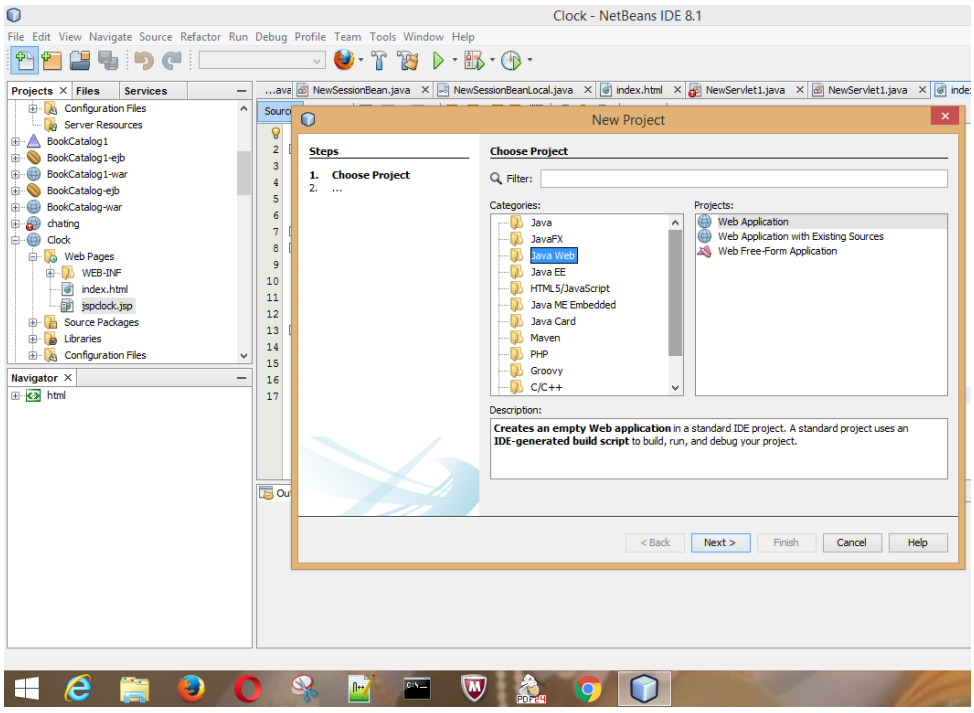
4. Write a java web application to integrate JSP &Servlets.

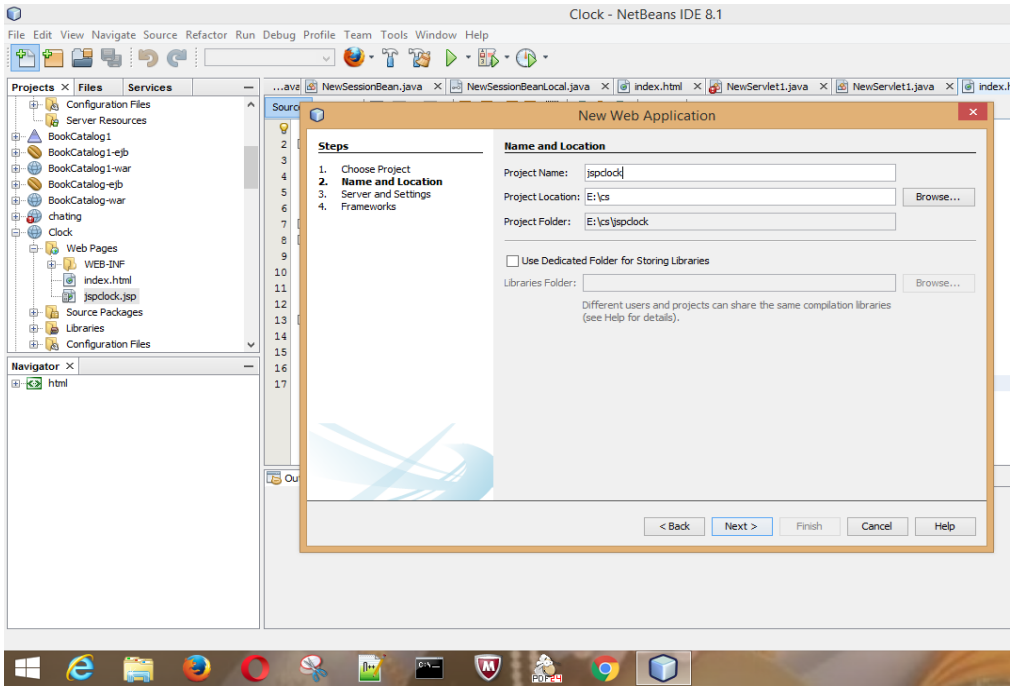
Aim:

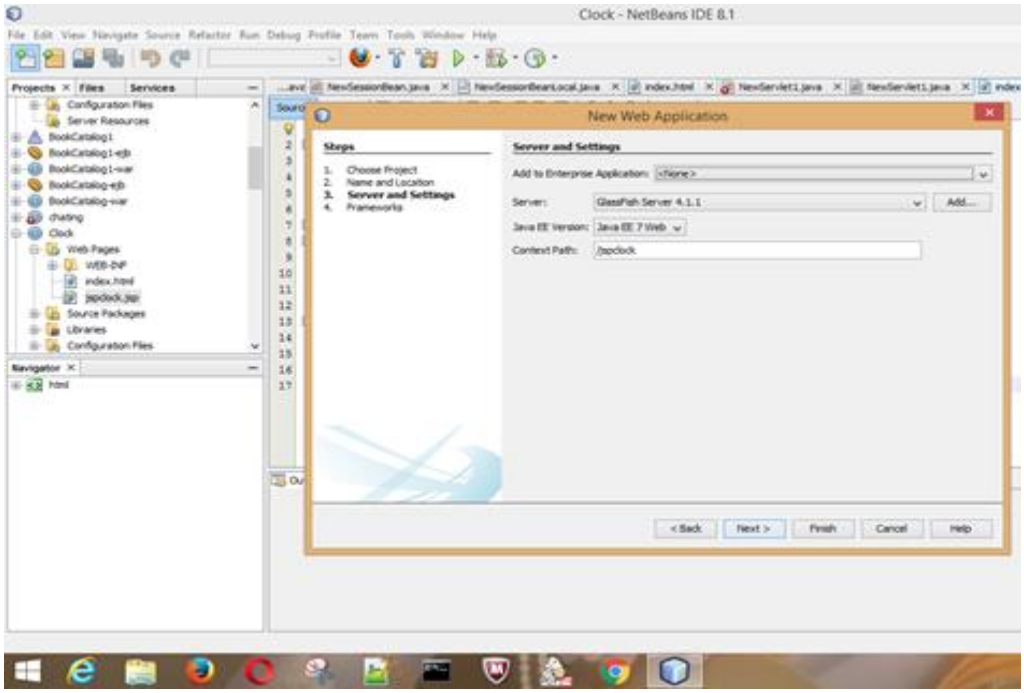
To demonstrate student details using JSP & Servlets.




Clock - NetBeans IDE 8.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
New Project... Ctrl+Shift+N
New File... Ctrl+N
Open Project... Ctrl+Shift+O
Open Recent Project
Close Project (Clock)
Close Other Projects
Close All Projects
Open File...
Open Recent File
Project Groups...
Project Properties (Clock)
Import Project
Export Project
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
Page Setup...
Print... Ctrl+Alt+Shift+P
Print to HTML...
Exit
NewSessionBean.java
NewSessionBeanLocal.java
index.html
NewServlet1.java
Source History
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<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>
TODO write content
</body>
</html>
Output


Clock - NetBeans IDE 8.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Files Services
Configuration Files
Server Resources
BookCatalog1
BookCatalog1-ejb
BookCatalog1-war
BookCatalog-ejb
BookCatalog-war
chating
Clock
Web Pages
WEB-INF
index.html
jspclock.jsp
Source Packages
Libraries
Configuration Files
Navigator
html
New Project
Steps
1. Choose Project
2. ...
Choose Project
Filter:
Categories:
Java
JavaFX
Java Web
Java EE
HTML5/JavaScript
Java ME Embedded
Java Card
Maven
PHP
Groovy
C/C++
Projects:
Web Application
Web Application with Existing Sources
Web Free-Form Application
Description:
Creates an empty Web application in a standard IDE project. A standard project uses an IDE-generated build script to build, run, and debug your project.
< Back
Next >
Finish
Cancel
Help





Source code:

```
<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="NewServlet">
```

```
        Enter roll :<input type="text" name="roll"/>
        Enter name :<input type="text" name="name"/>
        Enter mobno :<input type="text" name="mobno"/>
        <input type="submit" />
        </form>
    </body>
</html>
```

NewServlet.java:

```
import abc.NewSessionBeanLocal;
import java.io.IOException;
import java.io.PrintWriter;
import javax.ejb.EJB;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class NewServlet extends HttpServlet {
    @EJB
    private NewSessionBeanLocal n;
   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 NewServlet</title>");
            out.println("</head>");
```

```
            out.println("<body>");
            int roll= Integer.parseInt(request.getParameter("roll"));
            String name=request.getParameter("name");
            String mobno=request.getParameter("mobno");
           boolean k=  n.insert(roll, name, mobno);
           if (k==true)
               request.getRequestDispatcher("/login.jsp").include(request, response);
            else
              request.getRequestDispatcher("/loginfail.jsp").include(request, response);
             }
    }
    @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";
    }// </editor-fold>
}
```

login.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>
        <%
            out.println("<h1>new record had been inserted</h1>");
        %>
    </body>
</html>
```

loginfail.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>
        <%
         out.println("<h1>insertion failed !! Try again !!</h1>");
         %>
    </body>
</html>
```

NewSessionBeanLocal.java:

```
package abc;
import javax.ejb.Local;
@Local
```

```
public interface NewSessionBeanLocal {
    public boolean insert(int roll,String name,String mob);
}
```

NewSessionBean.java:

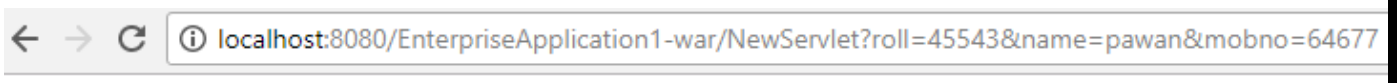
```
package abc;
import static java.lang.Character.UnicodeBlock.forName;
import javax.ejb.Stateless;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
@Stateless
public class NewSessionBean implements NewSessionBeanLocal {
    @Override
    public boolean insert(int roll,String name,String mob){
        int i=0;
        try {
            Class.forName("org.apache.derby.jdbc.ClientDriver");
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(NewSessionBean.class.getName()).log(Level.SEVERE, null, ex);
        }
        try {
            Connection con;
            con = DriverManager.getConnection("jdbc:derby://localhost:1527/it","it","it");
         PreparedStatement ps=con.prepareStatement("insert into UNTITLED values (?,?,?)");
         ps.setInt(1,roll);
        ps.setString(2,name);
```

```
        ps.setString(3,mob);
        i=ps.executeUpdate();
        } catch (SQLException ex) {
            Logger.getLogger(NewSessionBean.class.getName()).log(Level.SEVERE, null, ex);
        }
       if (i==1) {
        return true;
       }
       else
           return false;
   }
}
```

Web.xml:

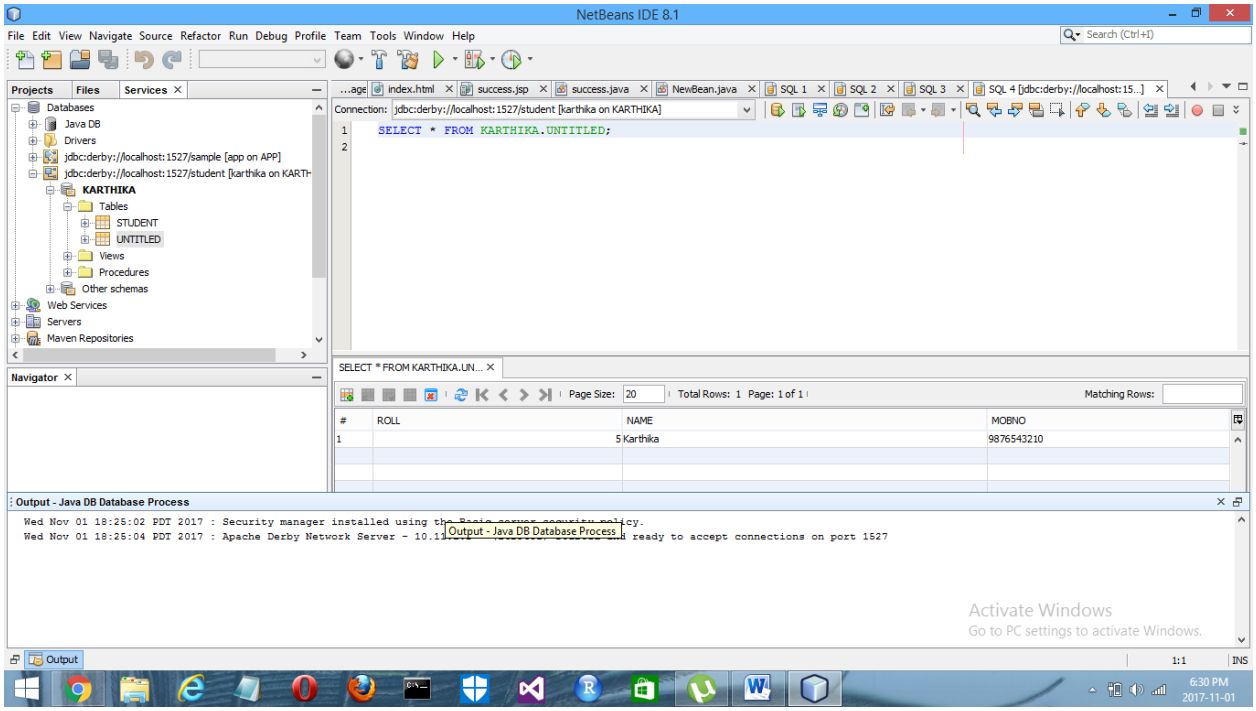
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd">
    <servlet>
        <servlet-name>NewServlet</servlet-name>
        <servlet-class>NewServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>NewServlet</servlet-name>
        <url-pattern>/NewServlet</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout> 30 </session-timeout>
    </session-config>
</web-app>
```

Output:



Hello World!

new record had been inserted



FOR INCORRECT DETAILS:

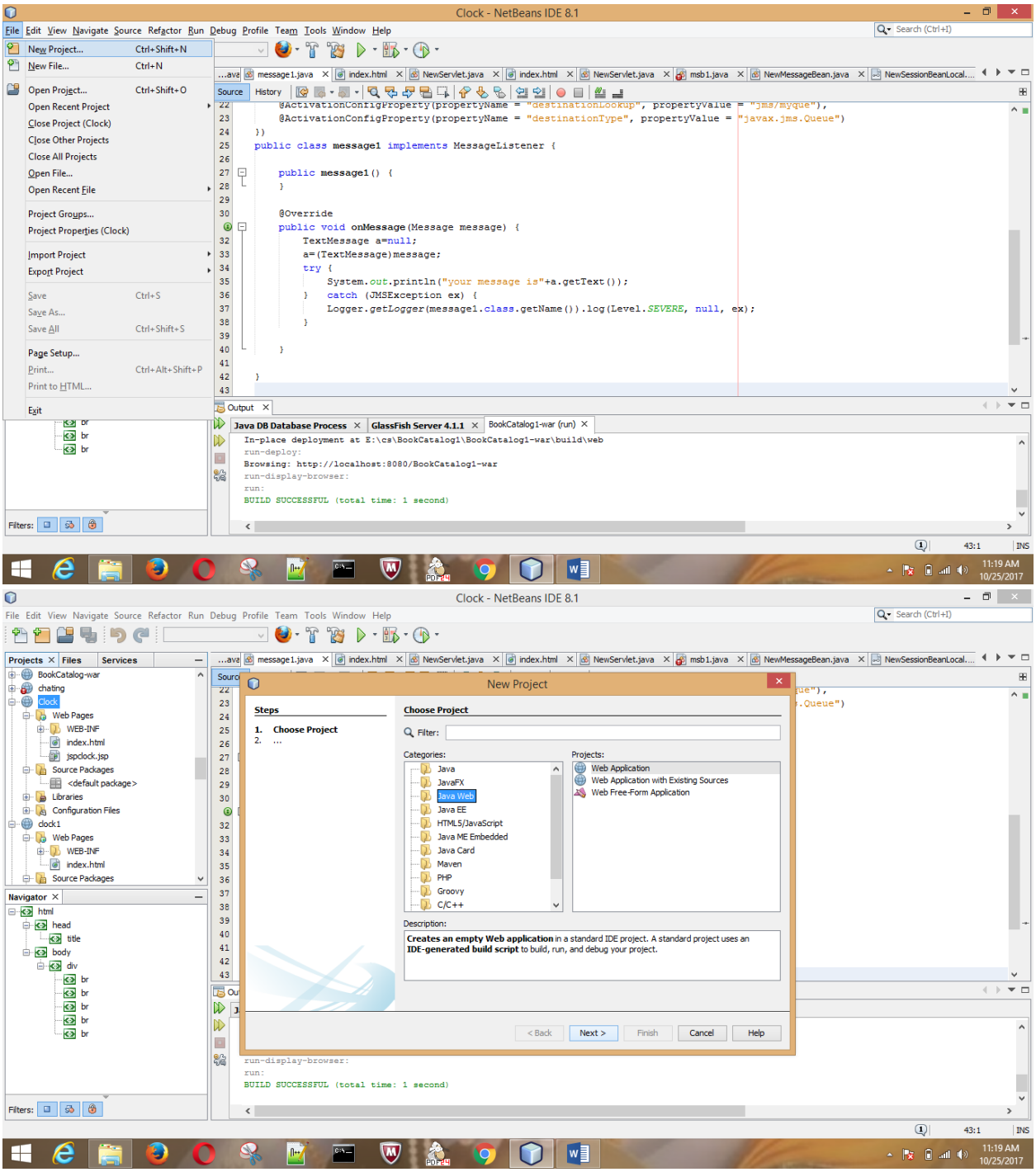
insertion failed !! Try again !!

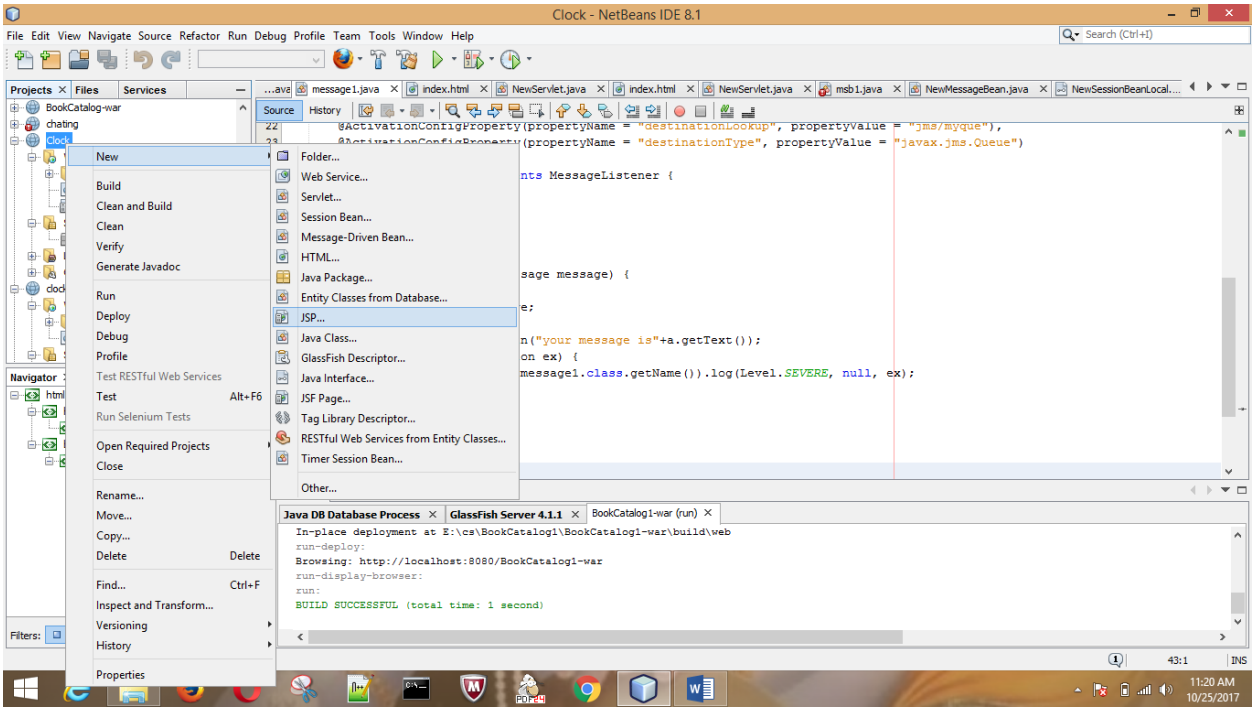
5.Write a program to demonstrate JSP custom tag.

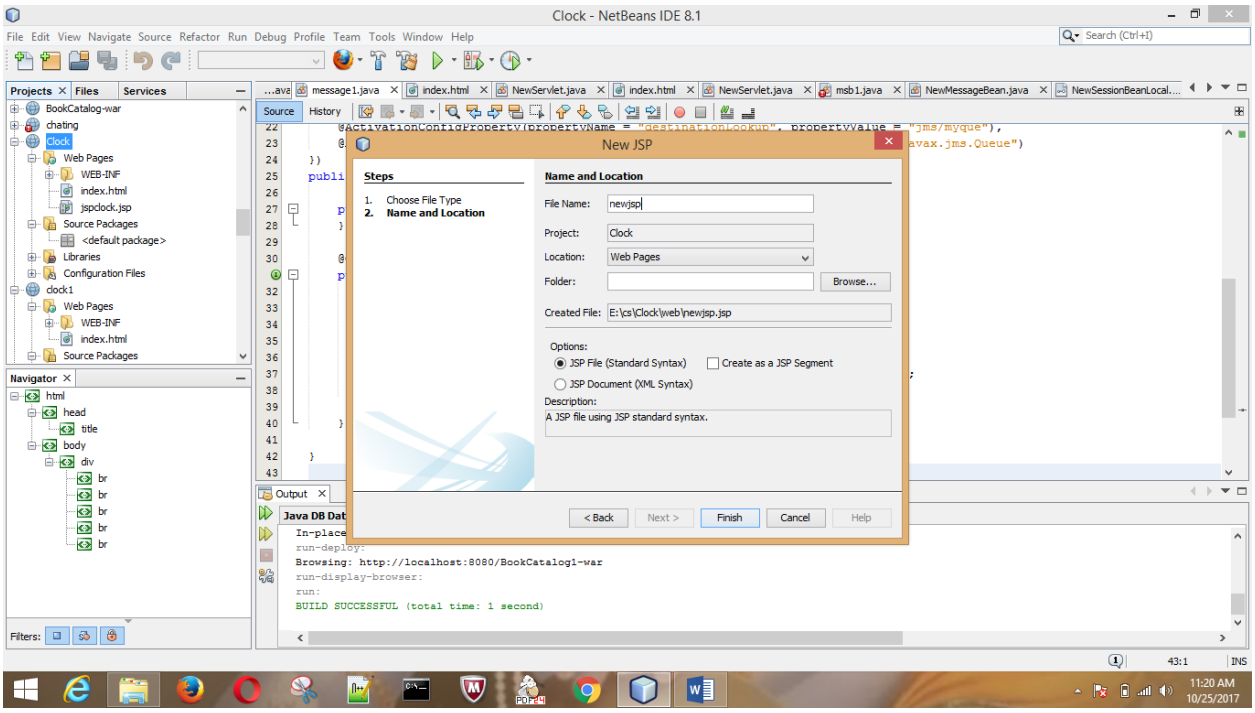
Aim:

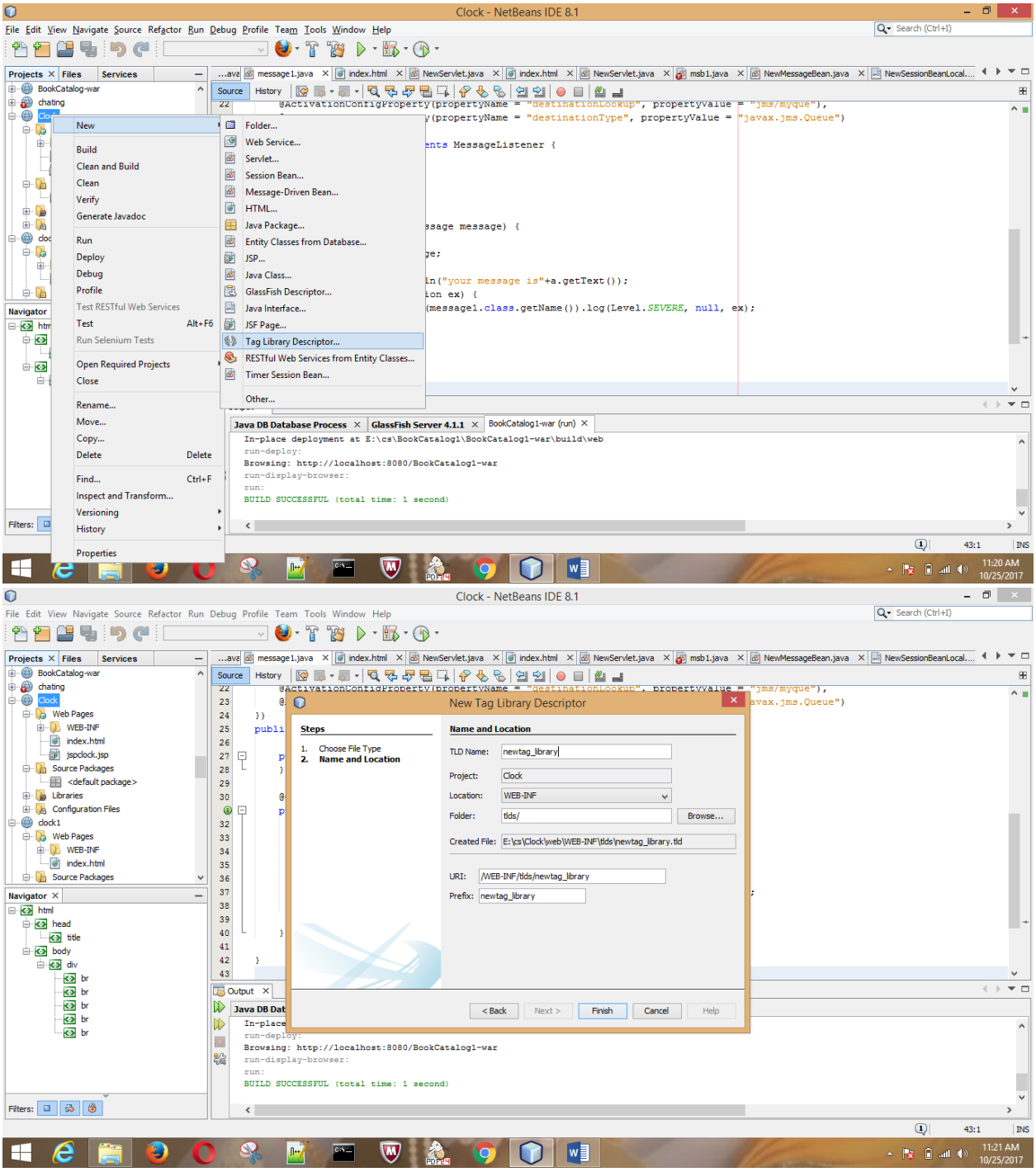
To demonstrate JSP custom tags using tag handler.

Source code:




Clock - NetBeans IDE 8.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Files Services
BookCatalog-war
chating
Clock
New
Build
Clean and Build
Clean
Verify
Generate Javadoc
Run
Deploy
Debug
Profile
Test RESTful Web Services
Test Alt+F6
Run Selenium Tests
Open Required Projects
Close
Rename...
Move...
Copy...
Delete Delete
Find... Ctrl+F
Inspect and Transform...
Versioning
History
Properties
Folder...
Web Service...
Servlet...
Session Bean...
Message-Driven Bean...
HTML...
Java Package...
Entity Classes from Database...
JSP...
Java Class...
GlassFish Descriptor...
Java Interface...
JSF Page...
Tag Library Descriptor...
RESTful Web Services from Entity Classes...
Timer Session Bean...
Other...
Navigator
Java DB Database Process
GlassFish Server 4.1.1
BookCatalog1-war (run)
In-place deployment at E:\cs\BookCatalog1\BookCatalog1-war\build\web
run-deploy:
Browsing: http://localhost:8080/BookCatalog1-war
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 1 second)
11:20 AM
10/25/2017


Clock - NetBeans IDE 8.1
New JSP
Steps
1. Choose File Type
2. Name and Location
Name and Location
File Name: newjsp
Project: Clock
Location: Web Pages
Folder:
Browse...
Created File: E:\cs\Clock\web\newjsp.jsp
Options:
JSP File (Standard Syntax)
Create as a JSP Segment
JSP Document (XML Syntax)
Description:
A JSP file using JSP standard syntax.
< Back
Next >
Finish
Cancel
Help
Output
Browsing: http://localhost:8080/BookCatalog1-war
BUILD SUCCESSFUL (total time: 1 second)
11:20 AM
10/25/2017



Index.jsp:

<html><head><title>Date</title></head>

<body><form action="date.jsp" method="get">

To Display current date and time
<input type="submit" value="on click"/>

</form></body></html>

Date.jsp:

<html><head>

```
<title>JSP Page</title>
</head><body>
Current date and time is:<m:today/>
</body></html>
```

Right click on project → New → Tag file

Clock(MyTagHandler).java:

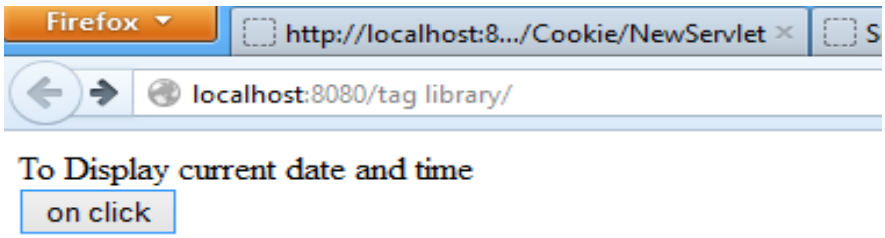
```
package bec;
import java.util.Calendar;
import java.servlet.jsp.JspException;
import java.servlet.jsp.JspWriter;
import java.servlet.tagext;
import java.servlet.TagSupport;
public class MyTagHandler extends TagSupport{
@override
public intdoStartTag() throws JspException
{
JspWriter out=pageContext.getOut();
try{
out.print(Calender.getInstance().getTime());
}
catch(Exception e){
System.out.println(e);
}
return SKIP_BODY;
}
}
```

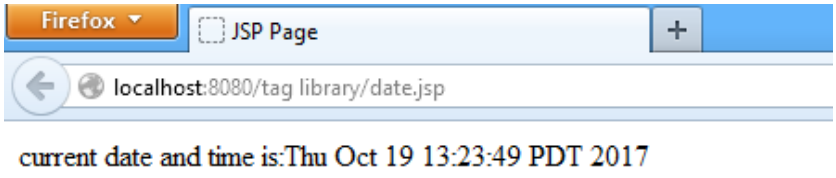
Right click on the project → New → Tag Descriptor file

NewTag-library.tld:

```
<?xml version="1.0" encoding="UTF-8"?>

<taglib version="2.1" xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://java.sun.com/xml/ns/javaee

http://java.sun.com/xml/ns/javaee/web-jsptaglibrary_2_1.xsd">

<tlib-version>1.0</tlib-version>

<short-name>newtag_library</short-name>

<uri>/WEB-INF/tlds/newtag_library</uri>

<tag>

<name>today</name>

<tag-class>bec.NewTagHandler</tag-class>

<body-content>scriptless</body-content>

</tag><tag>

<name>NewTagHandler</name>

<tag-class>bec.NewTagHandler</tag-class>

<body-content>scriptless</body-content>

</tag>

</taglib>
```

Output:


Firefox
http://localhost:8.../Cookie/NewServlet
localhost:8080/tag library/
To Display current date and time
on click

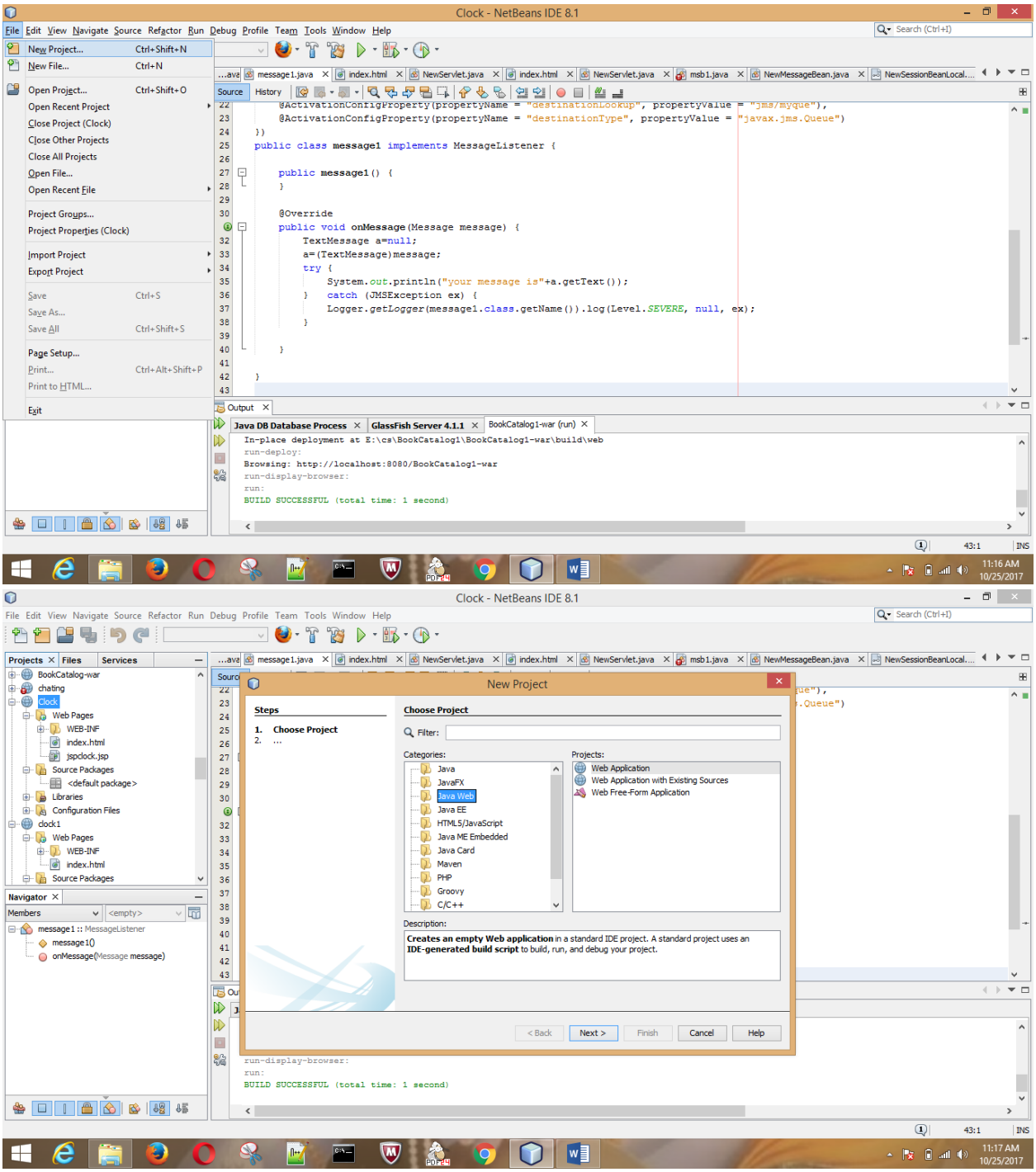

Firefox
JSP Page
localhost:8080/tag library/date.jsp
current date and time is:Thu Oct 19 13:23:49 PDT 2017

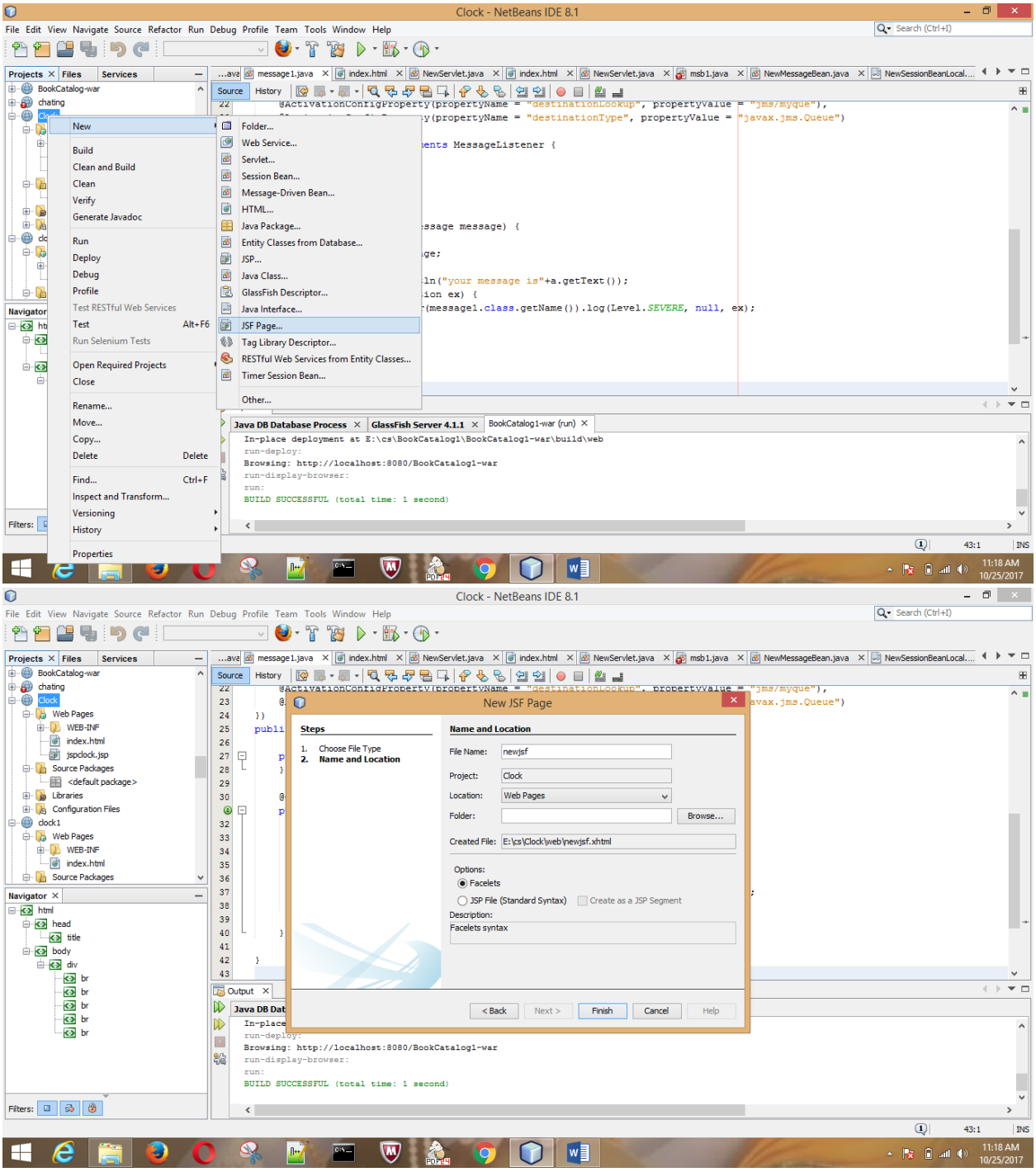
6.Write a program to demonstrate JSF.

Aim:

To demonstrate print current date and time using jsf.

Source code:





Index.xhtml:

<htmlxmlns=http://www.w3.org/1999/xhtmlxmlns:h="http://xmlns.jcp.org/jsf/html">

<h:head><title>JSF</title>

</h:head><h:body><h:form>

<h:inputText value="#{displayName.name}"/>

</h:form>

Hello to, #{displayName.name} !

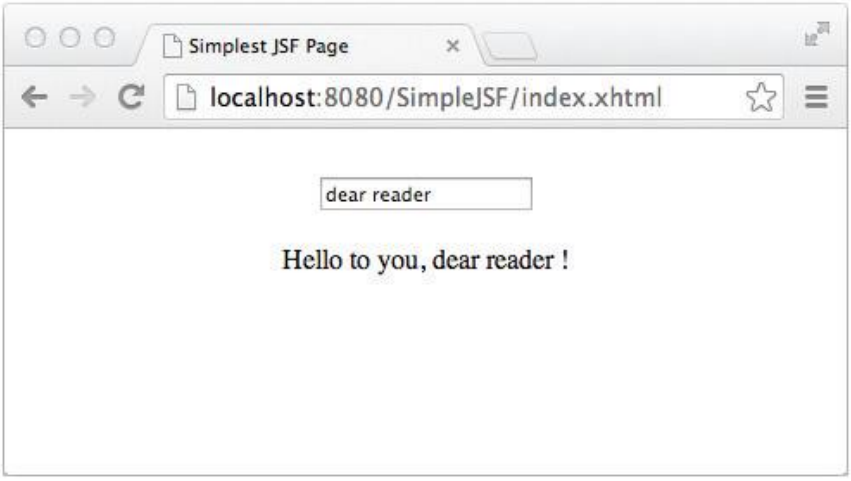
```
</h:body>
</html>
```

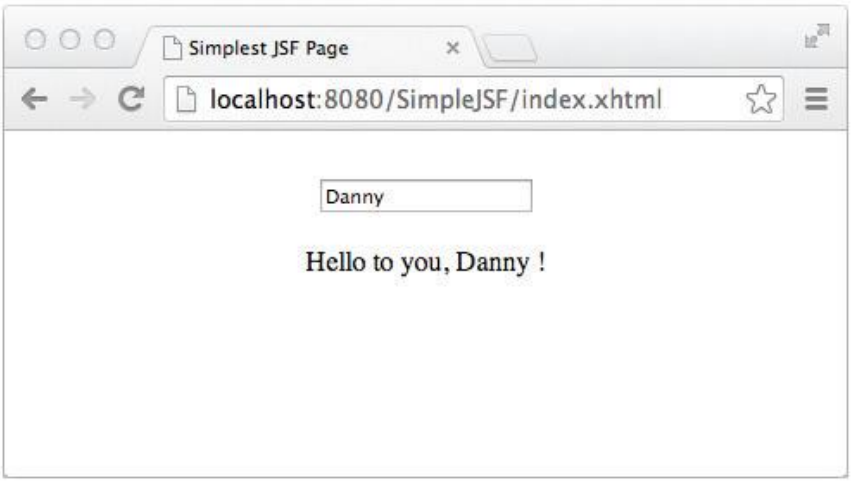
Right click on project → New → JSF Managed Bean

DisplayName.java:

```
import javax.inject.Named;
import javax.enterprise.context.RequestScoped;
@Named(value = "displayName")
@RequestScoped
public class DisplayName {
private String name = "dear reader";
public void setName(String name) {
this.name = name;
}
public String getName() {
return this.name;
public DisplayName() {
}
}
```

Output:


Simplest JSF Page
localhost:8080/SimpleJSF/index.xhtml
dear reader
Hello to you, dear reader !

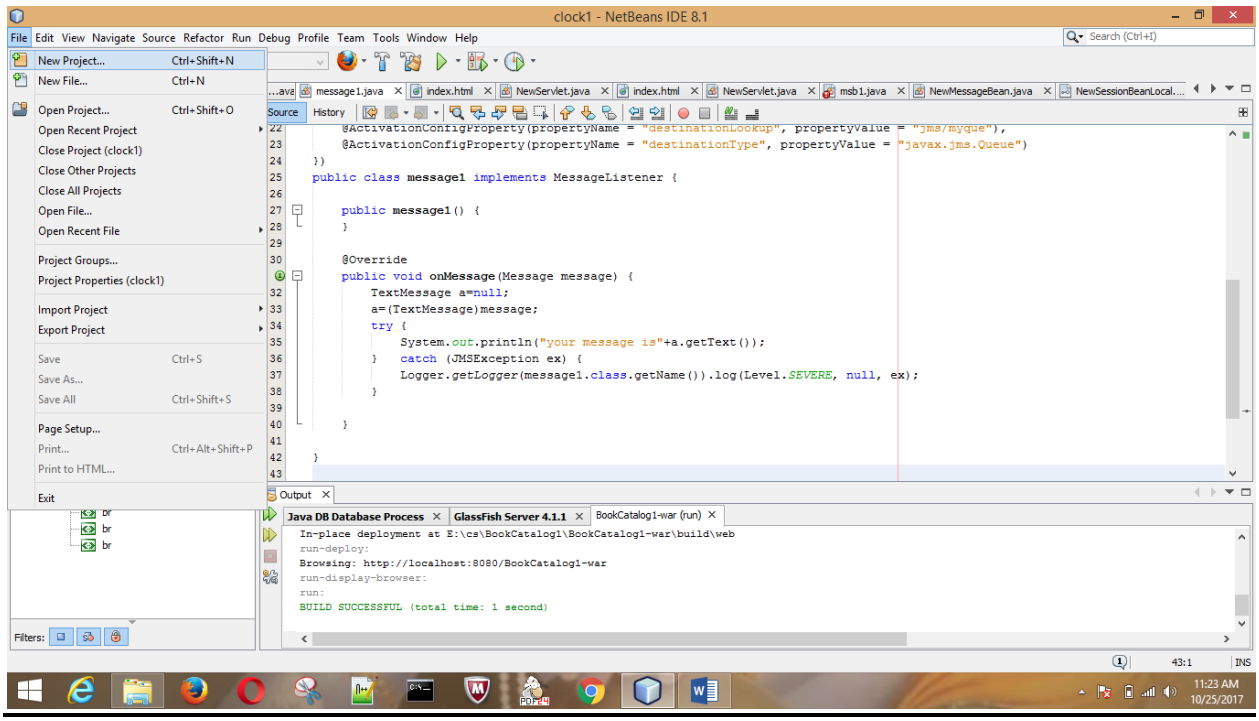

Simplest JSF Page
localhost:8080/SimpleJSF/index.xhtml
Danny
Hello to you, Danny !

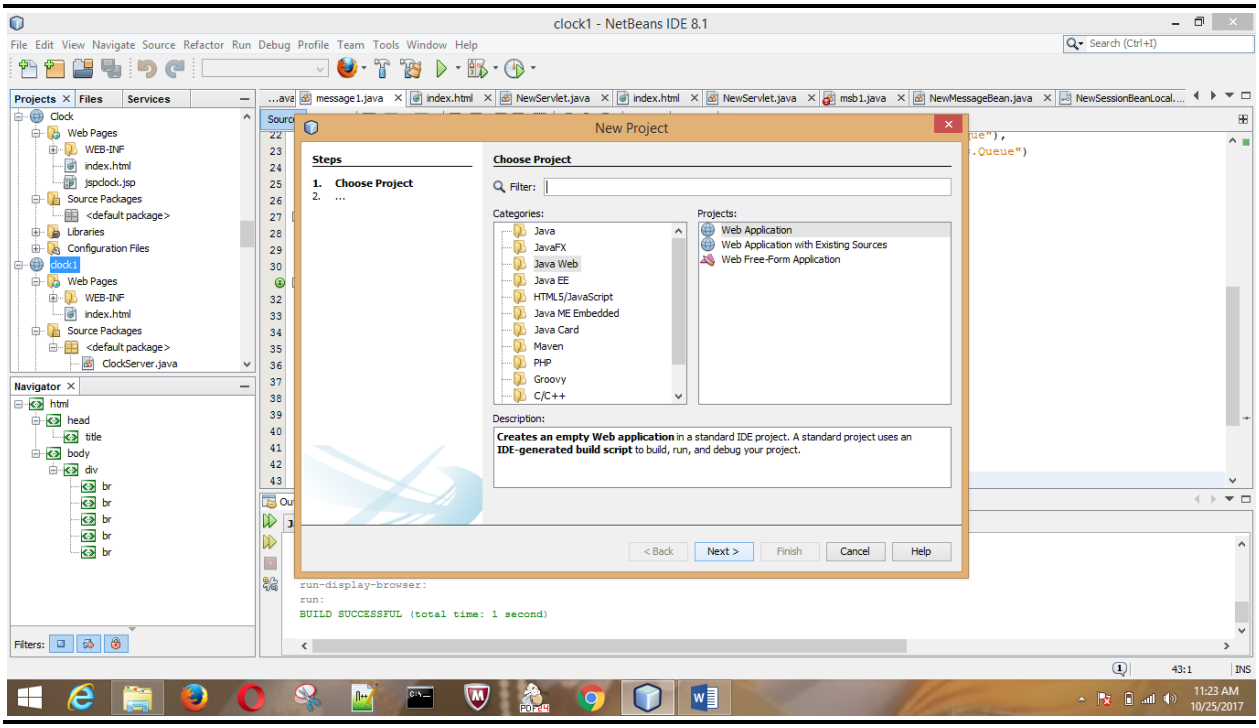
7.Write a program to demonstrate WEB SOCKETS.

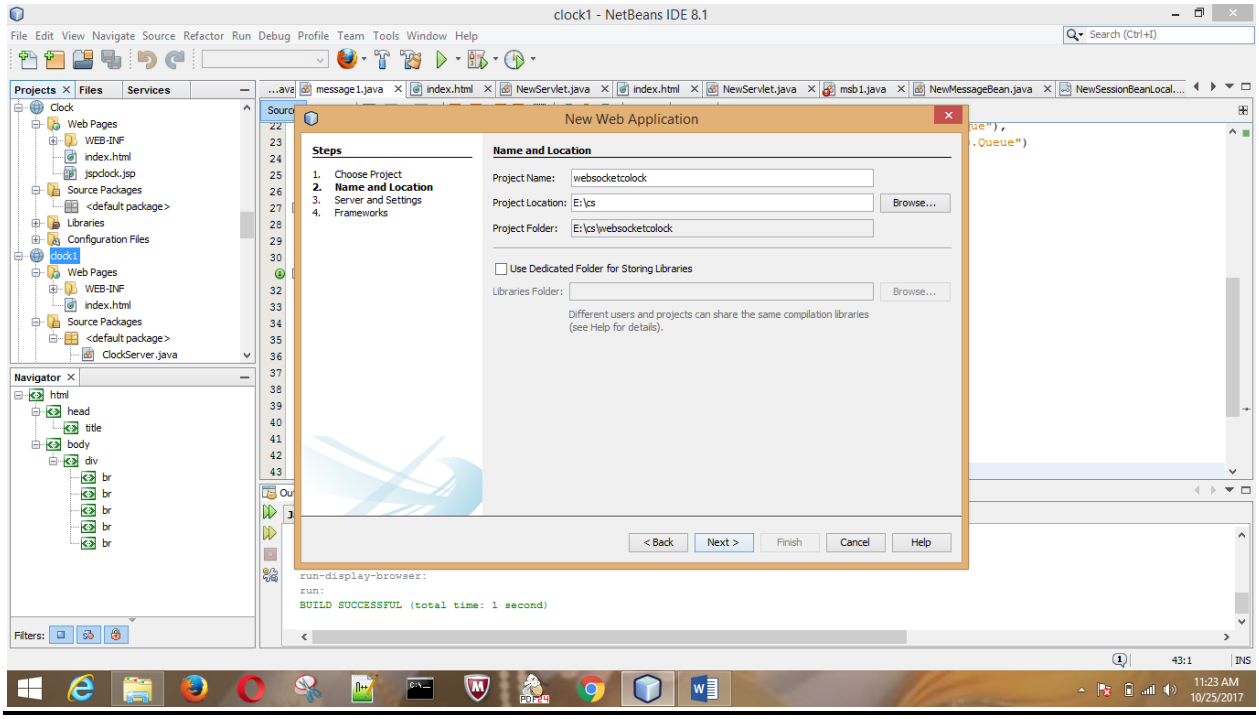
Aim:

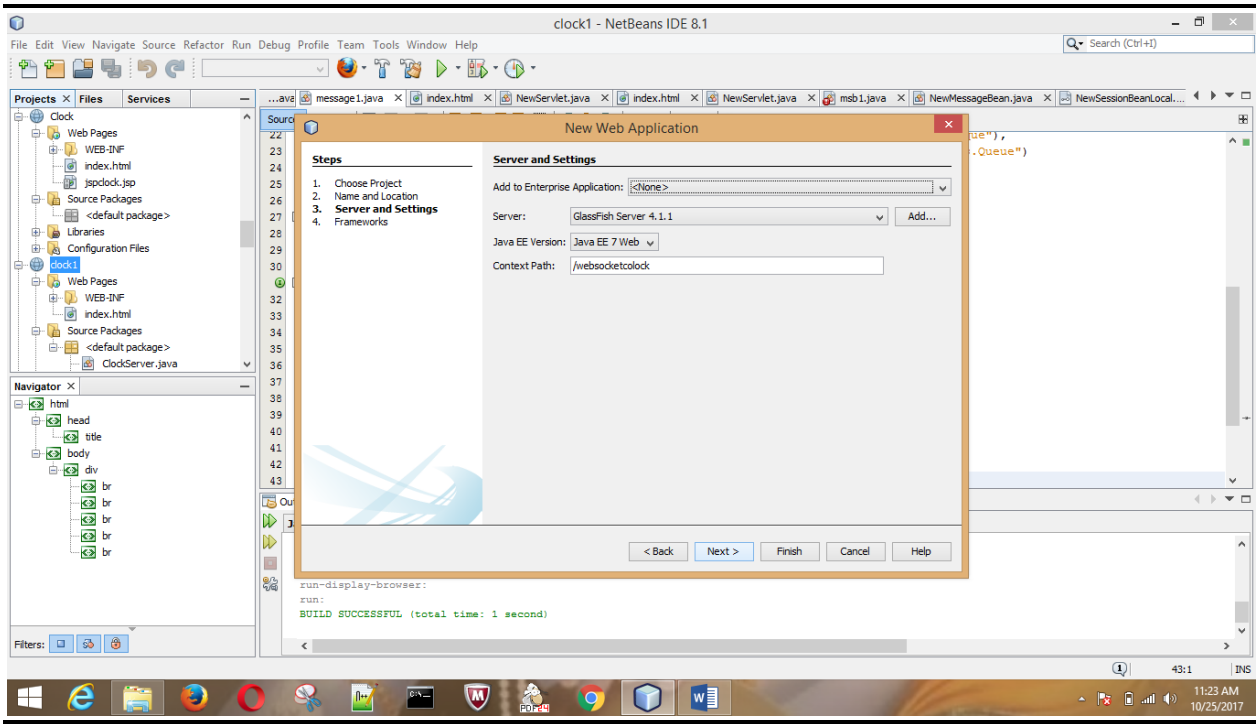
To demonstrate clock application using web sockets.

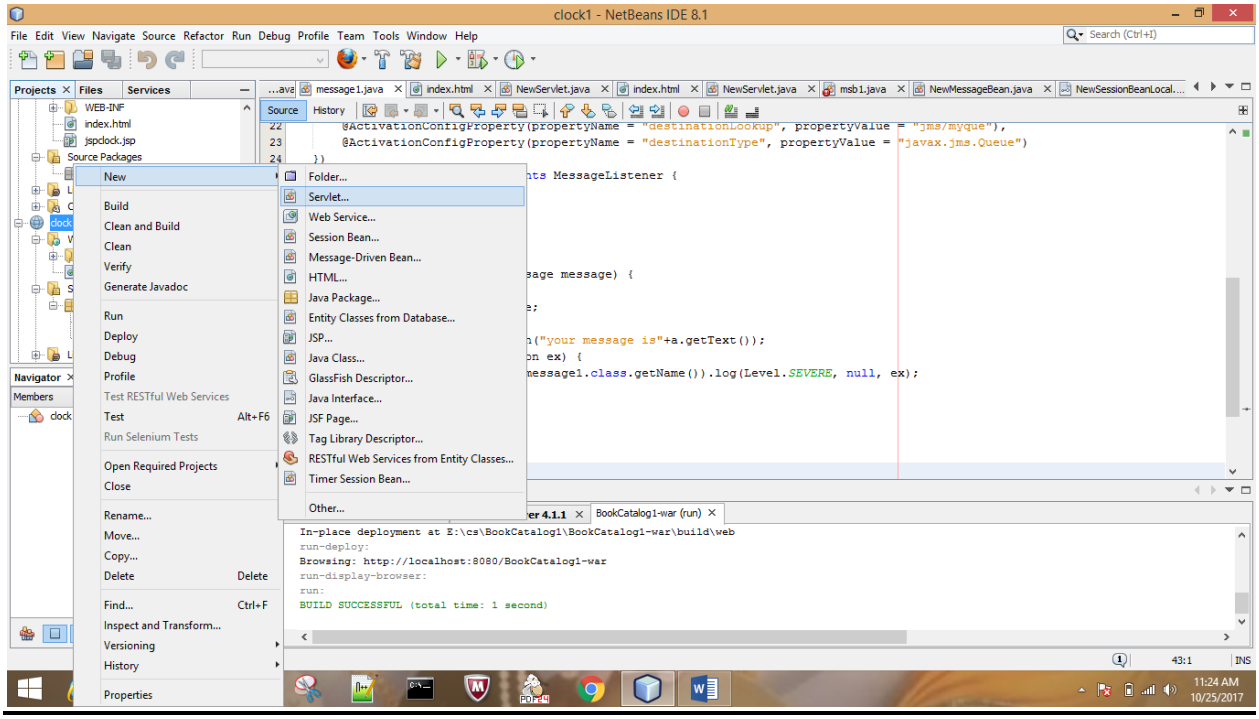
Source code:

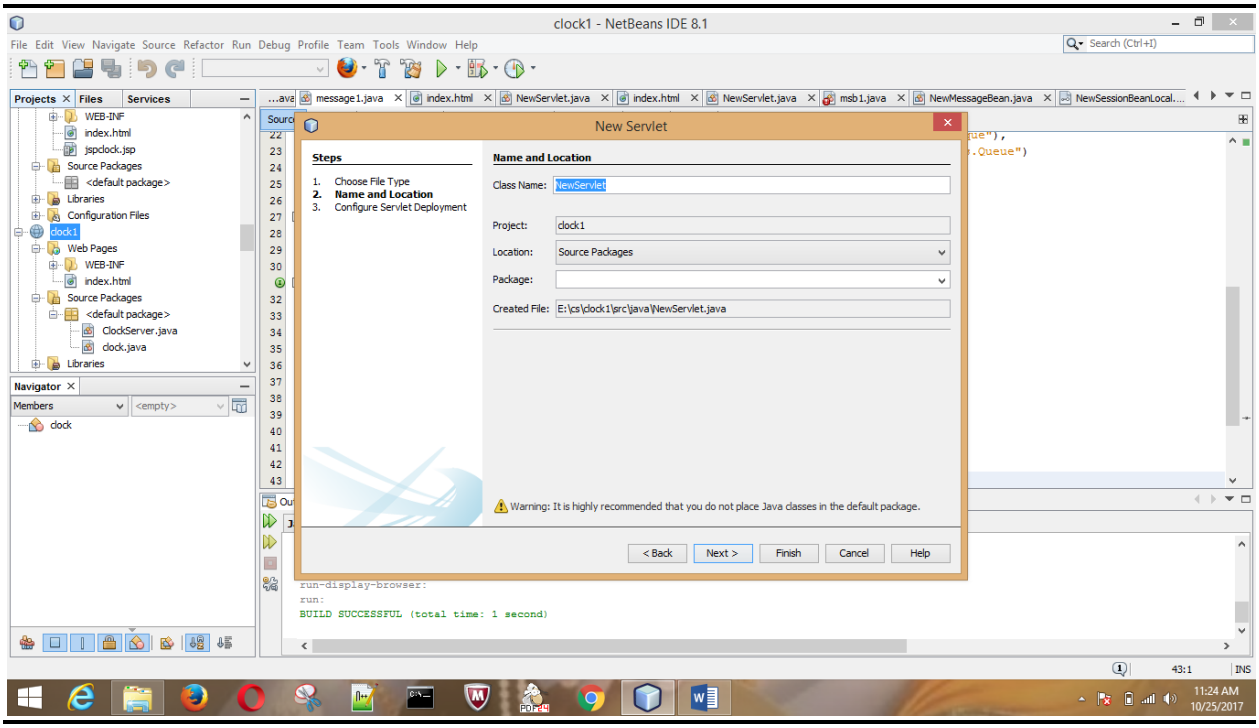


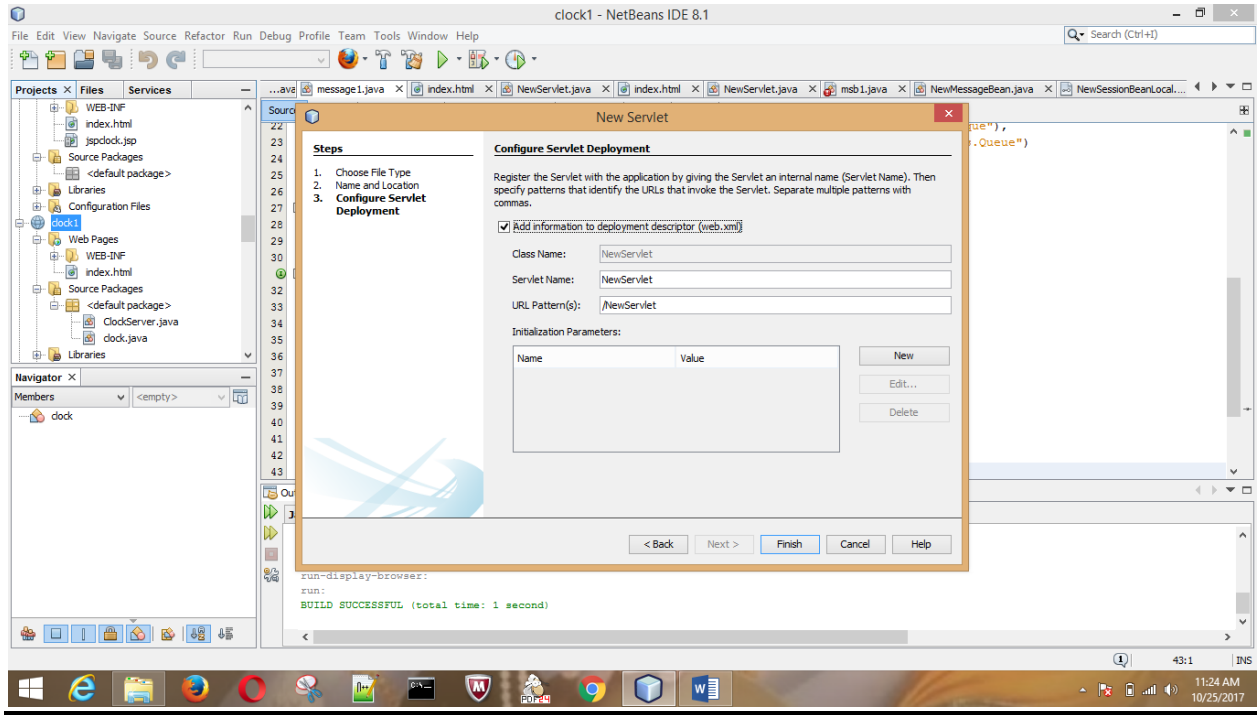



clock1 - NetBeans IDE 8.1
New Web Application
Steps
1. Choose Project
2. Name and Location
3. Server and Settings
4. Frameworks
Name and Location
Project Name: websocketcolock
Project Location: E:\cs
Project Folder: E:\cs\websocketcolock
Use Dedicated Folder for Storing Libraries
Libraries Folder:
Different users and projects can share the same compilation libraries (see Help for details).
< Back
Next >
Finish
Cancel
Help
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 1 second)


clock1 - NetBeans IDE 8.1
New Web Application
Steps
1. Choose Project
2. Name and Location
3. Server and Settings
4. Frameworks
Server and Settings
Add to Enterprise Application: <None>
Server: GlassFish Server 4.1.1
Add...
Java EE Version: Java EE 7 Web
Context Path: /websocketcolock
< Back
Next >
Finish
Cancel
Help
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 1 second)


clock1 - NetBeans IDE 8.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Files Services
New
Folder...
Servlet...
Web Service...
Session Bean...
Message-Driven Bean...
HTML...
Java Package...
Entity Classes from Database...
JSP...
Java Class...
GlassFish Descriptor...
Java Interface...
JSF Page...
Tag Library Descriptor...
RESTful Web Services from Entity Classes...
Timer Session Bean...
Other...
Build
Clean and Build
Clean
Verify
Generate Javadoc
Run
Deploy
Debug
Profile
Test RESTful Web Services
Test Alt+F6
Run Selenium Tests
Open Required Projects
Close
Rename...
Move...
Copy...
Delete Delete
Find... Ctrl+F
Inspect and Transform...
Versioning
History
Properties
In-place deployment at E:\cs\BookCatalog1\BookCatalog1-war\build\web
run-deploy:
Browsing: http://localhost:8080/BookCatalog1-war
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 1 second)


clock1 - NetBeans IDE 8.1
New Servlet
Steps
1. Choose File Type
2. Name and Location
3. Configure Servlet Deployment
Name and Location
Class Name: NewServlet
Project: clock1
Location: Source Packages
Package:
Created File: E:\cs\clock1\src\java\NewServlet.java
Warning: It is highly recommended that you do not place Java classes in the default package.
< Back
Next >
Finish
Cancel
Help
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 1 second)



Index.html:

```
<html>
<head><title>JSP</title>
<script language="javascript" type="text/javascript">
varwebsocket;
varlast_time;
function init() {
output = document.getElementById("output");
}
function start_clock() {
varwsUri = "ws://localhost:8080/Socket1/endpoint";
websocket = new WebSocket(wsUri);
websocket.onmessage = function (evt) {
last_time = evt.data;
writeToScreen("<span style='color: blue;'>" + last_time + "</span>");
};
websocket.onerror = function (evt) {
```

```
writeToScreen('<span style="color: red;">ERROR:</span> ' + evt.data);
websocket.close();
};
}
function stop_clock() {
websocket.send("stop");
}
function writeToScreen(message) {
var pre = document.createElement("p");
pre.style.wordWrap = "break-word";
pre.innerHTML = message;
oldChild = output.firstChild;
if ( oldChild == null) {
output.appendChild(pre);
} else {
output.removeChild(oldChild);
output.appendChild(pre);
}}
window.addEventListener("load", init, false);
</script>
</head>
<body><div style="text-align: center;font-family: Arial; font-size: large">
Web Socket Clock<br></br>
<form action=""><inputonclick="start_clock()"title="Press to start the clock on the
server"value="Start"type="button" />
<inputonclick="stop_clock()"title="Press to stop the clock on the server"
value="Stop"type="button" />
```

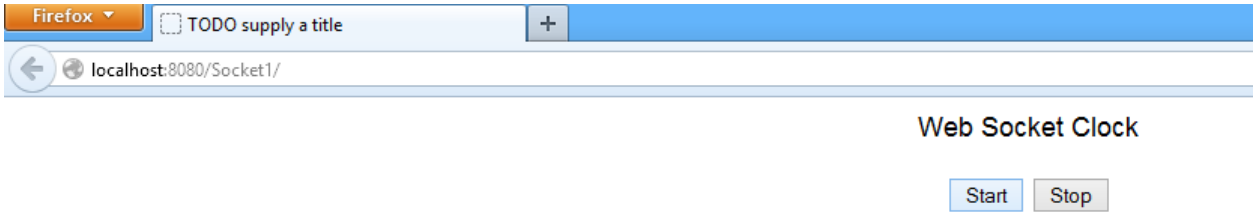
```
</form><div id="output"></div></div></body>
</html>
```

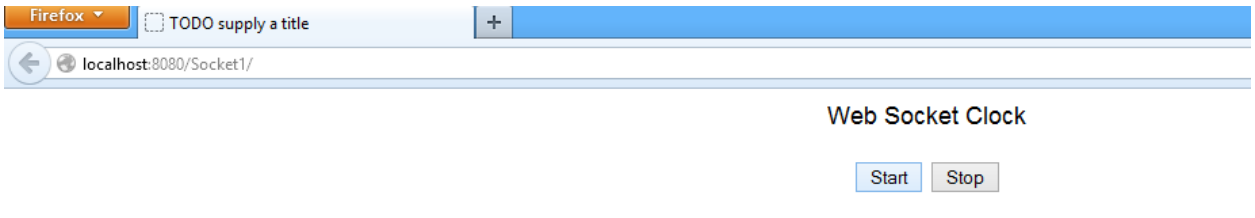
WebSocket1.java:

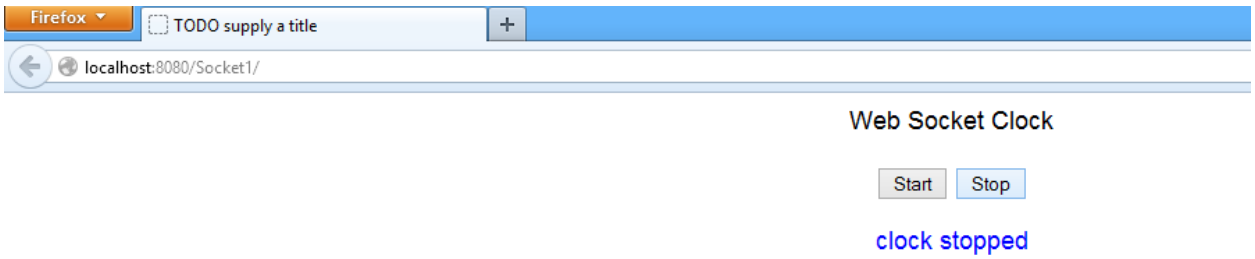
```
package bec;
import java.io.IOException;
import static java.lang.Thread.sleep;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.websocket.OnClose;
import javax.websocket.OnError;
import javax.websocket.OnMessage;
import javax.websocket.OnOpen;
import javax.websocket.Session;
import javax.websocket.server.ServerEndpoint;
@ServerEndpoint("/endpoint")
public class NewWSEndpoint {
Thread updateThread;
boolean running = false;
@OnOpen
public void startClock(Session session) {
final Session mySession = session;
this.running = true;
final SimpleDateFormatsdf = new SimpleDateFormat("h:mm:ss a");
this.updateThread = new Thread() {
public void run() {
while(running) {
```

```
String dateString = sdf.format(new Date());
try {
mySession.getBasicRemote().sendText(dateString);
sleep(1000);
} catch(IOException | InterruptedExceptionie) {
running = false;
}}}};
this.updateThread.start();
}
@OnMessage
public String handleMessage(String incomingMessage) {
if ("stop".equals(incomingMessage)) {
this.stopClock();
return "clock stopped";
} else {
return "unknown message: " + incomingMessage;
}}
@OnError
public void clockError(Throwable t) {
this.stopClock();
}@OnClose
public void stopClock() {
this.running = false;
this.updateThread = null;
}
}
```

Output:


Firefox
TODO supply a title
localhost:8080/Socket1/
Web Socket Clock
Start
Stop


Firefox
TODO supply a title
localhost:8080/Socket1/
Web Socket Clock
Start
Stop
1:14:09 PM

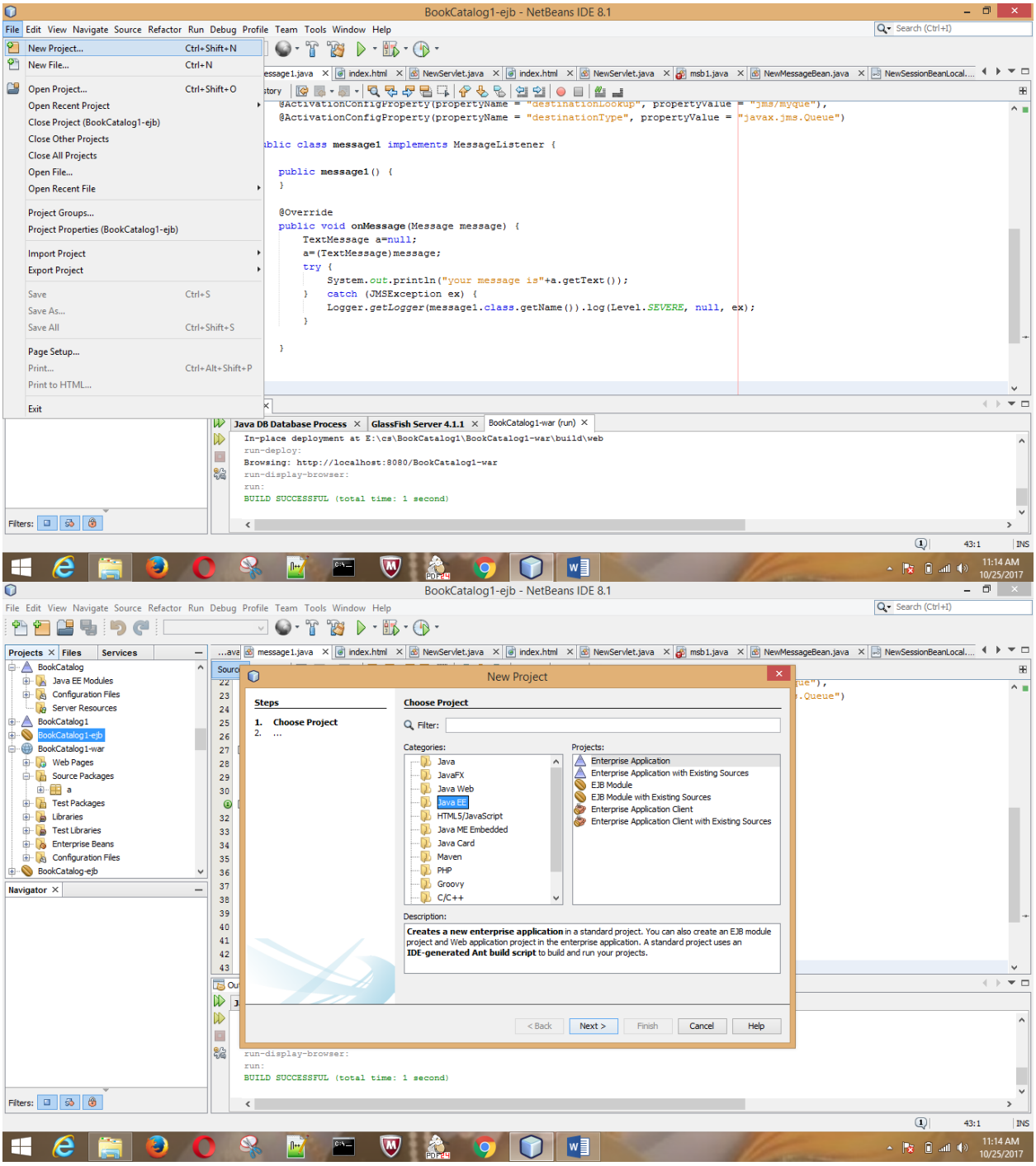

Firefox
TODO supply a title
localhost:8080/Socket1/
Web Socket Clock
Start
Stop
clock stopped

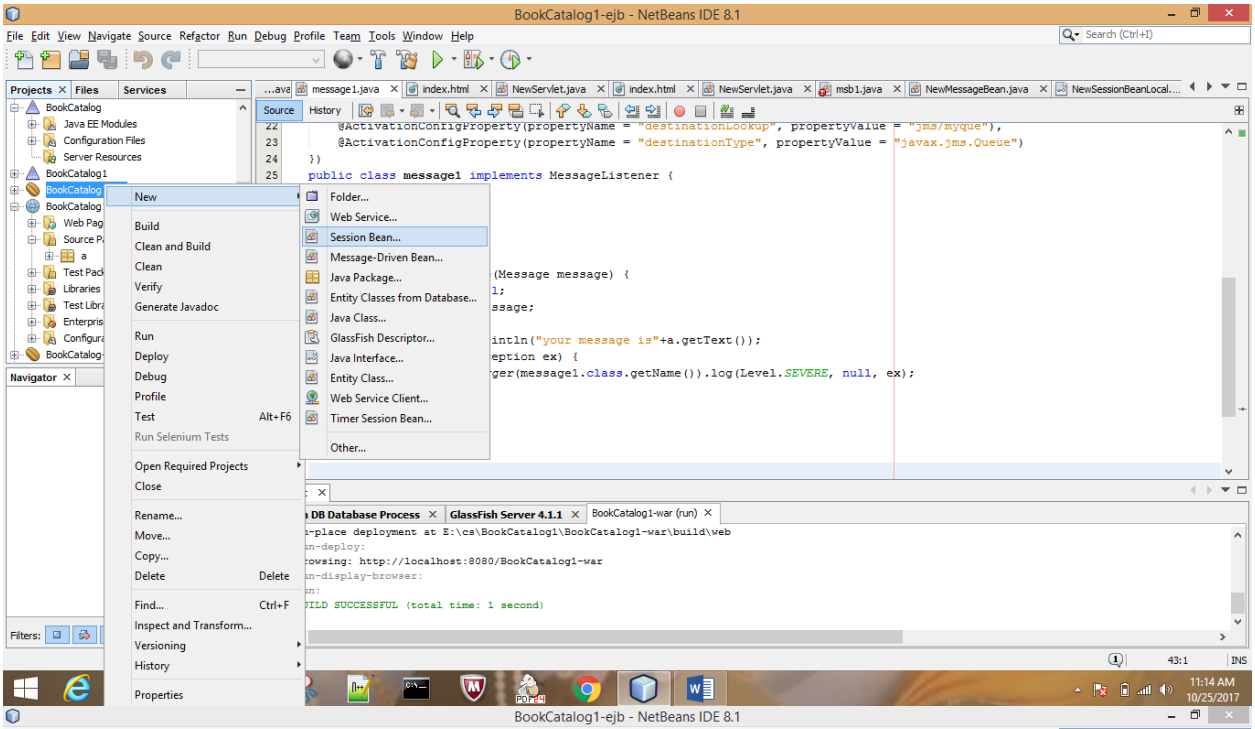
8.Write a program to demonstrate bank application using session bean.

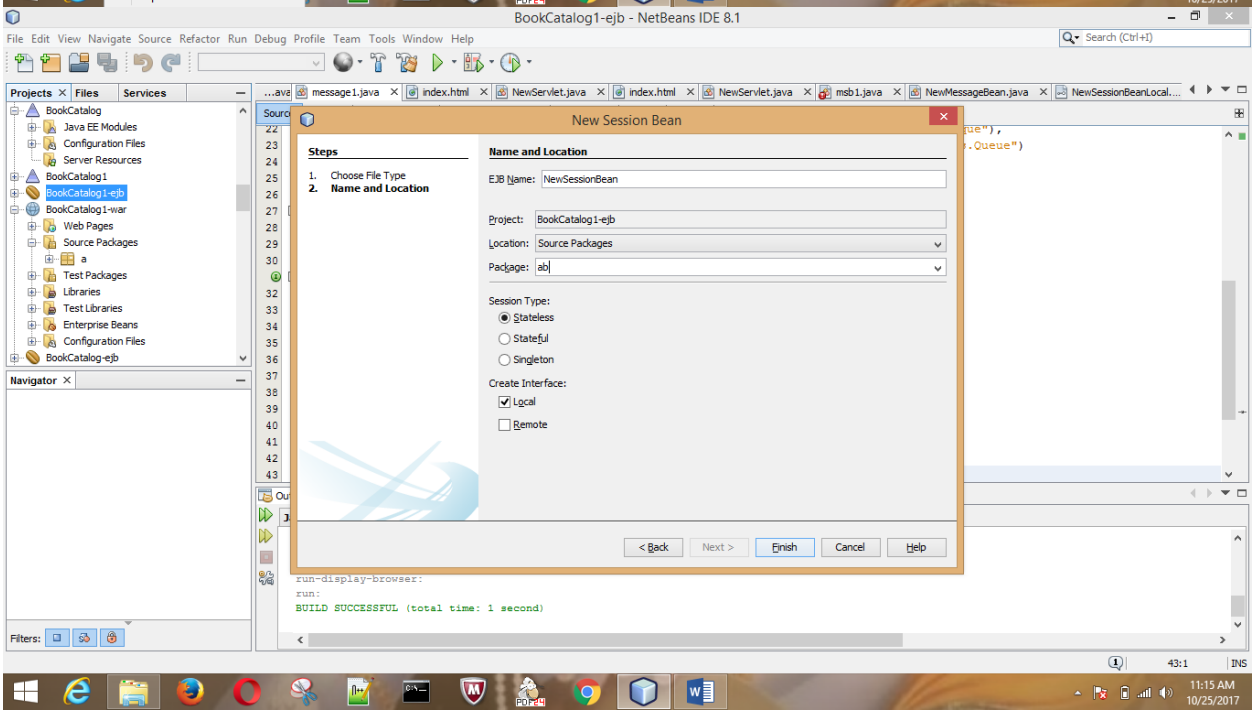
Aim:

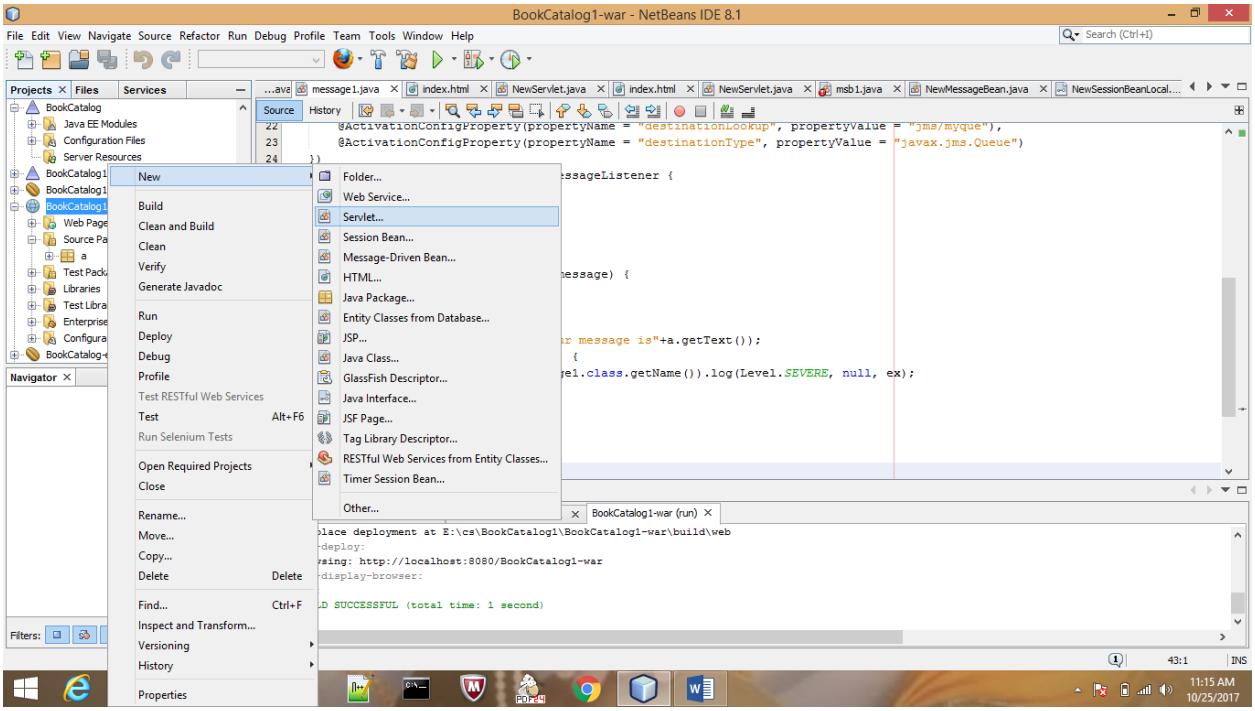
To demonstrate bank application using session bean.

Source code:




BookCatalog1-ejb - NetBeans IDE 8.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Files Services
BookCatalog
Java EE Modules
Configuration Files
Server Resources
BookCatalog1
BookCatalog1-ejb
BookCatalog1-war
Web Pages
Source Packages
Test Packages
Libraries
Test Libraries
Enterprise Beans
Navigator
message1.java
index.html
NewServlet.java
msb1.java
NewMessageBean.java
Source History
@ActivationConfigProperty(propertyName = "destinationLookup", propertyValue = "jms/myque"),
@ActivationConfigProperty(propertyName = "destinationType", propertyValue = "javax.jms.Queue")
public class message1 implements MessageListener {
New
Build
Clean and Build
Clean
Verify
Generate Javadoc
Run
Deploy
Debug
Profile
Test Alt+F6
Run Selenium Tests
Open Required Projects
Close
Rename...
Move...
Copy...
Delete Delete
Find... Ctrl+F
Inspect and Transform...
Versioning
History
Properties
Folder...
Web Service...
Session Bean...
Message-Driven Bean...
Java Package...
Entity Classes from Database...
Java Class...
GlassFish Descriptor...
Java Interface...
Entity Class...
Web Service Client...
Timer Session Bean...
Other...
DB Database Process
GlassFish Server 4.1.1
BookCatalog1-war (run)
BUILD SUCCESSFUL (total time: 1 second)
43:1 INS
11:14 AM
10/25/2017


BookCatalog1-ejb - NetBeans IDE 8.1
New Session Bean
Steps
1. Choose File Type
2. Name and Location
Name and Location
EJB Name: NewSessionBean
Project: BookCatalog1-ejb
Location: Source Packages
Package: ab
Session Type:
Stateless
Stateful
Singleton
Create Interface:
Local
Remote
< Back Next > Finish Cancel Help
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 1 second)
43:1 INS
11:15 AM
10/25/2017



Index.html:

```
<html><head>
<title>Bank</title>
</head>
<body><p align="center"><font size="6" color="#800000">
<b>Welcome to<br>
EJB tutorial</b></font>
Click<a href="form.jsp"> Bank Transaction example</a>
to execute Bank bean<br></p>
</body></html>
```

Form.jsp:

```
<html><head>
<title></title></head><body>
<body>
<h1><p align="center"><font size="6" color="#800000">
Bank Transaction Request form</h1>
<hr><br>
```

```
<table bgcolor="#FFCCCC" align="center">
<form action="accservlet" method="post">
<tr><td></td></tr><tr><td>
Enter amount in Rupees:<input type="text" name="amt" size="10"></tr>
</td><br>
<tr><td>
<b>Select your choice:</b></tr></td>
<br>
<tr><td>
<input type="radio" name="group1" value="dep">Deposit</tr>
</td>
<tr><td>
<input type="radio" name="group1" value="with">Withdraw</tr>
</td><tr><td>
<input type="submit" name="Transmit">
<input type="reset" value="reset"></tr>
</td></tr></td></form>
</table>
</body>
</html>
```

Right click on ejb module → New → Session Bean (Select local mode)

Account.java:

```
package ejbexample;
import javax.ejb.Stateful;
@Stateful
public class Account implements AccountLocal
```

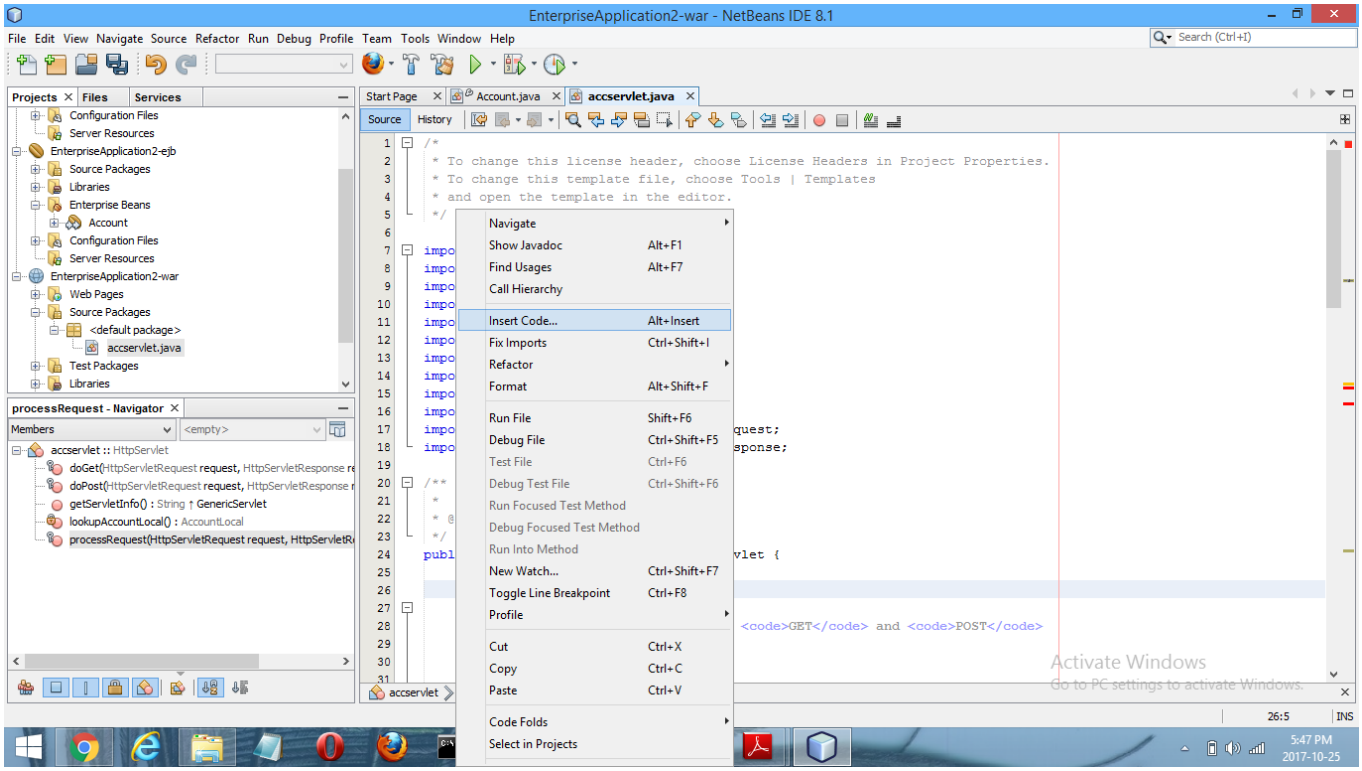
```
{
 float bal=0;
 @Override
 public float deposit(float amount)
{
bal+=amount;
 return bal;
 }
 @Override
 public float withdraw(float amount)
{
bal -=amount;
 return bal;
}
}
```

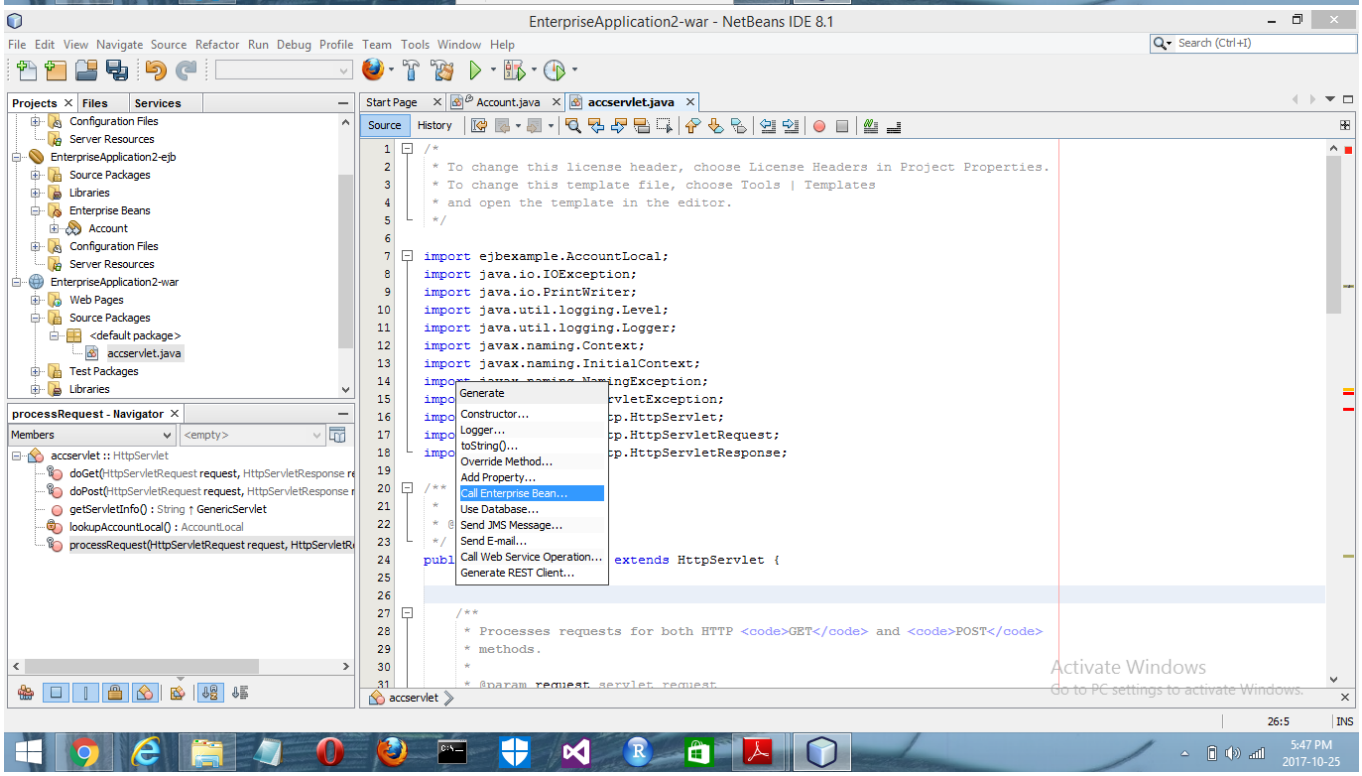
AccountLocal.java:

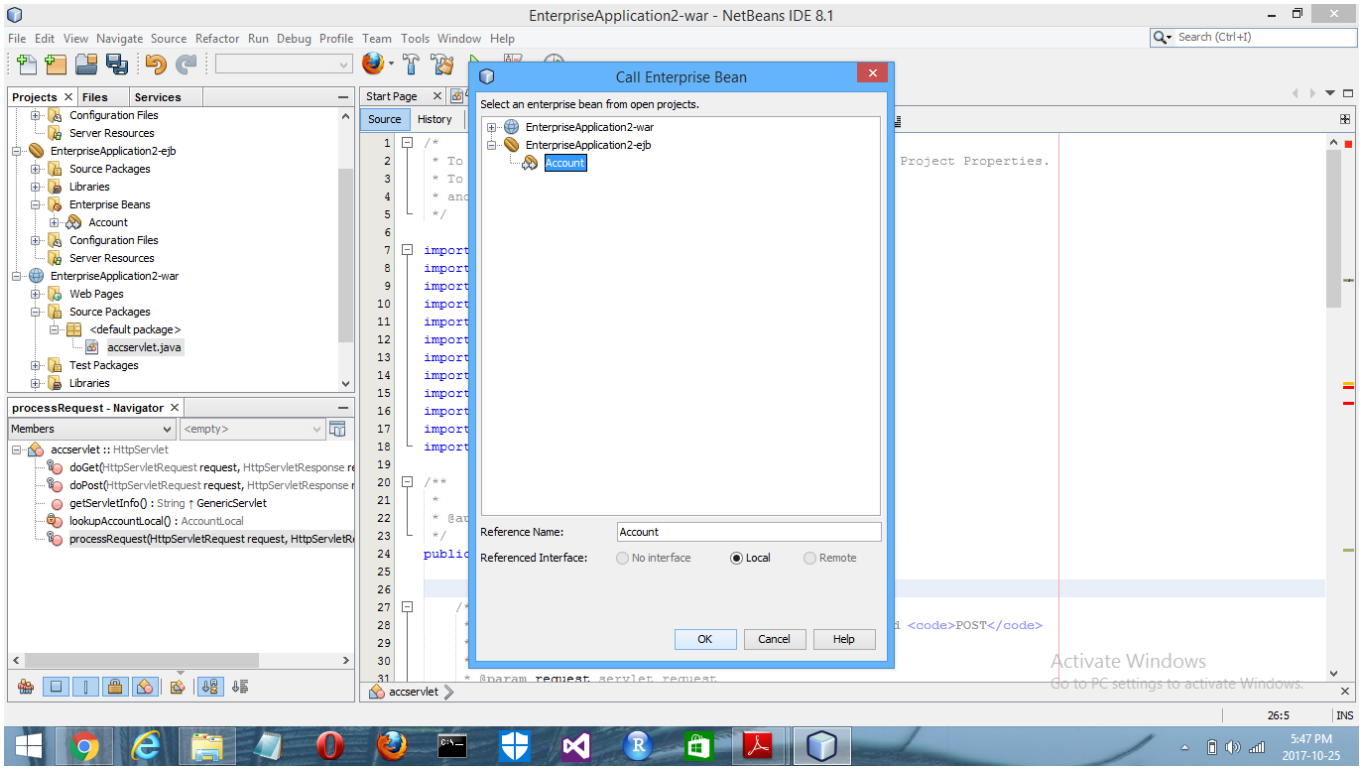
```
package ejbexample;
import javax.ejb.Local;
@Local
public interface AccountLocal
{
public float deposit(float amount);
public float withdraw(float amount);
}
```

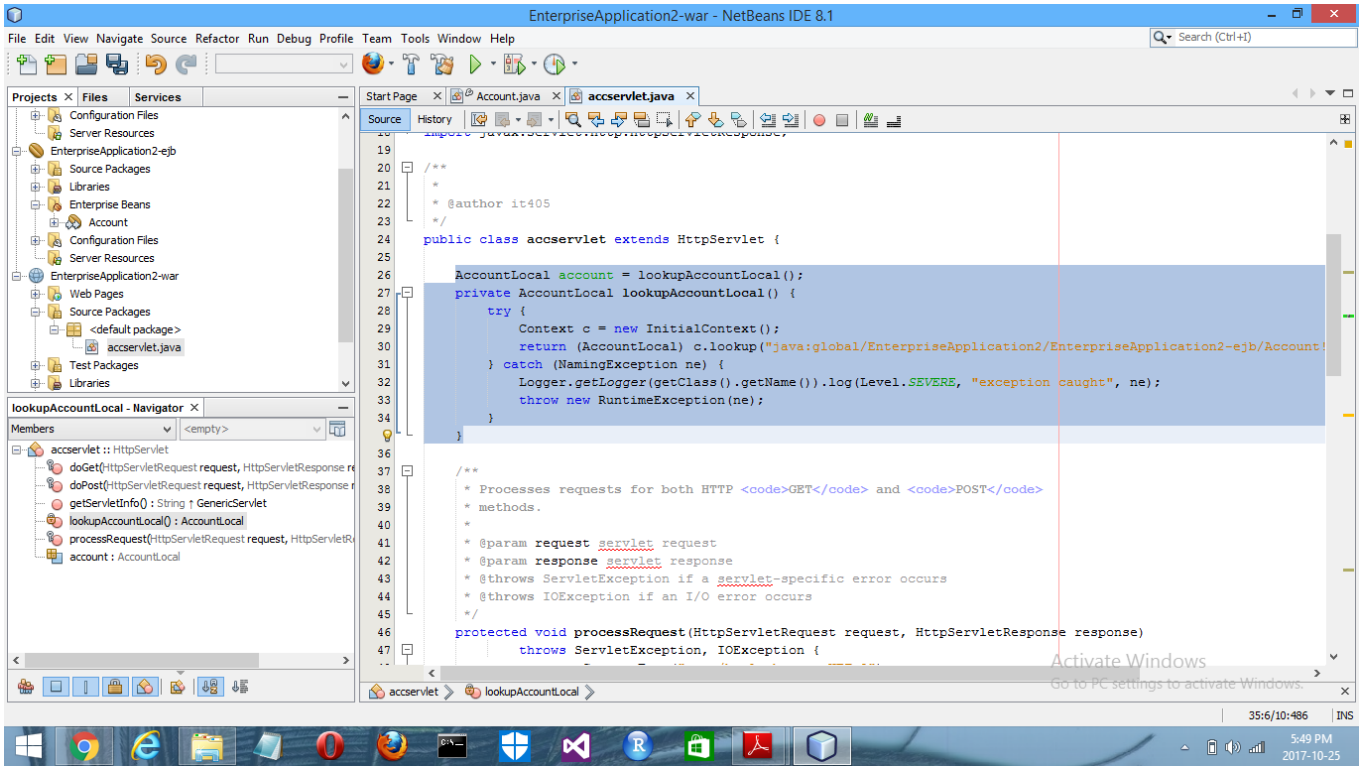
Right click on war module → New → Servlet

AccServlet.java:







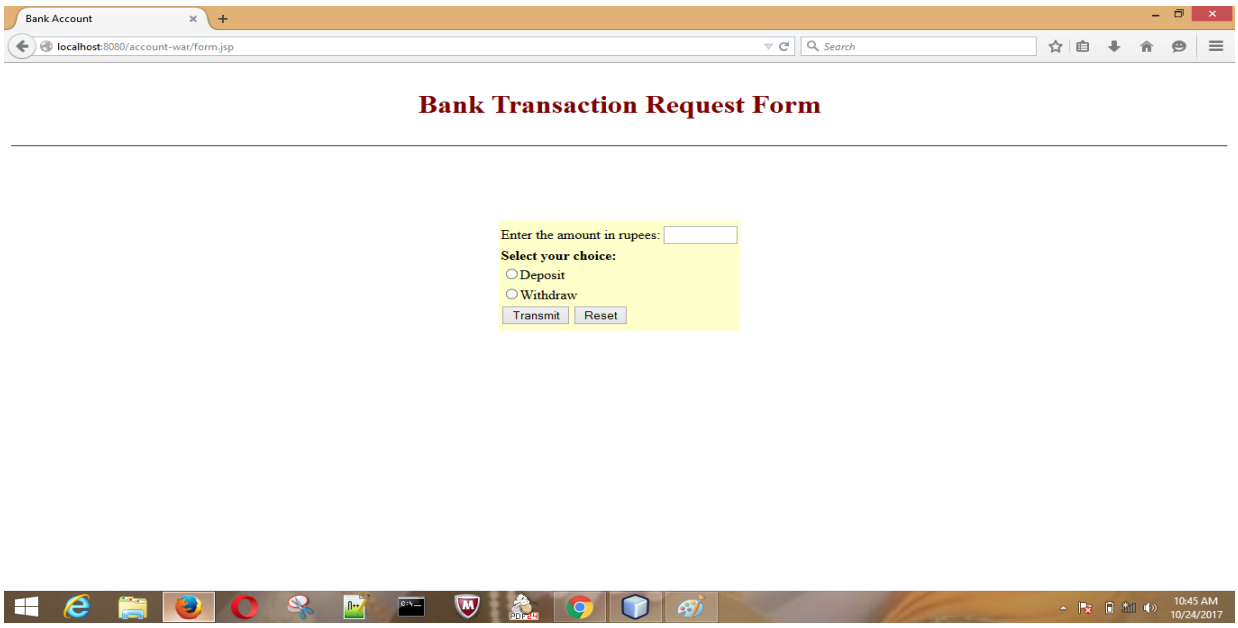


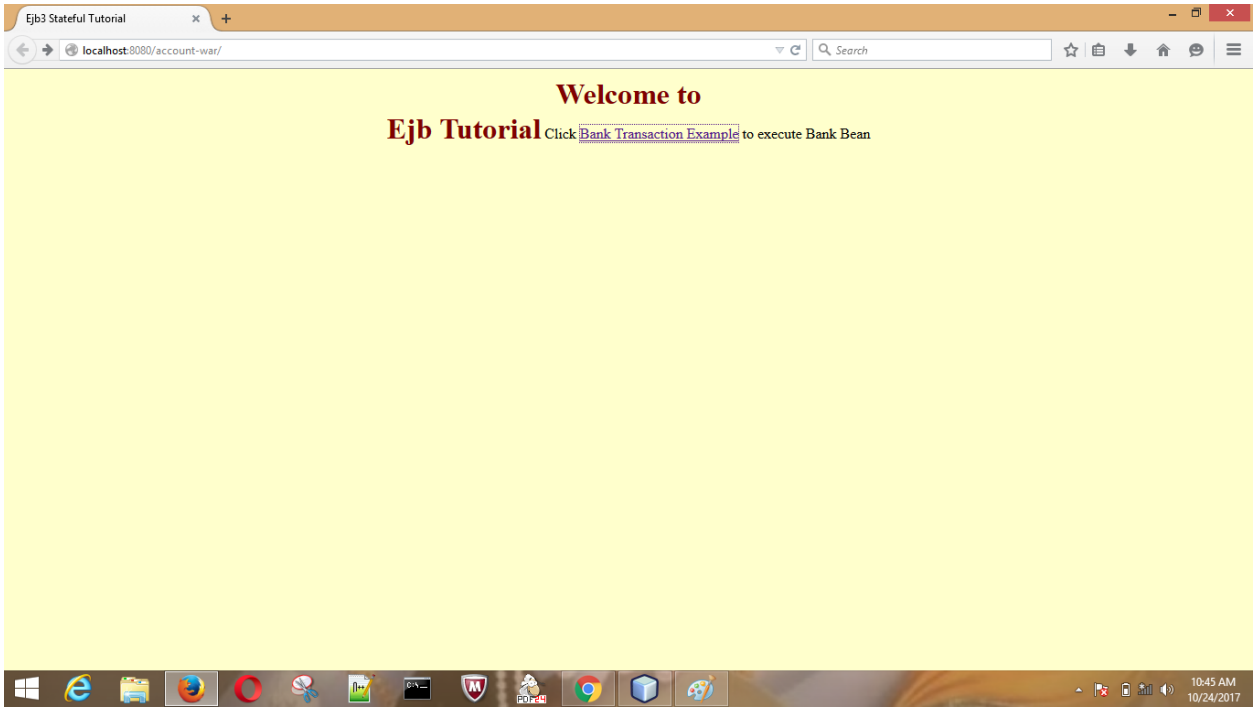

```
import ejbexample.AccountLocal;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.naming.Context;
```

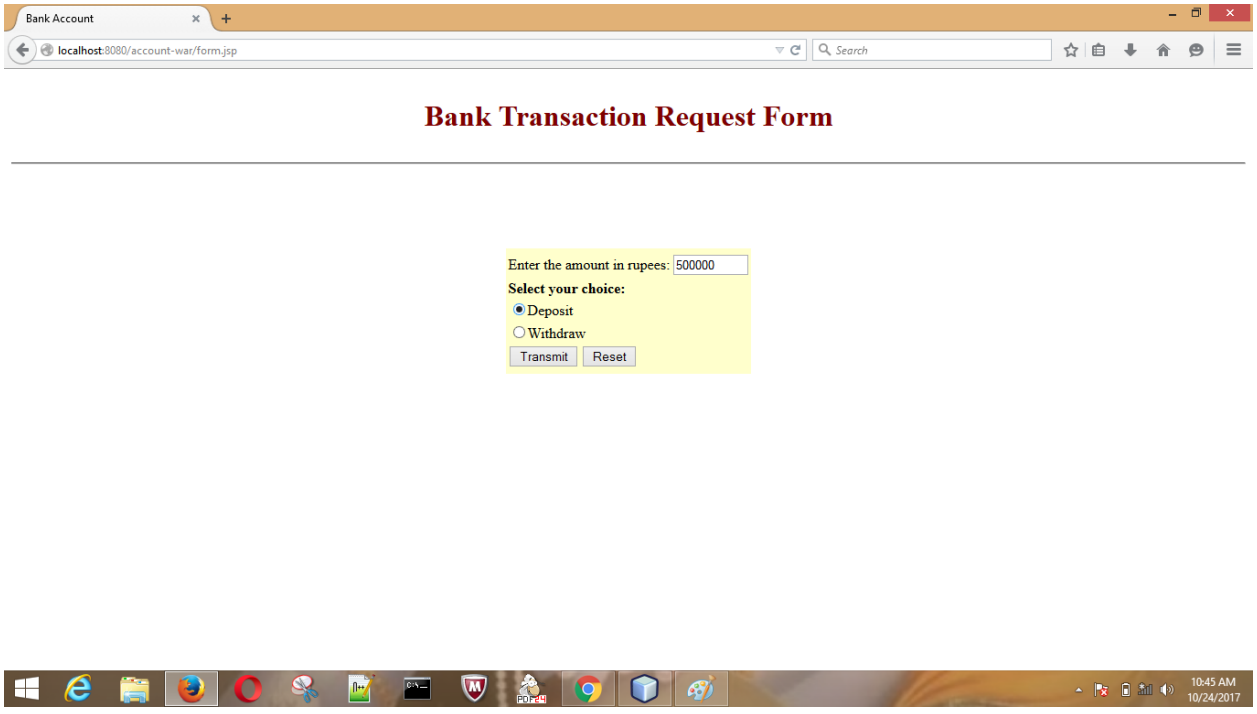
```
import javax.naming.InitialContext;
import javax.naming.NamingException;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class accservlet extends HttpServlet
{
AccountLocal account = lookupAccountLocal();
Public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException
{
response.setContentType("text/html;charset=UTF-8");
try (PrintWriter out = response.getWriter())
{
String s1=request.getParameter("amt");
String s2=request.getParameter("group1");
float bal=0;
if(s1!=null)
{
Float amt=new Float(s1);
if(s2.equals("dep"))
bal=account.deposit(amt);
else if(s2.equals("with"))
bal=account.withdraw(amt);
else
```

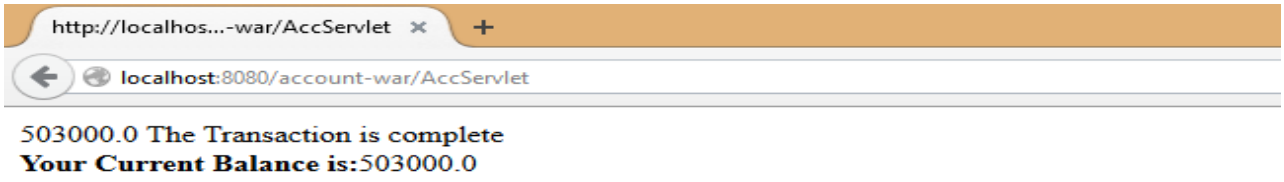
```
out.println("please enter your choice");
}
else
{
out.println("Please enter amount");
}
out.println("The transaction is complete");
out.println("your current balance is "+bal);
}
}
private AccountLocallookupAccountLocal()
{
try
{
Context c = new InitialContext();
return (AccountLocal);c.lookup("java:global/EnterpriseApplication2/EnterpriseApplication2-ejb/Account!ejbexample.AccountLocal");
}
 catch (NamingException ne)
{
Logger.getLogger(getClass().getName()).log(Level.SEVERE, "exception caught", ne);
throw new RuntimeException(ne);
}
}
```

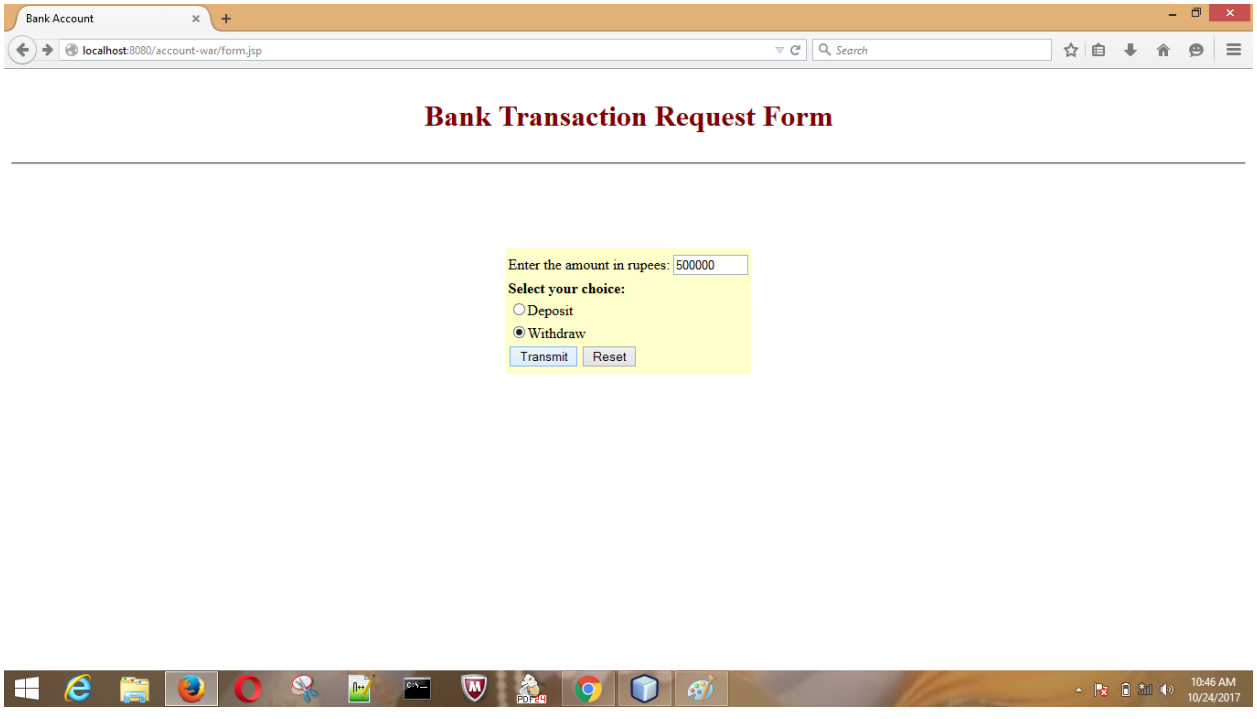
Output:

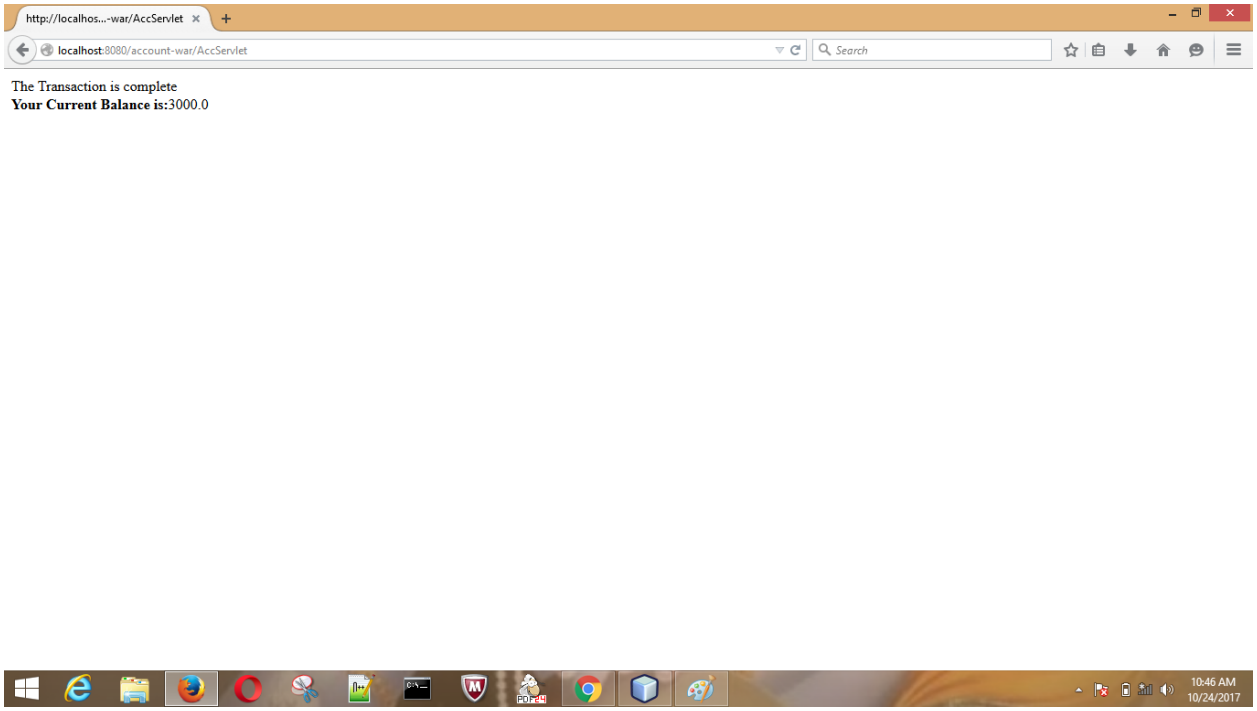

Bank Account
localhost:8080/account-war/form.jsp
Search
Bank Transaction Request Form
Enter the amount in rupees:
Select your choice:
Deposit
Withdraw
Transmit
Reset
10:45 AM
10/24/2017


Ejb3 Stateful Tutorial
localhost:8080/account-war/
Search
Welcome to
Ejb Tutorial
Click Bank Transaction Example to execute Bank Bean
10:45 AM
10/24/2017


Bank Account
localhost:8080/account-war/form.jsp
Search
Bank Transaction Request Form
Enter the amount in rupees: 500000
Select your choice:
Deposit
Withdraw
Transmit
Reset


http://localhos...-war/AccServlet
localhost:8080/account-war/AccServlet
503000.0 The Transaction is complete
Your Current Balance is:503000.0


Bank Account
localhost:8080/account-war/form.jsp
Search
Bank Transaction Request Form
Enter the amount in rupees: 500000
Select your choice:
Deposit
Withdraw
Transmit
Reset


http://localhos...-war/AccServlet
localhost:8080/account-war/AccServlet
Search
The Transaction is complete
Your Current Balance is:3000.0
10:46 AM
10/24/2017

9.Write a program to demonstrate message driven bean.

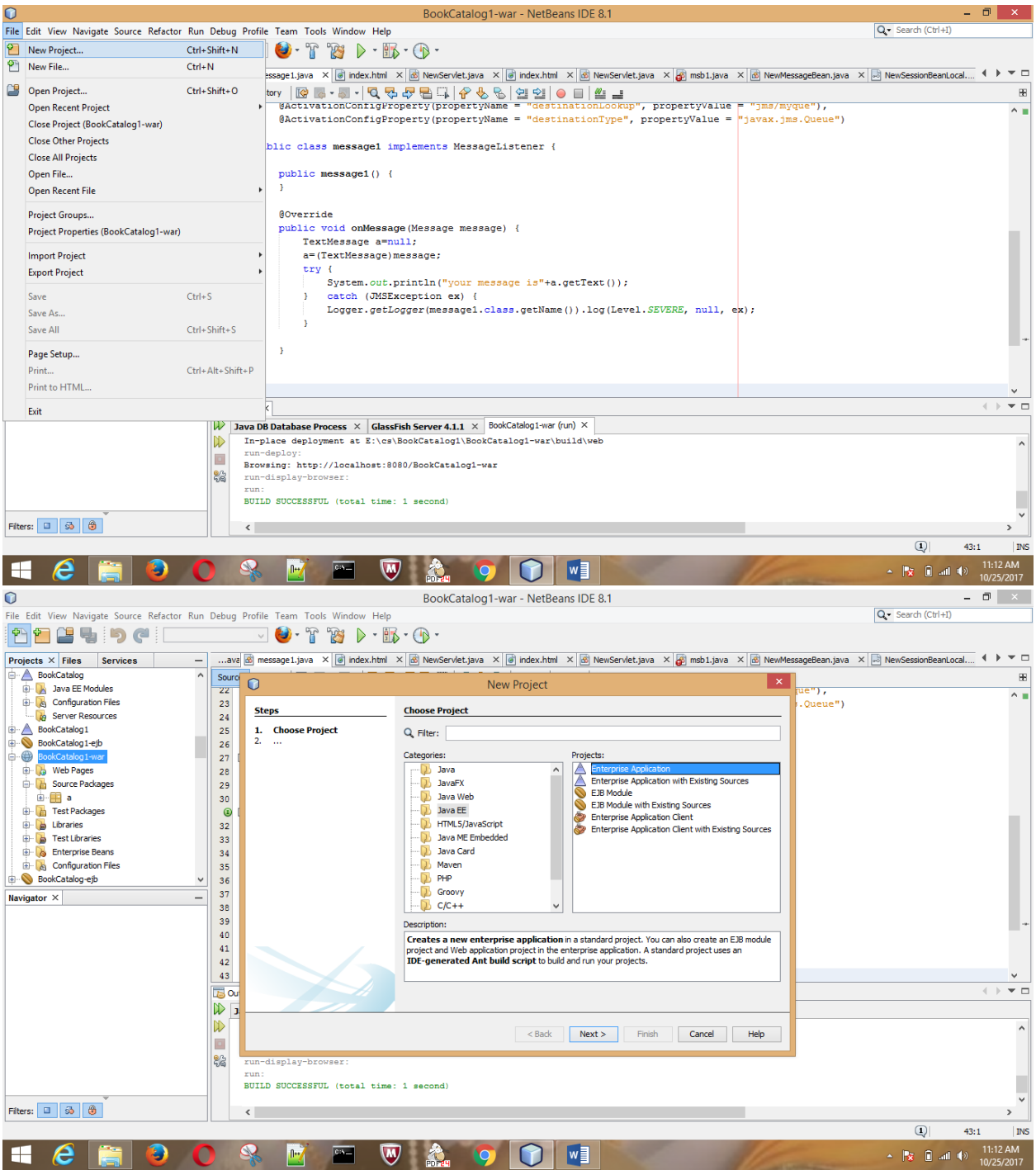
Aim:

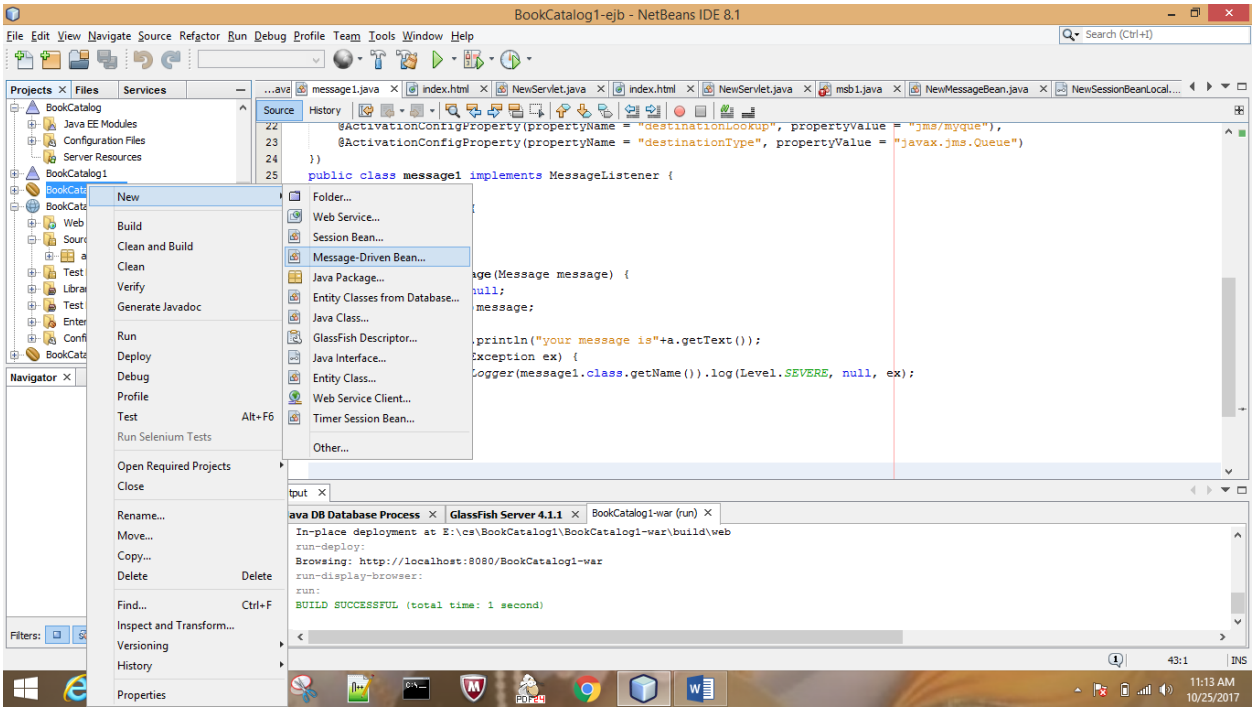
To demonstrate Message driven bean.

Source code:

File → New Project →JavaEE→ Enterprise Application

Right click on ejb module → New → Message Driven Bean





HelloBean.java:

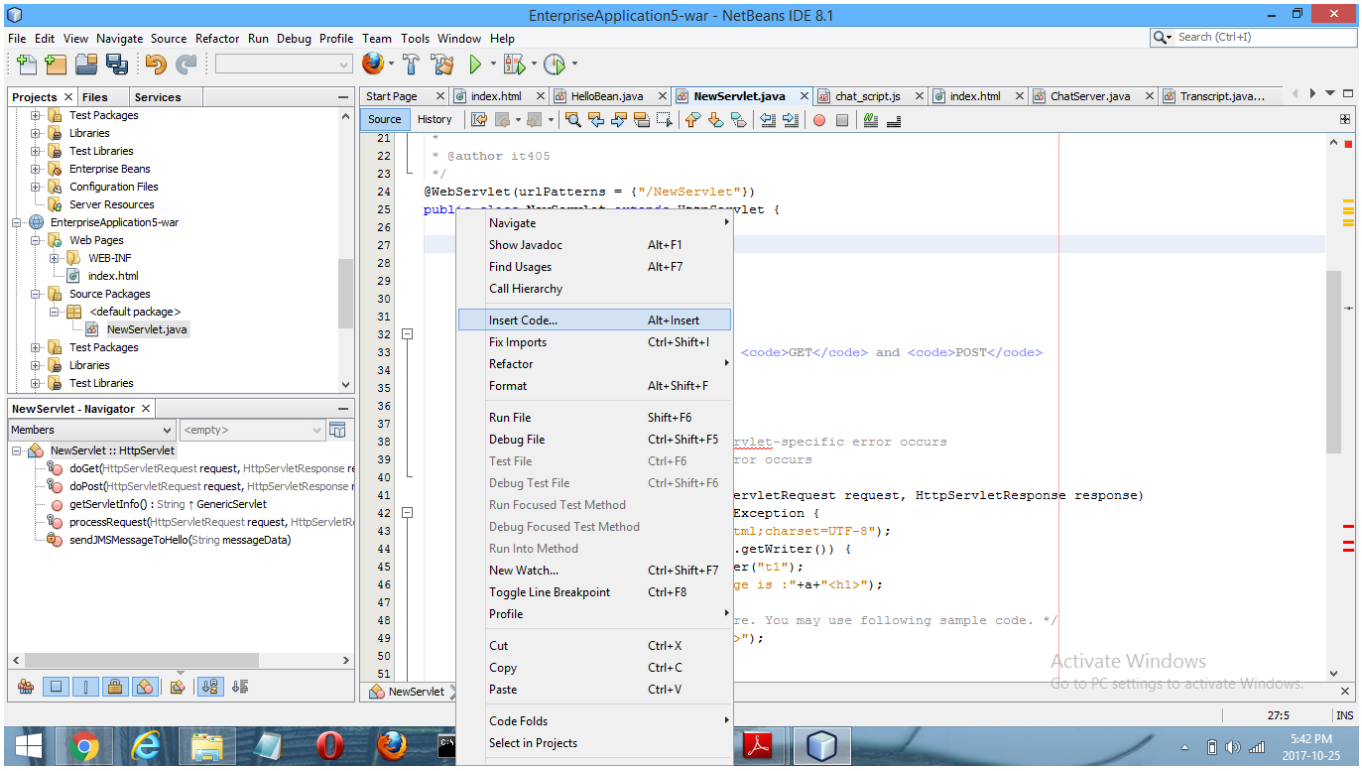
```
package bec;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.ejb.ActivationConfigProperty;

import javax.ejb.MessageDriven;

import javax.jms.JMSDestinationDefinition;

import javax.jms.JMSException;

import javax.jms.Message;

import javax.jms.MessageListener;

import javax.jms.TextMessage;

@JMSDestinationDefinition(name = "java:app/hello", interfaceName =
"javax.jms.Queue", resourceAdapter = "jmsra", destinationName = "hello")

@MessageDriven(activationConfig = {

@ActivationConfigProperty(propertyName = "destinationLookup", propertyValue =
"java:app/hello"),

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

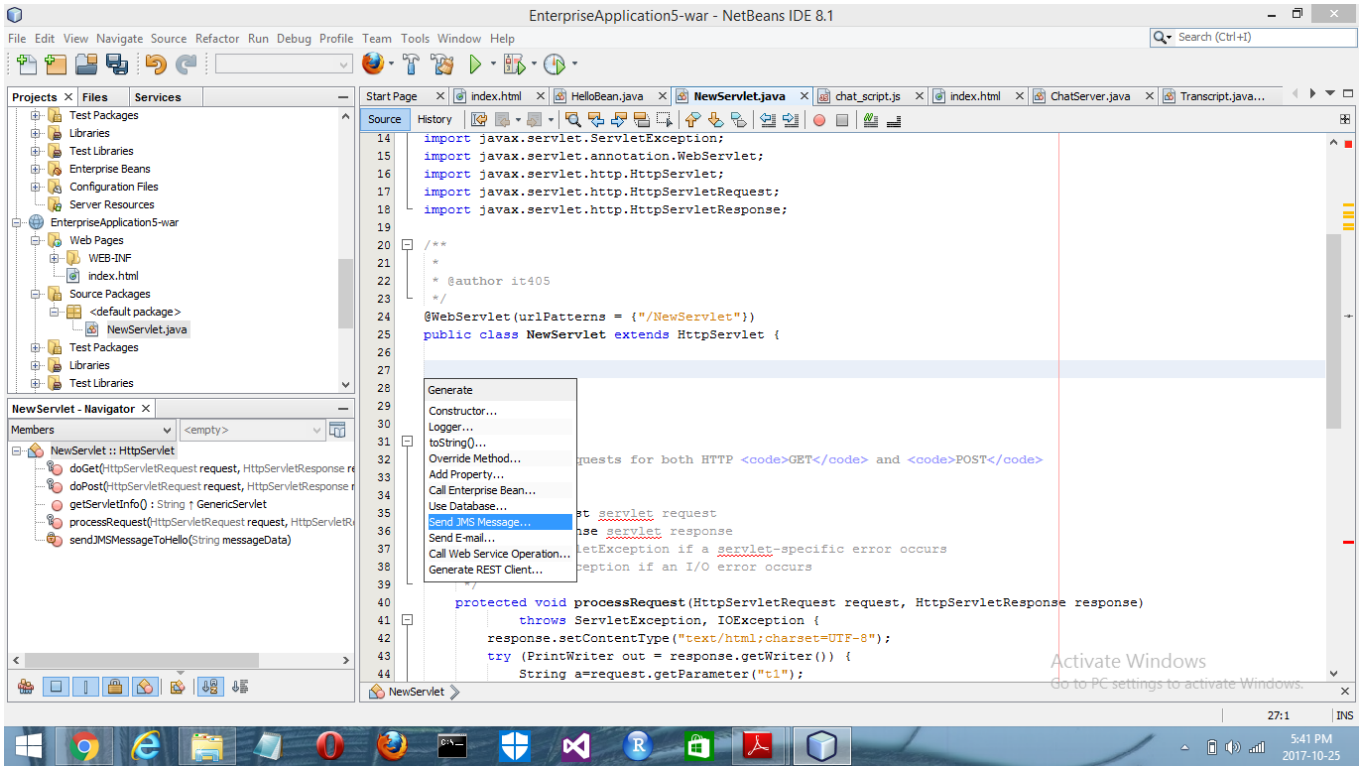
```
})
public class HelloBean implements MessageListener {
public HelloBean() {
}
@Override
public void onMessage(Message message) {
TextMessage a=null;
a=(TextMessage)message;
try {
System.out.println("your message is "+a.getText());
} catch (JMSException ex) {
Logger.getLogger(HelloBean.class.getName()).log(Level.SEVERE, null, ex);
}
}
}
```

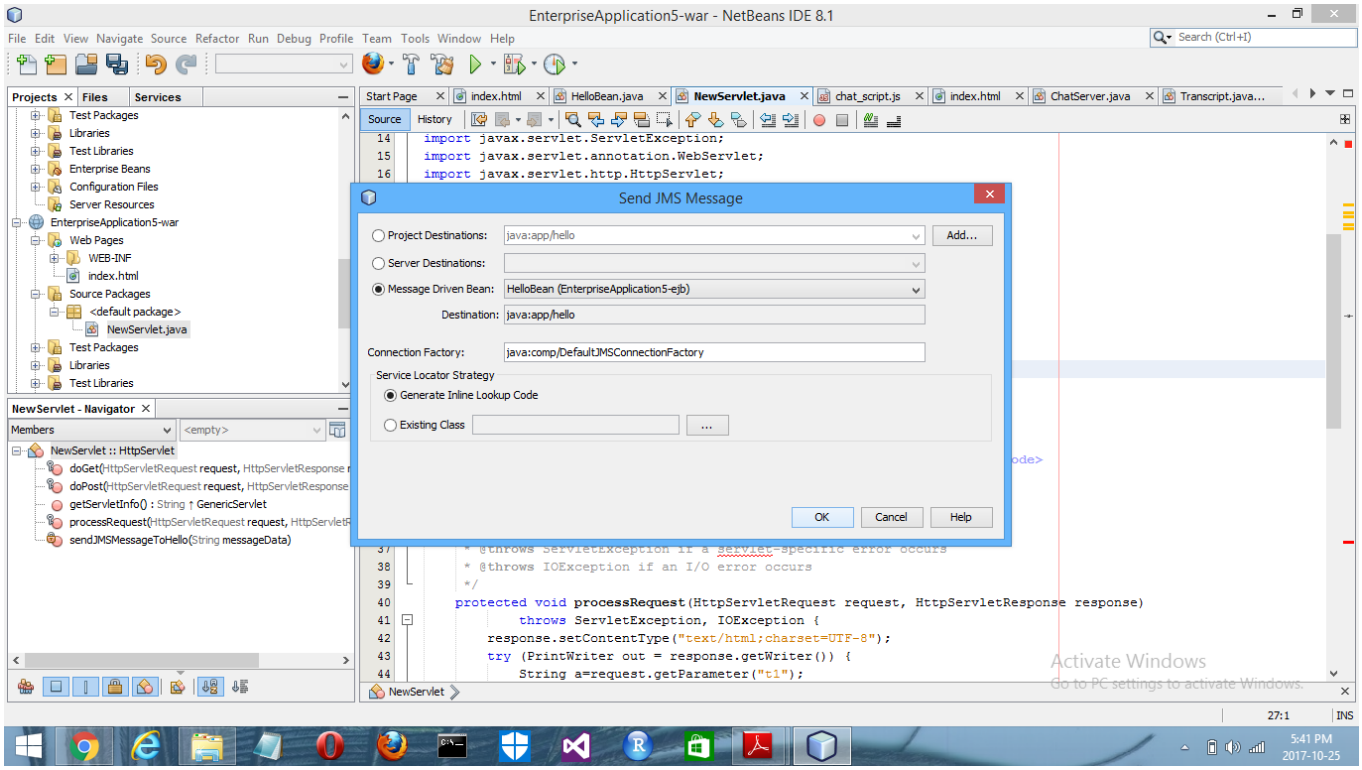
Index.html:

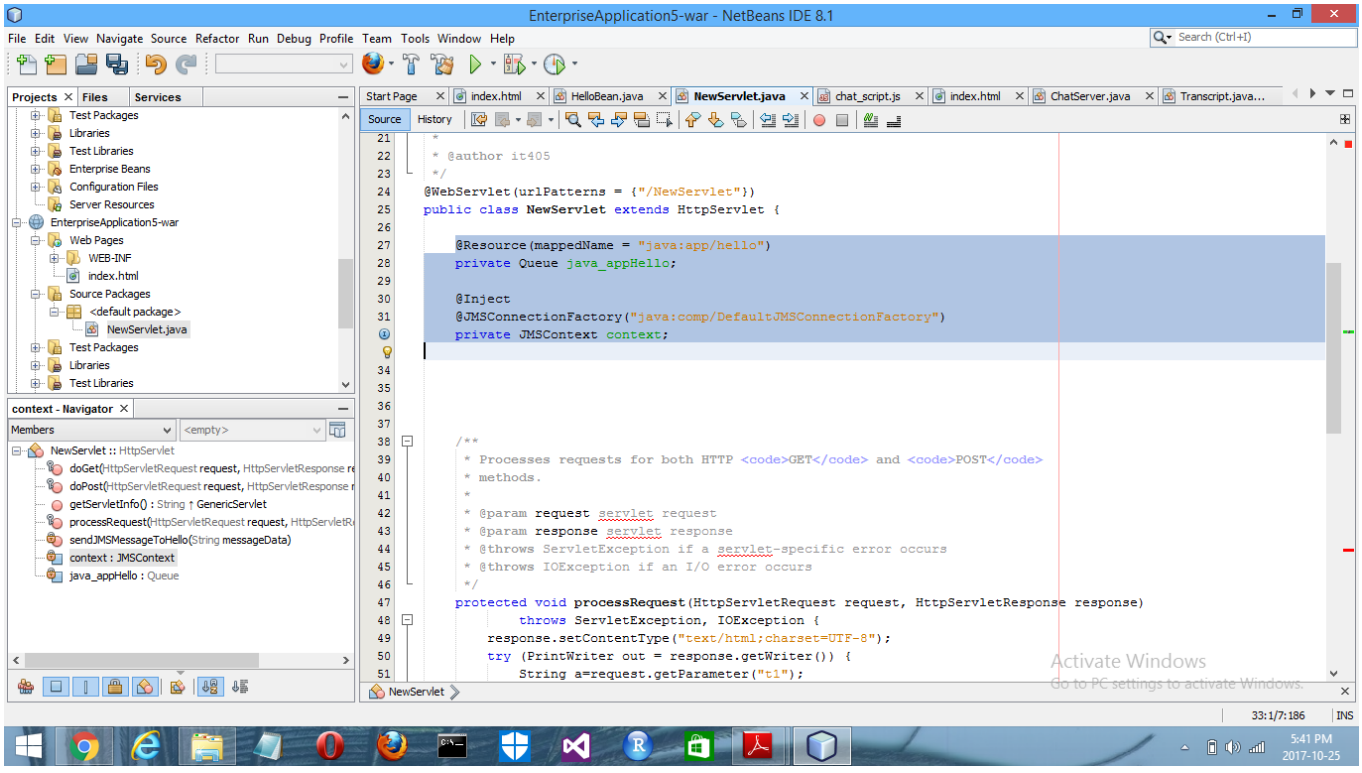
```
<html><head><title>MDB</title>
</head>
<body><h1>Enter your message :</h1>
<form action="NewServlet" method="post">
<input type="text" name="t1" />
<input type="submit" value="go">
</form>
</body>
</html>
```

NewServlet.java:


EnterpriseApplication5-war - NetBeans IDE 8.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects Files Services
Test Packages
Libraries
Test Libraries
Enterprise Beans
Configuration Files
Server Resources
EnterpriseApplication5-war
Web Pages
WEB-INF
index.html
Source Packages
<default package>
NewServlet.java
Test Packages
Libraries
Test Libraries
NewServlet - Navigator
Members
<empty>
NewServlet :: HttpServlet
doGet(HttpServletRequest request, HttpServletResponse re
doPost(HttpServletRequest request, HttpServletResponse r
getServletInfo() : String ↑ GenericServlet
processRequest(HttpServletRequest request, HttpServletR
sendJMSMessageToHello(String messageData)
Start Page
index.html
HelloBean.java
NewServlet.java
chat_script.js
index.html
ChatServer.java
Transcript.java...
Source
History
@author it405
@WebServlet(urlPatterns = {"/NewServlet"})
Navigate
Show Javadoc Alt+F1
Find Usages Alt+F7
Call Hierarchy
Insert Code... Alt+Insert
Fix Imports Ctrl+Shift+I
Refactor
Format Alt+Shift+F
Run File Shift+F6
Debug File Ctrl+Shift+F5
Test File Ctrl+F6
Debug Test File Ctrl+Shift+F6
Run Focused Test Method
Debug Focused Test Method
Run Into Method
New Watch... Ctrl+Shift+F7
Toggle Line Breakpoint Ctrl+F8
Profile
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Code Folds
Select in Projects
<code>GET</code> and <code>POST</code>
Activate Windows
Go to PC settings to activate Windows.
NewServlet
27:5
INS
5:42 PM
2017-10-25


EnterpriseApplication5-war - NetBeans IDE 8.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects Files Services
Test Packages
Libraries
Test Libraries
Enterprise Beans
Configuration Files
Server Resources
EnterpriseApplication5-war
Web Pages
WEB-INF
index.html
Source Packages
<default package>
NewServlet.java
Test Packages
Libraries
Test Libraries
NewServlet - Navigator
Members
<empty>
NewServlet :: HttpServlet
doGet(HttpServletRequest request, HttpServletResponse re
doPost(HttpServletRequest request, HttpServletResponse r
getServletInfo() : String ↑ GenericServlet
processRequest(HttpServletRequest request, HttpServletR
sendJMSMessageToHello(String messageData)
Start Page
index.html
HelloBean.java
NewServlet.java
chat_script.js
index.html
ChatServer.java
Transcript.java...
Source
History
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@author it405
@WebServlet(urlPatterns = {"/NewServlet"})
public class NewServlet extends HttpServlet {
Generate
Constructor...
Logger...
toString()...
Override Method...
Add Property...
Call Enterprise Bean...
Use Database...
Send JMS Message...
Send E-mail...
Call Web Service Operation...
Generate REST Client...
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
try (PrintWriter out = response.getWriter()) {
String a=request.getParameter("t1");
Activate Windows
Go to PC settings to activate Windows.
NewServlet
27:1
INS
5:41 PM
2017-10-25






```
import java.io.IOException;

import java.io.PrintWriter;

import javax.annotation.Resource;

import javax.inject.Inject;

import javax.jms.JMSConnectionFactory;

import javax.jms.JMSContext;
```

```
import javax.jms.Queue;
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(urlPatterns = {"/NewServlet"})
public class NewServlet extends HttpServlet {
@Resource(mappedName = "java:app/hello")
private Queue java_appHello;
@Inject
@JMSConnectionFactory("java:comp/DefaultJMSConnectionFactory")
private JMSContext context;
protected void processRequest(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
response.setContentType("text/html;charset=UTF-8");
try (PrintWriter out = response.getWriter()) {
String a=request.getParameter("t1");
out.println("<h1>Your message is :"+a+"<h1>");
sendJMSMessageToHello(a);
}
}
private void sendJMSMessageToHello(String messageData) {
context.createProducer().send(java_appHello, messageData);
}
```

Output:


http://localhost...rise-war/servlet
JSP Page
localhost:35469/EnterpriseMDriven-war/
Enter your message
hello
go


http://localhost...rise-war/servlet
http://localhost...ven-war/servlet
localhost:35469/EnterpriseMDriven-war/servlet
Your message is hello

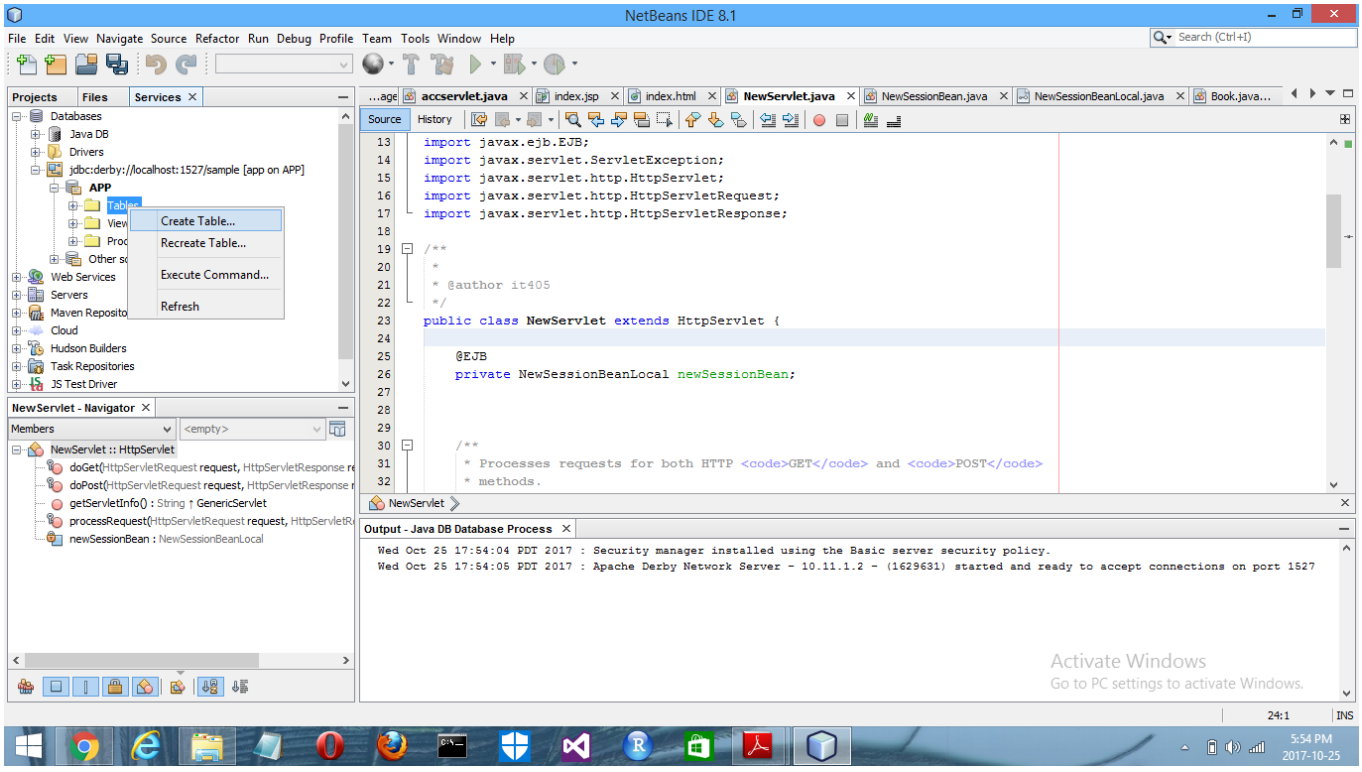
10.Write a program to demonstrate entity bean.

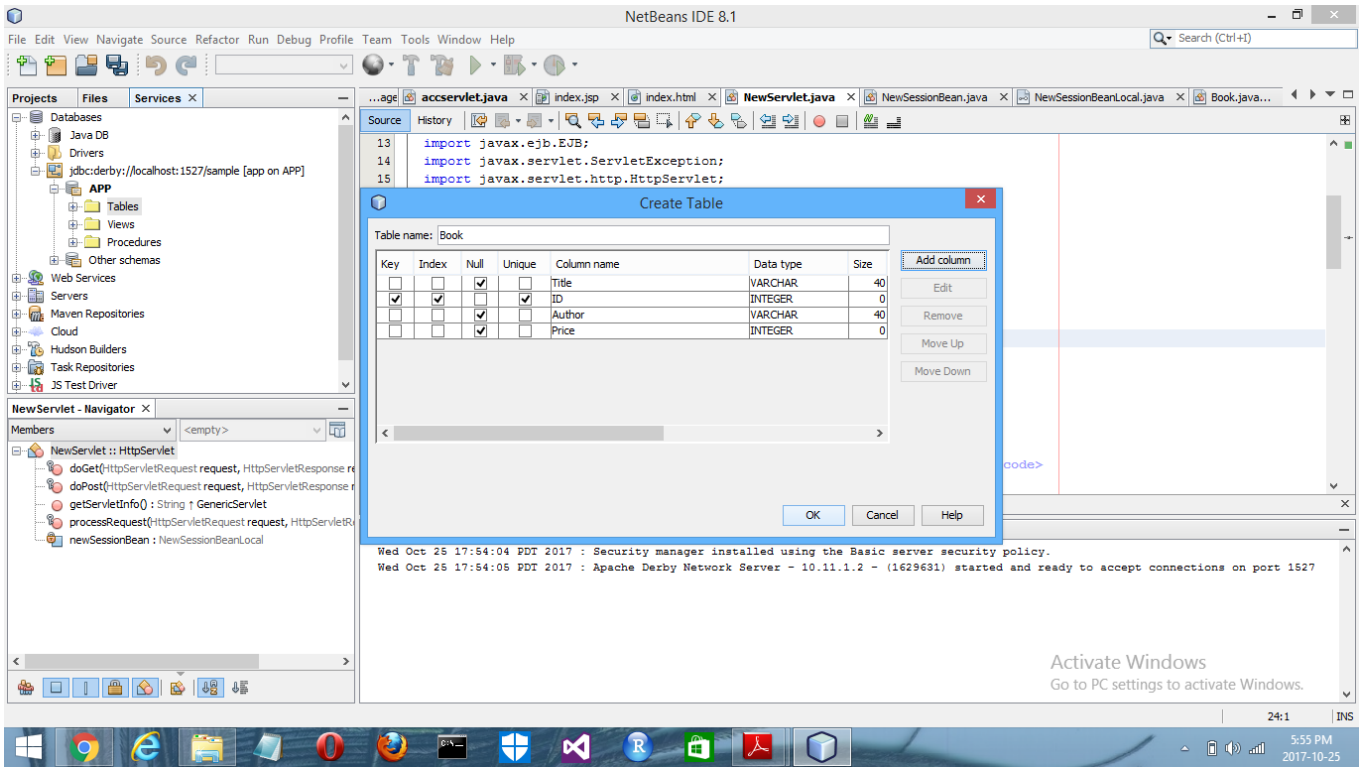
Aim:

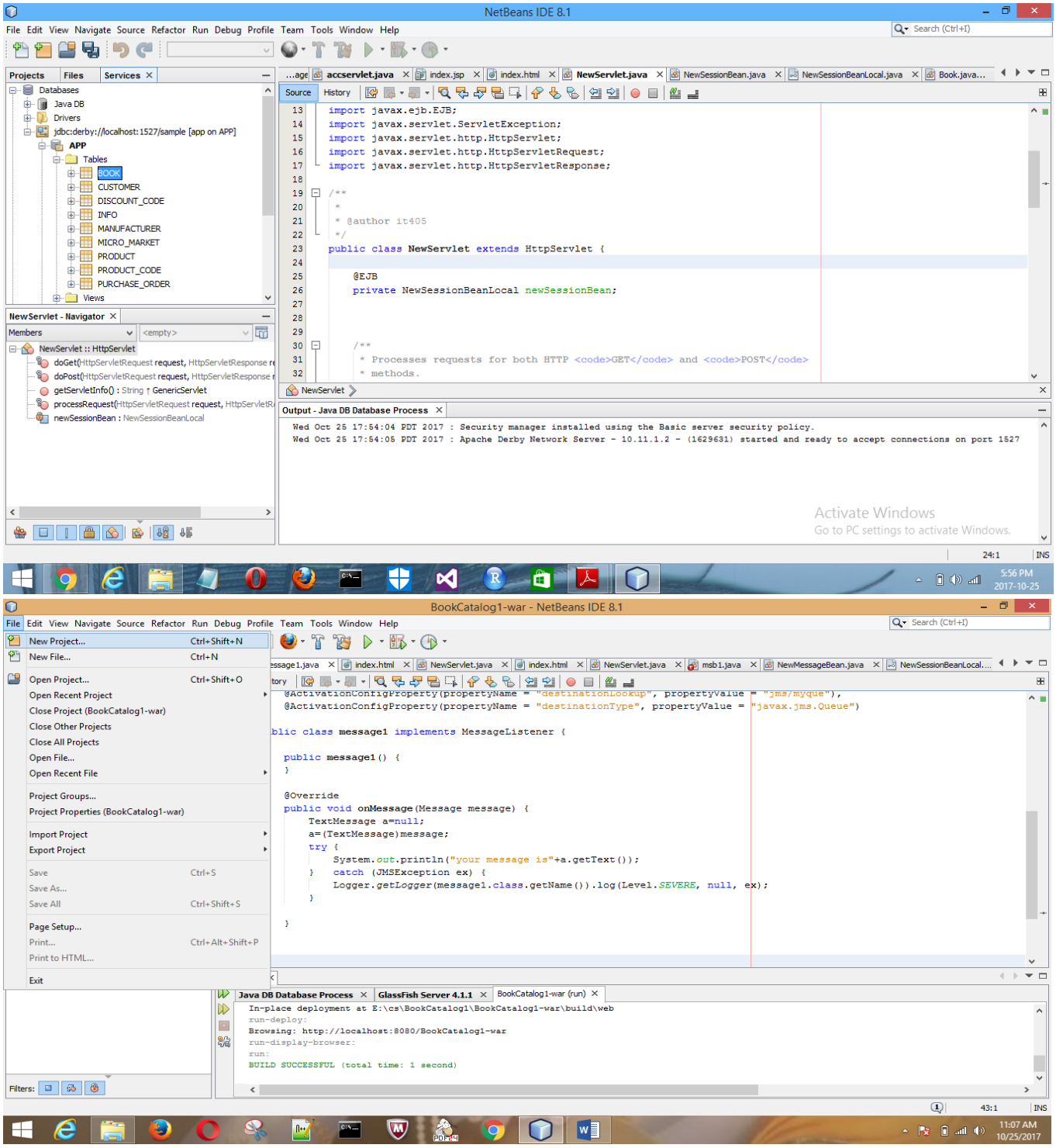
To demonstrate entity bean.

Source code:

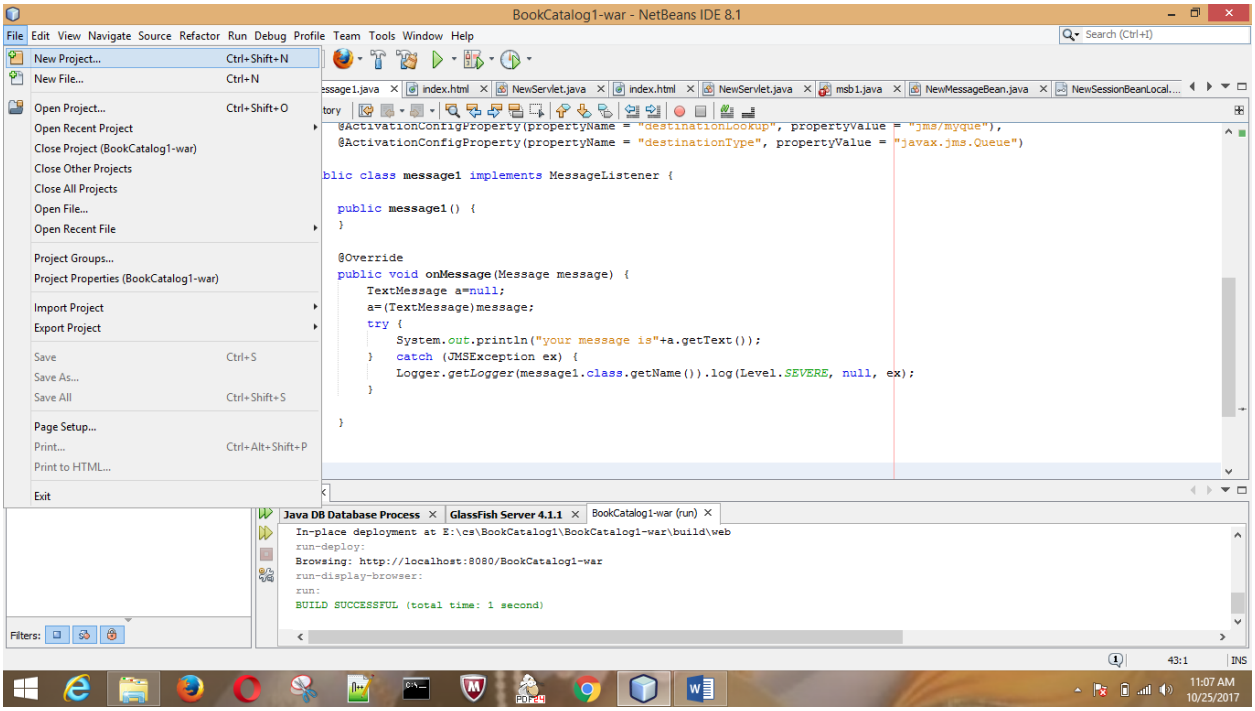
Create a table named ‘Book’ in a sample database .

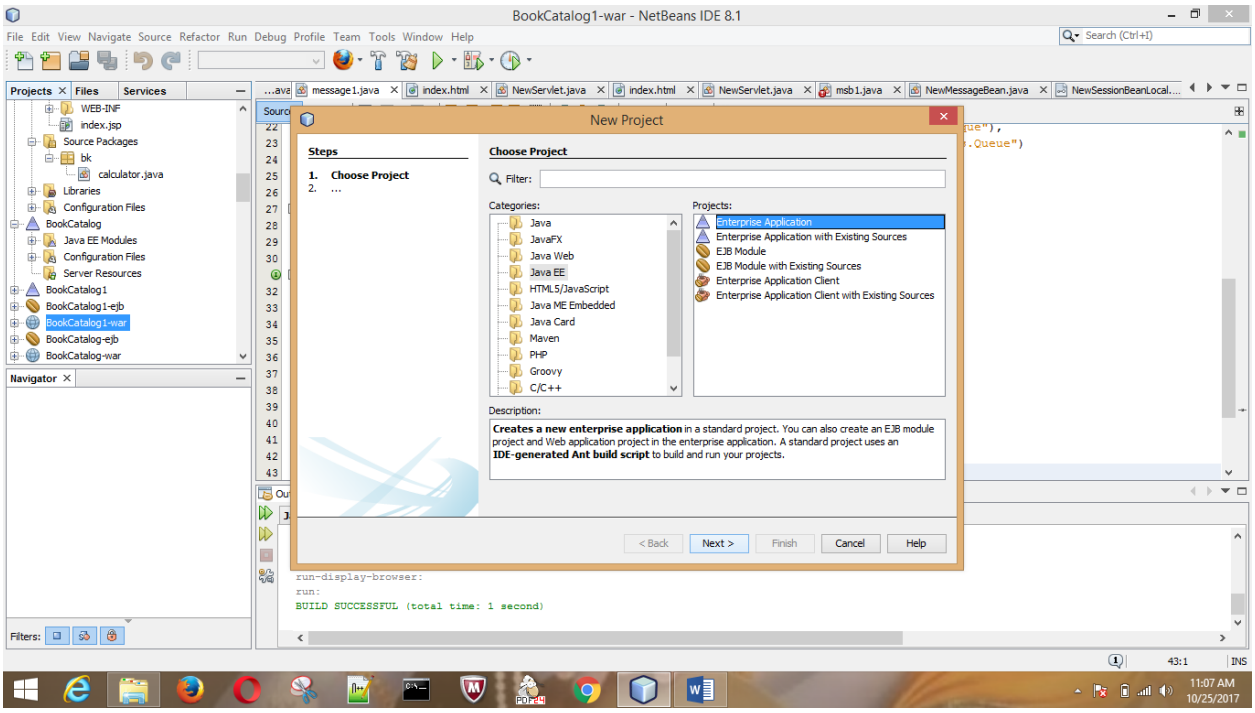


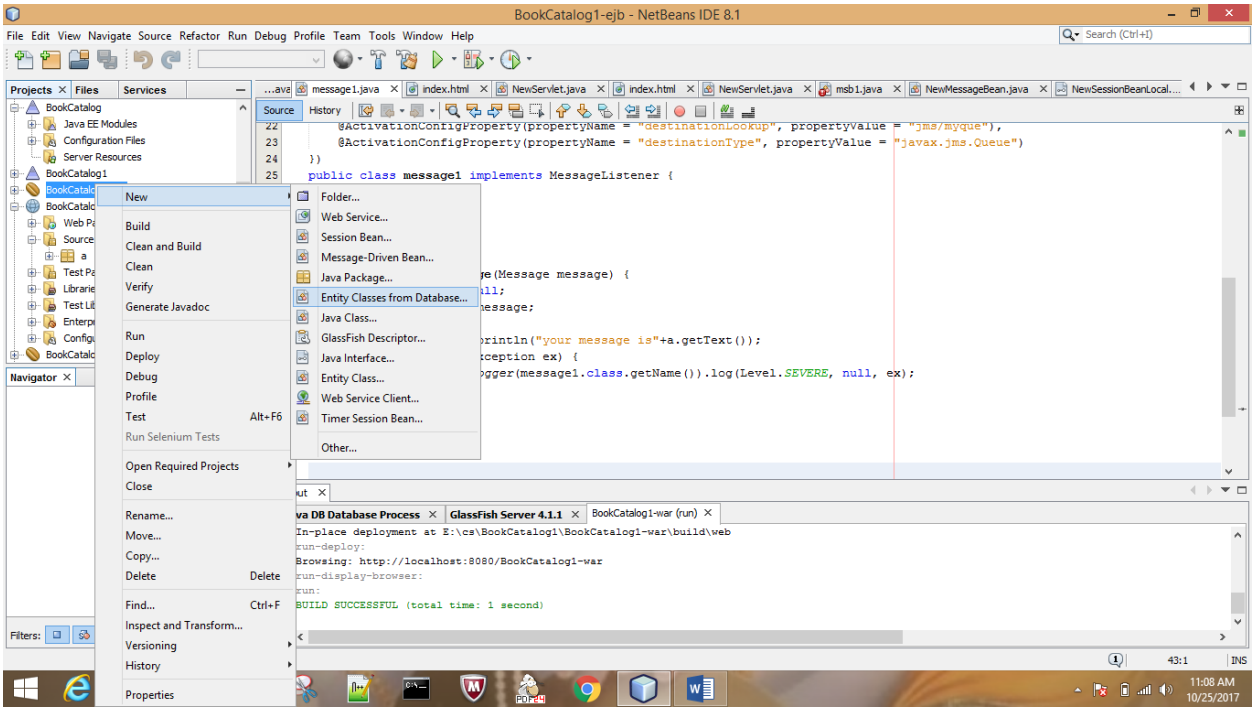


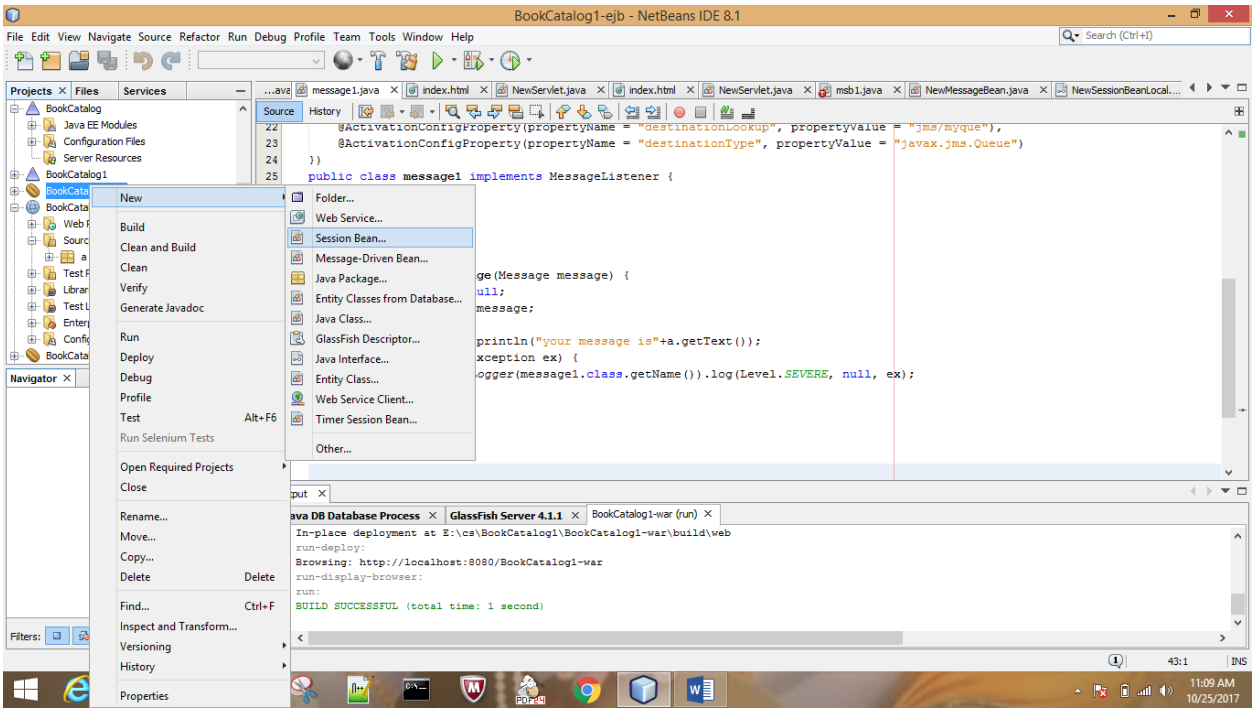

NetBeans IDE 8.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Files Services
Databases
Java DB
Drivers
jdbc:derby://localhost:1527/sample [app on APP]
APP
Tables
BOOK
CUSTOMER
DISCOUNT_CODE
INFO
MANUFACTURER
MICRO_MARKET
PRODUCT
PRODUCT_CODE
PURCHASE_ORDER
Views
NewServlet - Navigator
NewServlet :: HttpServlet
doGet(HttpServletRequest request, HttpServletResponse
doPost(HttpServletRequest request, HttpServletResponse
getServletInfo() : String ↑ GenericServlet
processRequest(HttpServletRequest request, HttpServletR
newSessionBean : NewSessionBeanLocal
accservlet.java
index.jsp
index.html
NewServlet.java
NewSessionBean.java
NewSessionBeanLocal.java
Book.java...
Source History
13 import javax.ejb.EJB;
14 import javax.servlet.ServletException;
15 import javax.servlet.http.HttpServlet;
16 import javax.servlet.http.HttpServletRequest;
17 import javax.servlet.http.HttpServletResponse;
18
19 /**
20 *
21 * @author it405
22 */
23 public class NewServlet extends HttpServlet {
24
25 @EJB
26 private NewSessionBeanLocal newSessionBean;
27
28
29
30 /**
31 * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
32 * methods.
NewServlet
Output - Java DB Database Process
Wed Oct 25 17:54:04 PDT 2017 : Security manager installed using the Basic server security policy.
Wed Oct 25 17:54:05 PDT 2017 : Apache Derby Network Server - 10.11.1.2 - (1629631) started and ready to accept connections on port 1527
Activate Windows
Go to PC settings to activate Windows.
24:1 INS
5:56 PM
2017-10-25

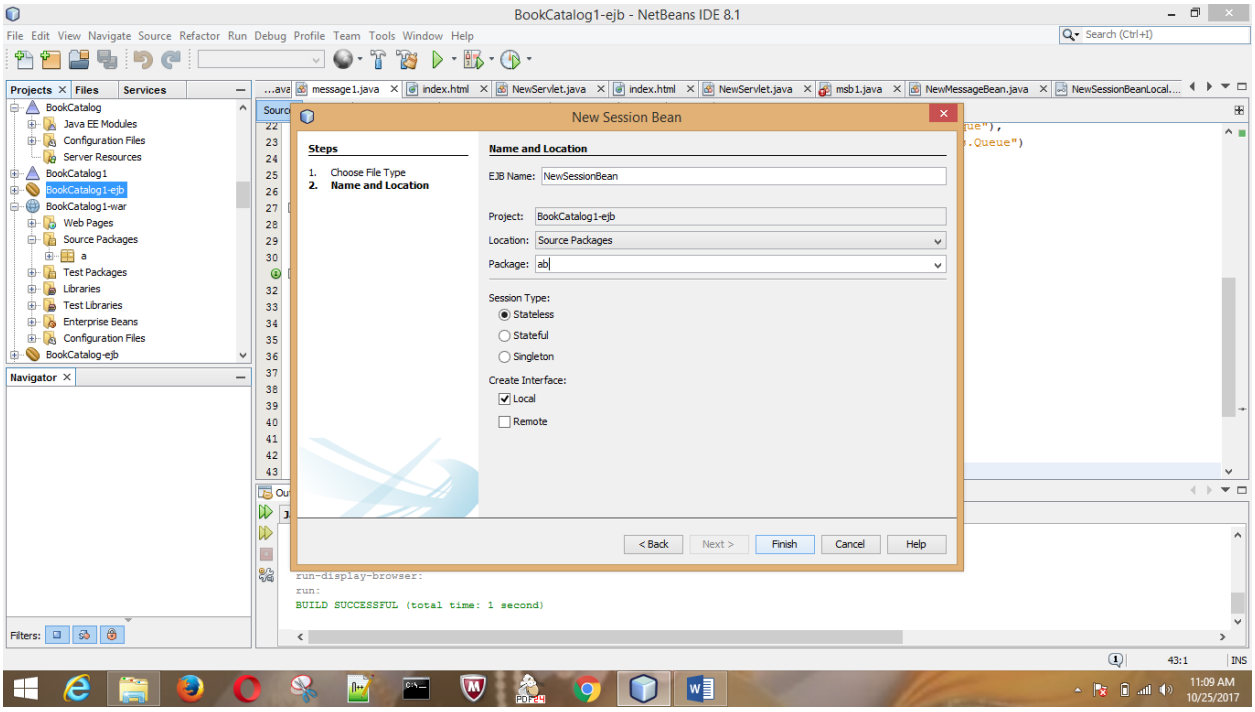
BookCatalog1-war - NetBeans IDE 8.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
New Project... Ctrl+Shift+N
New File... Ctrl+N
Open Project... Ctrl+Shift+O
Open Recent Project
Close Project (BookCatalog1-war)
Close Other Projects
Close All Projects
Open File...
Open Recent File
Project Groups...
Project Properties (BookCatalog1-war)
Import Project
Export Project
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
Page Setup...
Print... Ctrl+Alt+Shift+P
Print to HTML...
Exit
@ActivationConfigProperty(propertyName = "destinationLookup", propertyValue = "jms/myque"),
@ActivationConfigProperty(propertyName = "destinationType", propertyValue = "javax.jms.Queue")
blic class message1 implements MessageListener {
public message1() {
}
@Override
public void onMessage(Message message) {
TextMessage a=null;
a=(TextMessage)message;
try {
System.out.println("your message is"+a.getText());
} catch (JMSException ex) {
Logger.getLogger(message1.class.getName()).log(Level.SEVERE, null, ex);
}
}
Java DB Database Process
GlassFish Server 4.1.1
BookCatalog1-war (run)
In-place deployment at E:\cs\BookCatalog1\BookCatalog1-war\build\web
run-deploy:
Browsing: http://localhost:8080/BookCatalog1-war
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 1 second)
Filters:
43:1 INS
11:07 AM
10/25/2017

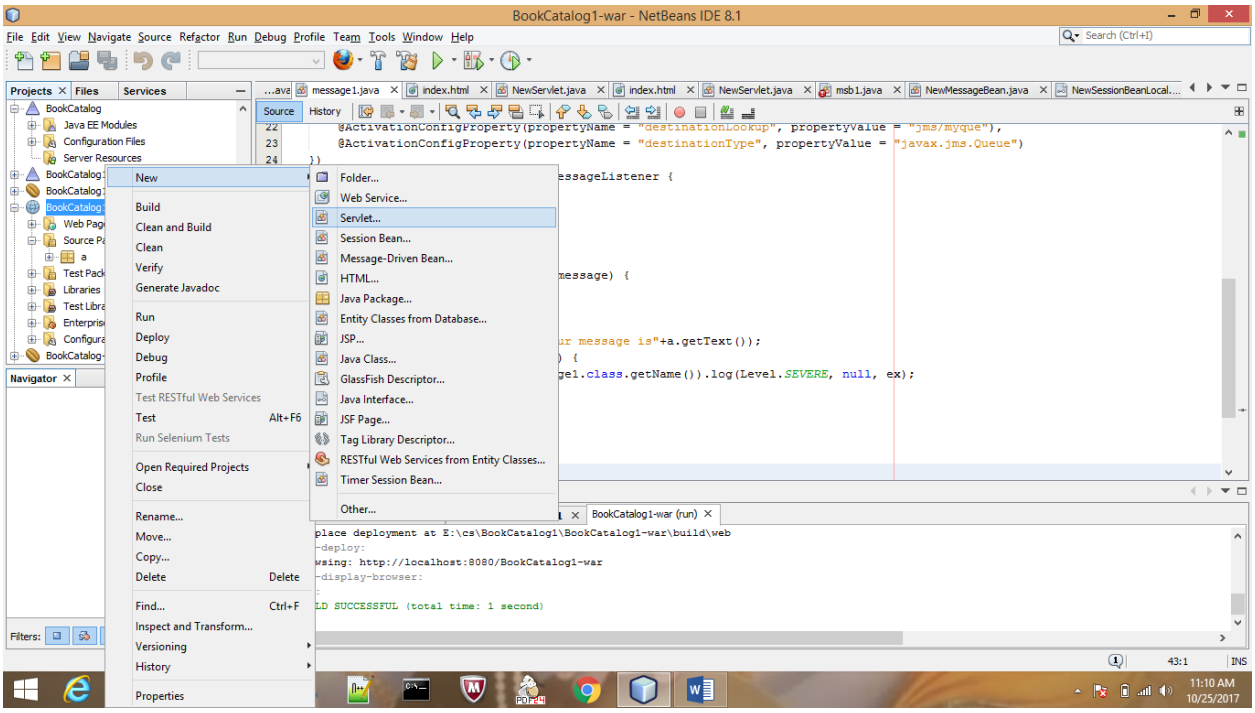

BookCatalog1-war - NetBeans IDE 8.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
New Project... Ctrl+Shift+N
New File... Ctrl+N
Open Project... Ctrl+Shift+O
Open Recent Project
Close Project (BookCatalog1-war)
Close Other Projects
Close All Projects
Open File...
Open Recent File
Project Groups...
Project Properties (BookCatalog1-war)
Import Project
Export Project
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
Page Setup...
Print... Ctrl+Alt+Shift+P
Print to HTML...
Exit
@ActivationConfigProperty(propertyName = "destinationLookup", propertyValue = "jms/myque"),
@ActivationConfigProperty(propertyName = "destinationType", propertyValue = "javax.jms.Queue")
blic class message1 implements MessageListener {
public message1() {
}
@Override
public void onMessage(Message message) {
TextMessage a=null;
a=(TextMessage)message;
try {
System.out.println("your message is"+a.getText());
} catch (JMSException ex) {
Logger.getLogger(message1.class.getName()).log(Level.SEVERE, null, ex);
}
}
Java DB Database Process
GlassFish Server 4.1.1
BookCatalog1-war (run)
In-place deployment at E:\cs\BookCatalog1\BookCatalog1-war\build\web
run-deploy:
Browsing: http://localhost:8080/BookCatalog1-war
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 1 second)
Filters:
43:1 INS
11:07 AM
10/25/2017


BookCatalog1-war - NetBeans IDE 8.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Files Services
WEB-INF
index.jsp
Source Packages
bk
calculator.java
Libraries
Configuration Files
BookCatalog
Java EE Modules
Configuration Files
Server Resources
BookCatalog1
BookCatalog1-ejb
BookCatalog1-war
BookCatalog-ejb
BookCatalog-war
Navigator
New Project
Steps
1. Choose Project
2. ...
Choose Project
Filter:
Categories:
Java
JavaFX
Java Web
Java EE
HTML5/JavaScript
Java ME Embedded
Java Card
Maven
PHP
Groovy
C/C++
Projects:
Enterprise Application
Enterprise Application with Existing Sources
EJB Module
EJB Module with Existing Sources
Enterprise Application Client
Enterprise Application Client with Existing Sources
Description:
Creates a new enterprise application in a standard project. You can also create an EJB module project and Web application project in the enterprise application. A standard project uses an IDE-generated Ant build script to build and run your projects.
< Back
Next >
Finish
Cancel
Help
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 1 second)
Filters:
43:1 INS
11:07 AM
10/25/2017


BookCatalog1-ejb - NetBeans IDE 8.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Files Services
BookCatalog
Java EE Modules
Configuration Files
Server Resources
BookCatalog1
New
Build
Clean and Build
Clean
Verify
Generate Javadoc
Run
Deploy
Debug
Profile
Test Alt+F6
Run Selenium Tests
Open Required Projects
Close
Rename...
Move...
Copy...
Delete Delete
Find... Ctrl+F
Inspect and Transform...
Versioning
History
Properties
Folder...
Web Service...
Session Bean...
Message-Driven Bean...
Java Package...
Entity Classes from Database...
Java Class...
GlassFish Descriptor...
Java Interface...
Entity Class...
Web Service Client...
Timer Session Bean...
Other...
Source History
@ActivationConfigProperty(propertyName = "destinationLookup", propertyValue = "jms/myque"),
@ActivationConfigProperty(propertyName = "destinationType", propertyValue = "javax.jms.Queue")
public class message1 implements MessageListener {
Java DB Database Process
GlassFish Server 4.1.1
BookCatalog1-war (run)
In-place deployment at E:\cs\BookCatalog1\BookCatalog1-war\build\web
run-deploy:
Browsing: http://localhost:8080/BookCatalog1-war
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 1 second)
11:08 AM
10/25/2017


BookCatalog1-ejb - NetBeans IDE 8.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Projects Files Services
BookCatalog
Java EE Modules
Configuration Files
Server Resources
BookCatalog1
New
Build
Clean and Build
Clean
Verify
Generate Javadoc
Run
Deploy
Debug
Profile
Test Alt+F6
Run Selenium Tests
Open Required Projects
Close
Rename...
Move...
Copy...
Delete Delete
Find... Ctrl+F
Inspect and Transform...
Versioning
History
Properties
Folder...
Web Service...
Session Bean...
Message-Driven Bean...
Java Package...
Entity Classes from Database...
Java Class...
GlassFish Descriptor...
Java Interface...
Entity Class...
Web Service Client...
Timer Session Bean...
Other...
Source History
@ActivationConfigProperty(propertyName = "destinationLookup", propertyValue = "jms/myque"),
@ActivationConfigProperty(propertyName = "destinationType", propertyValue = "javax.jms.Queue")
public class message1 implements MessageListener {
Java DB Database Process
GlassFish Server 4.1.1
BookCatalog1-war (run)
In-place deployment at E:\cs\BookCatalog1\BookCatalog1-war\build\web
run-deploy:
Browsing: http://localhost:8080/BookCatalog1-war
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 1 second)
11:09 AM
10/25/2017





File → New Project →JavaEE→ Enterprise Application

Index.html:

<html><head>

<title>entitybean</title>

</head><body>

<form action="index.jsp" method="post">

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

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

Index.jsp:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html><head>
<title>entity</title>
</head><body>
<form action="NewServlet" method="get">
Enter ID :<input type="text" name="t0" >
Enter title :<input type="text" name="t1" >
Enter Author :<input type="text" name="t2" >
Enter price :<input type="text" name="t3" >
<input type="submit" value="submit" >
<input type="submit" value="reset" >
</form>
</body>
</html>
```

Right click on ejb module → New → Entity Classes from database → Add the required database

Book.java:

```
package bec;
import java.io.Serializable;
import javax.persistence.Basic;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.NamedQueries;
```

```
import javax.persistence.NamedQuery;

import javax.persistence.Table;

import javax.validation.constraints.NotNull;

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.findById", query = "SELECT b FROM Book b WHERE
b.id = :id"),

@NamedQuery(name = "Book.findByTitle", query = "SELECT b FROM Book b
WHERE b.title = :title"),

@NamedQuery(name = "Book.findByAuthor", query = "SELECT b FROM Book b
WHERE b.author = :author"),

@NamedQuery(name = "Book.findByPrice", query = "SELECT b FROM Book b
WHERE b.price = :price")})

public class Book implements Serializable {

private static final long serialVersionUID = 1L;

@Id

@Basic(optional = false)

@NotNull

@Column(name = "ID")

private Integer id;

@Size(max = 40)

@Column(name = "TITLE")

private String title;
```

```
@Size(max = 40)
@Column(name = "AUTHOR")
private String author;
@Column(name = "PRICE")
private Integer price;
public Book() {
}
public Book(Integer id) {
this.id = id;
}
public Book(int id, String title, String author, int price) {
this.id=id;
this.title=title;
this.author=author;
this.price=price;
}
public Integer getId() {
return id;
}
public void setId(Integer id) {
this.id = id;
}
public String getTitle() {
return title;
}
public void setTitle(String title) {
```

```
this.title = title;
}
public String getAuthor() {
return author;
}
public void setAuthor(String author) {
this.author = author;
}
public Integer getPrice() {
return price;
}
public void setPrice(Integer price) {
this.price = price;
}
@Override
public inthashCode() {
int hash = 0;
hash += (id != null ? id.hashCode() : 0);
return hash;
}
@Override
public booleanequals(Object object) {
if (!(object instanceof Book)) {
return false;
}
Book other = (Book) 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 "bec.Book[ id=" + id + " ]";
}
}
```

Right click on ejb module → New →SessionBean (Select Local Mode)

NewSessionBeanLocal.java:

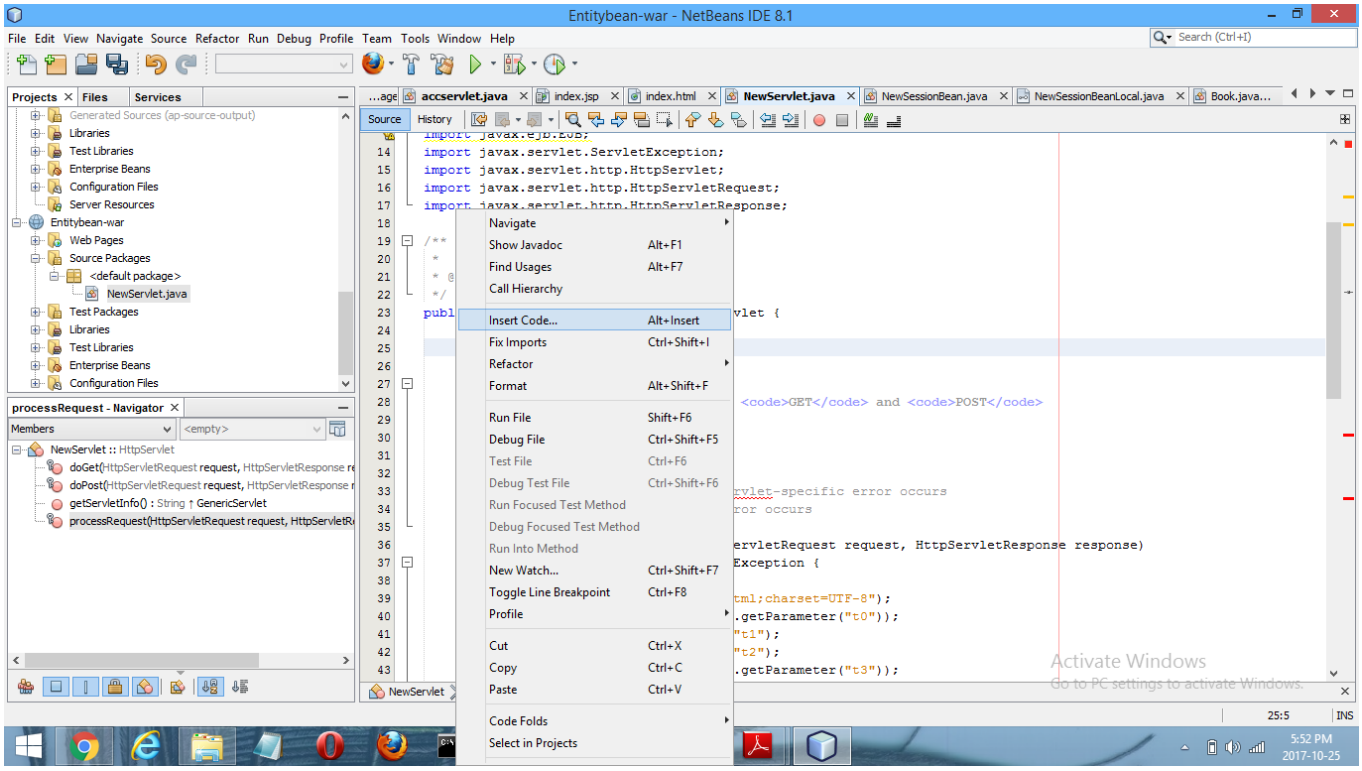
```
package ejb;
import bec.Book;
import java.util.Collection;
import javax.ejb.Local;
@Local
public interface NewSessionBeanLocal
{
public void addBook(int id, String title, String author, int price);
public Collection<Book>getAllBooks();
}
```

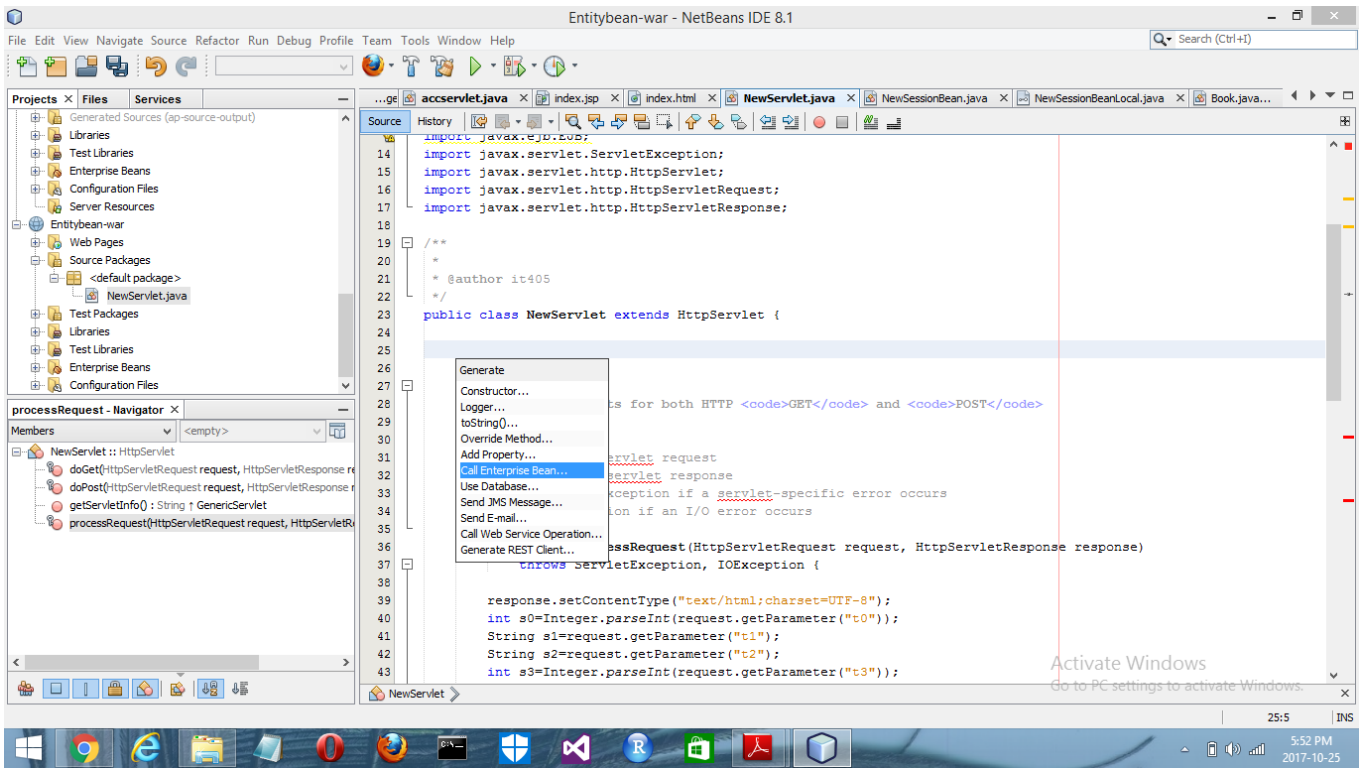
NewSessionBean.java:

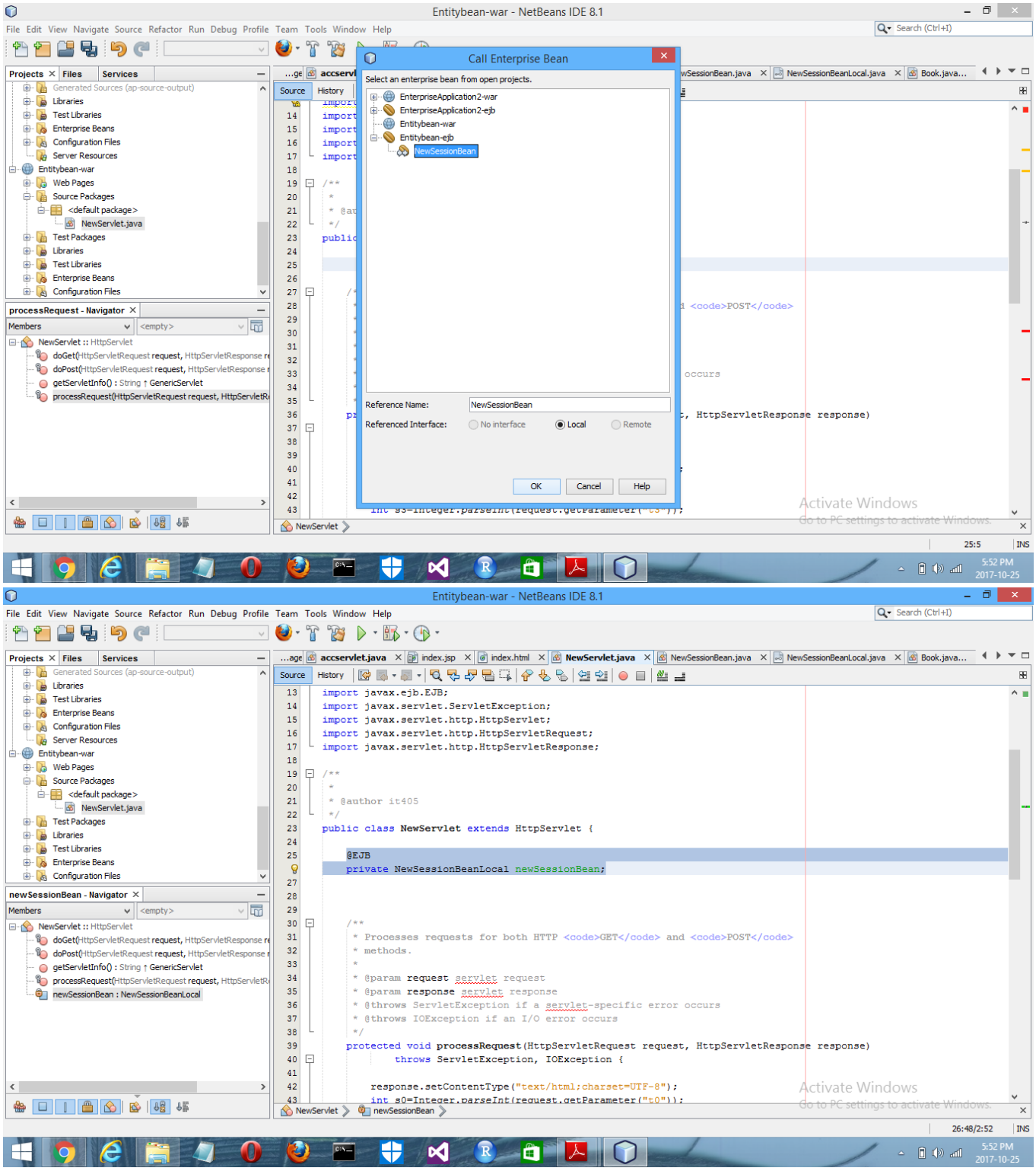
```
package ejb;
import bec.Book;
import java.util.Collection;
```

```
import javax.ejb.Stateless;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
@Stateless
public class NewSessionBean implements NewSessionBeanLocal
{
@PersistenceContext
EntityManagerem;
protected Book book;
protected Collection<Book> booklist;
@Override
public void addBook(intid,Stringtitle,Stringauthor,int price)
{
if(book==null)
book=new Book(id,title,author,price);
em.persist(book);
}
@Override
public Collection<Book>getAllBooks()
{
booklist=em.createQuery("Select b from Bookbank b").getResultList();
return booklist;
}
}
```

Right click on war module → New → Servlet


Entitybean-war - NetBeans IDE 8.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects Files Services
Entitybean-war
Web Pages
Source Packages
<default package>
NewServlet.java
Test Packages
Libraries
Test Libraries
Enterprise Beans
Configuration Files
Server Resources
processRequest - Navigator
Members
NewServlet :: HttpServlet
doGet(HttpServletRequest request, HttpServletResponse r
doPost(HttpServletRequest request, HttpServletResponse r
getServletInfo() : String ↑ GenericServlet
processRequest(HttpServletRequest request, HttpServletR
accservlet.java
index.jsp
index.html
NewServlet.java
NewSessionBean.java
NewSessionBeanLocal.java
Book.java...
Source
History
Navigate
Show Javadoc Alt+F1
Find Usages Alt+F7
Call Hierarchy
Insert Code... Alt+Insert
Fix Imports Ctrl+Shift+I
Refactor
Format Alt+Shift+F
Run File Shift+F6
Debug File Ctrl+Shift+F5
Test File Ctrl+F6
Debug Test File Ctrl+Shift+F6
Run Focused Test Method
Debug Focused Test Method
Run Into Method
New Watch... Ctrl+Shift+F7
Toggle Line Breakpoint Ctrl+F8
Profile
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Code Folds
Select in Projects
Activate Windows
Go to PC settings to activate Windows.
25:5
INS
5:52 PM
2017-10-25


Entitybean-war - NetBeans IDE 8.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Projects Files Services
Entitybean-war
Web Pages
Source Packages
<default package>
NewServlet.java
Test Packages
Libraries
Test Libraries
Enterprise Beans
Configuration Files
Server Resources
processRequest - Navigator
Members
NewServlet :: HttpServlet
doGet(HttpServletRequest request, HttpServletResponse r
doPost(HttpServletRequest request, HttpServletResponse r
getServletInfo() : String ↑ GenericServlet
processRequest(HttpServletRequest request, HttpServletR
accservlet.java
index.jsp
index.html
NewServlet.java
NewSessionBean.java
NewSessionBeanLocal.java
Book.java...
Source
History
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
/**
*
* @author it405
*/
public class NewServlet extends HttpServlet {
Generate
Constructor...
Logger...
toString()...
Override Method...
Add Property...
Call Enterprise Bean...
Use Database...
Send JMS Message...
Send E-mail...
Call Web Service Operation...
Generate REST Client...
response.setContentType("text/html;charset=UTF-8");
int s0=Integer.parseInt(request.getParameter("t0"));
String s1=request.getParameter("t1");
String s2=request.getParameter("t2");
int s3=Integer.parseInt(request.getParameter("t3"));
Activate Windows
Go to PC settings to activate Windows.
25:5
INS
5:52 PM
2017-10-25



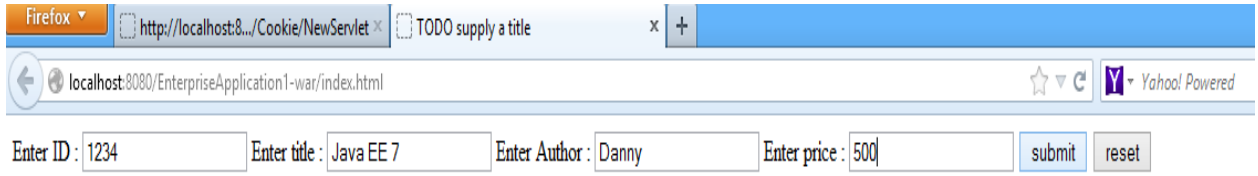
NewServlet.java:

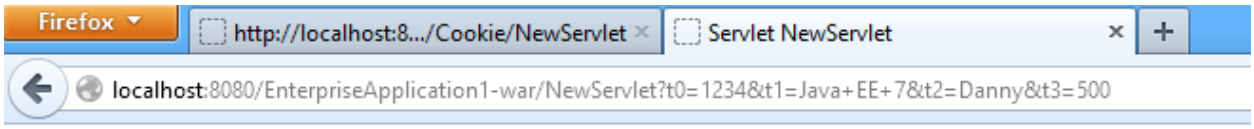
```
import bec.Book;

import ejb.NewSessionBeanLocal;

import java.io.IOException;

import java.io.PrintWriter;
```

```
import java.util.Collection;
import java.util.Iterator;
import javax.ejb.EJB;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class NewServlet extends HttpServlet
{
@EJB
private NewSessionBeanLocalnewSessionBean;
protected void processRequest(HttpServletRequest request, HttpServletResponse
response)throws ServletException, IOException
{
response.setContentType("text/html;charset=UTF-8");
int s0=Integer.parseInt(request.getParameter("t0"));
String s1=request.getParameter("t1");
String s2=request.getParameter("t2");
int s3=Integer.parseInt(request.getParameter("t3"));
try (PrintWriter out = response.getWriter())
{
if(s1!=null&&s2!=null&&s3!=0)
{
newSessionBean.addBook(s0,s1,s2,s3);
}
out.println("<h1> record added</h1>");
Collection list=(Collection)newSessionBean.getAllBooks();
```

```
for(Iterator iter=list.iterator();iter.hasNext();)
{
Book element=(Book)iter.next();
out.println("<p>Book id:<b>"+element.getId()+"</b></p>");
out.println("<p>Book title:<b>"+element.getTitle()+"</b></p>");
out.println("<p>Book author:<b>"+element.getAuthor()+"</b></p>");
out.println("<p>Book price:<b>"+element.getPrice()+"</b></p>");
}
}
}
```

Output:





record added

Right click on the database to check whether the row added ,