# Open Ended Problem

## **AIM:** Using J2EE JSP/Servlet API develop student’s management system required to manage student’s academic activity such as student’s profile, student’s day to day assignment submission as per instructions and assignment given by teacher . Provide MVC based interface using spring framework and do the database design using Hibernet framework and also provide two login roles

## one for teachers providing assignment and notification for class and other for students to submit their assignments and can view notices published by teachers

userlogin.java

<%-- Document : userlogin Created on : 9 May, 2017, 6:56:41 PM--%>

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

<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form"%>

<!DOCTYPE html>

<html> <head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title> </head>

<body> <h1>Login to System</h1>

<form:form method="POST" commandName="useroep">

<table style="align-content: center;margin-left: 200px">

<tr> <td>User Name :</td> <td><form:input path="uname" /></td> </tr>

<tr> <td>Password :</td> <td><form:password path="password" /></td></tr>

<tr> <td>Faculty :</td><td><form:radiobutton path="faculty" value="true" label="Faculty" />

<form:radiobutton path="faculty" value="false" label="Student" /></td> </tr>

<tr> <td colspan="2"><input type="submit" value="Login" /></td> </tr>

</table>

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

**student.jsp**

<%-- Document : student Created on : 10 May, 2017, 12:39:02 AM--%>

<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

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

<%@taglib uri="http://www.springframework.org/tags/form" prefix="form"%>

<!DOCTYPE html>

<html> <head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title> </head> <body>

<h1>Hello Student your Name is ${uname}</h1>

<form action="stsubmission" method="post">

<table>

<tr> <td>Submission ID</td> <td><input type="text" name="subid" /></td> </tr>

<tr> <td>Assignment ID</td> <td><input type="text" name="asid" /></td> </tr>

<tr><td>User Name</td><td><input type="text" name="uname" value="${uname}" /></td></tr>

<tr><td>Assignment Solution Answer</td>

<td><textarea name="assignment" cols="25" rows="10" ></textarea></td> </tr>

<tr><td colspan="2"><input type="submit" value="Post Assignment Statement" /></td></tr>

</table>

</form> <br/><br/><br/> <h1>Assignment Questions By Faculty</h1>

<table cellpadding="10">

<tr> <th>Assignment ID</th> <th>Username</th> <th>Assignment Statement</th></tr>

<c:forEach items="${assignments}" var="sub">

<tr> <td><c:out value="${sub.asid}"/></td> <td><c:out value="${sub.uname}"/></td>

<td><c:out value="${sub.astmt}"/></td> </tr>

</c:forEach>

</table>

</body></html>

**Faculty.jsp**

<%-- Document : faculty Created on : 10 May, 2017, 12:33:45 AM--%>

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

<%@taglib uri="http://www.springframework.org/tags/form" prefix="form"%>

<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<!DOCTYPE html>

<html> <head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head> <body>

<h1>Hello Faculty ${uname}</h1>

<form action="facultyquestion" method="post">

<table>

<tr> <td>Assignment ID</td> <td><input type="text" name="asid" /></td> </tr>

<tr><td>User Name</td><td><input type="text" name="uname" value="${uname}" /></td></tr>

<tr> <td>Assignment Question</td> <td><input type="text" name="astmt" /></td></tr>

<tr> <td colspan="2"><input type="submit" value="Post Assignment Statement" /></td></tr>

</table>

</form> <br/><br/><br/><br/>

<h1>Submission Submitted by Student with Their Name and submission ID</h1>

<table cellpadding="10">

<tr><th>Submission ID</th><th>Assignment ID</th><th>Username</th><th>Assignment Submitted Contents</th> </tr>

<c:forEach items="${submissions}" var="sub">

<tr><td><c:out value="${sub.subid}"/></td><td><c:out value="${sub.asid}"/></td> <td><c:out value="${sub.uname}"/></td><td><c:out value="${sub.assignment}"/></td></tr>

</c:forEach>

</table>

</body></html>

**errorlogin.jsp**

<%-- Document : errorlogin Created on : 10 May, 2017, 11:38:41 AM--%>

<%@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>Sorry !!! Please insert correct username and password go Back</h1>

</body></html>

**dispatcher-servlet.xml**

<?xml version='1.0' encoding='UTF-8' ?>

<!-- was: <?xml version="1.0" encoding="UTF-8"?> -->

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:p="http://www.springframework.org/schema/p"

xmlns:aop="http://www.springframework.org/schema/aop"

xmlns:tx="http://www.springframework.org/schema/tx"

xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.0.xsd

http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-4.0.xsd

http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-4.0.xsd">

<bean class="org.springframework.web.servlet.mvc.support.ControllerClassNameHandlerMapping"/>

<bean class="service.UserServiceImpl" id="userService"/>

<bean class="controller.UserController" p:successView="success" p:formView="userlogin" p:userService-ref="userService" name="/userlogin.html"/>

<bean class="service.FacultyService" id="facultyService"/>

<bean class="controller.FacultyController" p:commandName="aspost" p:successView="apostsuccess" p:formView="faculty" p:facultyService-ref="facultyService" name="/faculty.html"/>

<bean p:suffix=".jsp" p:prefix="/WEB-INF/jsp/" class="org.springframework.web.servlet.view.InternalResourceViewResolver" id="viewResolver"/>

</beans>

**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">

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>/WEB-INF/applicationContext.xml</param-value>

</context-param>

<listener>

<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

</listener>

<servlet>

<servlet-name>dispatcher</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet>

<servlet-name>facultyquestion</servlet-name>

<servlet-class>facultyquestion</servlet-class>

</servlet>

<servlet>

<servlet-name>stsubmission</servlet-name>

<servlet-class>stsubmission</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>dispatcher</servlet-name>

<url-pattern>\*.html</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>facultyquestion</servlet-name>

<url-pattern>/facultyquestion</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>stsubmission</servlet-name>

<url-pattern>/stsubmission</url-pattern>

</servlet-mapping>

<session-config>

<session-timeout> 30 </session-timeout>

</session-config>

<welcome-file-list>

<welcome-file>userlogin.html</welcome-file>

<welcome-file>faculty.html</welcome-file>

</welcome-file-list>

</web-app>

**hibernate.cfg.xml**

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

<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN" "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

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

<property name="hibernate.connection.driver\_class">org.apache.derby.jdbc.ClientDriver</property>

<property name="hibernate.connection.url">jdbc:derby://localhost:1527/MRV</property>

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

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

<mapping resource="domain/Useroep.hbm.xml"/>

<mapping resource="domain/Asubmitted.hbm.xml"/>

<mapping resource="domain/Aspost.hbm.xml"/>

</session-factory>

</hibernate-configuration>

**UserController.java**

package controller;

import domain.HibernateUtil;

import org.springframework.web.servlet.mvc.SimpleFormController;

import service.UserService;

import domain.Useroep;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.boot.registry.StandardServiceRegistryBuilder;

import org.hibernate.cfg.Configuration;

import org.hibernate.service.ServiceRegistry;

import org.springframework.validation.BindException;

import org.springframework.web.servlet.ModelAndView;

import service.FacultyService;

import service.StudentService;

public class UserController extends SimpleFormController {

private UserService userService;

public UserController() {

setCommandClass(Useroep.class);

setCommandName("useroep");

}

public void setUserService(UserService userService) { this.userService = userService; }

@Override

protected ModelAndView onSubmit(

HttpServletRequest request,

HttpServletResponse response,

Object command,

BindException errors) throws Exception {

Useroep user = (Useroep) command;

boolean status=false;

ModelAndView mv = null;

if(user.getFaculty()==true && !user.getUname().isEmpty() && !user.getPassword().isEmpty()){

status = userService.loginfaculty(user);

if(status == true){

mv = new ModelAndView("faculty");

mv.addObject("uname",user.getUname());

FacultyService fs = new FacultyService();

mv.addObject("submissions", fs.getsubmissions());

} else{ mv = new ModelAndView("errorlogin"); }

}

else if(user.getFaculty()==false && !user.getUname().isEmpty() && !user.getPassword().isEmpty()){

status = userService.loginstudent(user);

if(status == true){

mv = new ModelAndView("student");

mv.addObject("uname",user.getUname());

StudentService ss = new StudentService();

mv.addObject("assignments", ss.getassignments());

} else{ mv = new ModelAndView("errorlogin"); }

} else{mv = new ModelAndView("errorlogin");}

System.out.println(mv.toString()+status);

return mv;

} }

**Facultyquestion.java**

import domain.Aspost;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import service.FacultyService;

public class facultyquestion extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

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

try (PrintWriter out = response.getWriter()) {

/\* TODO output your page here. You may use following sample code. \*/

Aspost aspost = new Aspost();

aspost.setAsid(request.getParameter("asid"));

aspost.setUname(request.getParameter("uname"));

aspost.setAstmt(request.getParameter("astmt"));

boolean insertstatus = false;

FacultyService fservice = new FacultyService();

insertstatus = fservice.postasmnt(aspost);

out.println("<!DOCTYPE html>");

out.println("<html>");

out.println("<head>");

out.println("<title>Servlet facultyquestion</title>");

out.println("</head>");

out.println("<body>");

out.println("<h1>Servlet "+request.getParameter("uname")+"facultyquestion at " + request.getContextPath() + "</h1>");

if(insertstatus==true)

out.println("<h1>Data Inserted Successfully</h1>");

else

out.println("<h1>Data not Inserted Successfully</h1>");

out.println("</body>");

out.println("</html>");

}

}

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

public String getServletInfo() {

return "Short description";

}// </editor-fold>

}

**Stsubmission.java**

import domain.Asubmitted;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import service.StudentService;

public class stsubmission extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

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

try (PrintWriter out = response.getWriter()) {

Asubmitted as = new Asubmitted();

as.setSubid(request.getParameter("subid"));

as.setAsid(request.getParameter("asid"));

as.setUname(request.getParameter("uname"));

as.setAssignment(request.getParameter("assignment"));

boolean insertstatus = false;

StudentService ss = new StudentService();

insertstatus = ss.submitasmt(as);

out.println("<!DOCTYPE html>");

out.println("<html>");

out.println("<head>");

out.println("<title>Servlet stsubmission</title>");

out.println("</head>");

out.println("<body>");

out.println("<h1>Servlet stsubmission at " + request.getContextPath() + "</h1>");

if(insertstatus==true)

out.println("<h1>Data Inserted Successfully</h1>");

else

out.println("<h1>Data not Inserted Successfully</h1>");

out.println("</body>");

out.println("</html>");

}

}

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

public String getServletInfo() {

return "Short description";

}// </editor-fold>

}

**domain/Aspost.hbm.xml**

<?xml version="1.0"?>

<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<!-- Generated 10 May, 2017 2:09:43 PM by Hibernate Tools 4.3.1 -->

<hibernate-mapping>

<class name="domain.Aspost" table="ASPOST" schema="APP">

<id name="asid" type="string">

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

<generator class="assigned" />

</id> <property name="uname" type="string"> <column name="UNAME" length="20" />

</property>

<property name="astmt" type="string"> <column name="ASTMT" length="32700" /> </property>

</class>

</hibernate-mapping>

**Asubmitted.hbm.xml**

<?xml version="1.0"?>

<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<!-- Generated 10 May, 2017 2:09:43 PM by Hibernate Tools 4.3.1 -->

<hibernate-mapping>

<class name="domain.Asubmitted" table="ASUBMITTED" schema="APP">

<id name="subid" type="string"> <column name="SUBID" length="20" />

<generator class="assigned" />

</id> <property name="asid" type="string"> <column name="ASID" length="20" />

</property>

<property name="uname" type="string"> <column name="UNAME" length="20" />

</property>

<property name="assignment" type="string"><column name="ASSIGNMENT" length="32700" />

</property>

</class>

</hibernate-mapping>

**Useroep.hbm.xml**

<?xml version="1.0"?>

<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">

<!-- Generated 10 May, 2017 2:09:43 PM by Hibernate Tools 4.3.1 -->

<hibernate-mapping>

<class name="domain.Useroep" table="USEROEP" schema="APP">

<id name="uname" type="string"> <column name="UNAME" length="20" />

<generator class="assigned"></generator>

</id> <property name="password" type="string"> <column name="PASSWORD" length="20" />

</property><property name="faculty" type="java.lang.Boolean"> <column name="FACULTY" />

</property>

</class>

</hibernate-mapping>

**Aspost.java**

package domain;

public class Aspost implements java.io.Serializable {

private String asid;

private String uname;

private String astmt;

public Aspost() { }

public Aspost(String asid) { this.asid = asid; }

public Aspost(String asid, String uname, String astmt) { this.asid = asid;

this.uname = uname;

this.astmt = astmt;

}

public String getAsid() { return this.asid; }

public void setAsid(String asid) { this.asid = asid; }

public String getUname() { return this.uname; }

public void setUname(String uname) { this.uname = uname; }

public String getAstmt() { return this.astmt; }

public void setAstmt(String astmt) { this.astmt = astmt; }

}

**Asubmitted.java**

package domain;

public class Asubmitted implements java.io.Serializable {

private String subid;

private String asid;

private String uname;

private String assignment;

public Asubmitted() { }

public Asubmitted(String subid) { this.subid = subid; }

public Asubmitted(String subid, String asid, String uname, String assignment) {

this.subid = subid;

this.asid = asid;

this.uname = uname;

this.assignment = assignment;

}

public String getSubid() { return this.subid; }

public void setSubid(String subid) { this.subid = subid; }

public String getAsid() { return this.asid; }

public void setAsid(String asid) { this.asid = asid; }

public String getUname() { return this.uname; }

public void setUname(String uname) { this.uname = uname; }

public String getAssignment() { return this.assignment; }

public void setAssignment(String assignment) { this.assignment = assignment; }

}

**Useroep.java**

package domain;

public class Useroep implements java.io.Serializable {

private String uname;

private String password;

private Boolean faculty;

public Useroep() { }

public Useroep(String uname) { this.uname = uname; }

public Useroep(String uname, String password, Boolean faculty) { this.uname = uname;

this.password = password;

this.faculty = faculty;

}

public String getUname() { return this.uname; }

public void setUname(String uname) { this.uname = uname; }

public String getPassword() { return this.password; }

public void setPassword(String password) { this.password = password; }

public Boolean getFaculty() { return this.faculty; }

public void setFaculty(Boolean faculty) { this.faculty = faculty; }

}

**UserService.java**

package service;

import domain.Useroep;

import java.util.List;

public interface UserService {

public boolean loginstudent(Useroep useroep);

public boolean loginfaculty(Useroep useroep);

public List getsubmissions();

}

**UserServiceImpl.java**

package service;

import domain.Asubmitted;

import domain.HibernateUtil;

import domain.Useroep;

import java.util.Iterator;

import java.util.List;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import org.hibernate.boot.registry.StandardServiceRegistryBuilder;

import org.hibernate.cfg.Configuration;

import org.hibernate.service.ServiceRegistry;

public class UserServiceImpl implements UserService {

@Override

public boolean loginstudent(Useroep user) {

boolean status = false;

Session session = null;

try{

session = HibernateUtil.getSessionFactory().openSession();

session.beginTransaction();

List result;

result = session.createQuery("from Useroep where uname='"+user.getUname()+"' and faculty='false'").list();

Iterator i = result.iterator();

Useroep user1 = (Useroep)i.next();

if(user1.getPassword().equals(user.getPassword())){ status = true; }

else{ status=false; }

session.getTransaction().commit();

} catch(Exception ex) { System.err.println("Exception::"+ex); }

finally{ System.out.println("Succesfull 123456789 "+status); }

System.out.println("Status "+status);

return status;

}

@Override

public boolean loginfaculty(Useroep useroep) {

boolean status = false;

Session session = null;

try{

session = HibernateUtil.getSessionFactory().openSession();

session.beginTransaction();

List result;

result = session.createQuery("from Useroep where uname='"+useroep.getUname()+"'and faculty='true'").list();

Iterator i = result.iterator();

Useroep user1 = (Useroep)i.next();

if(user1.getPassword().equals(useroep.getPassword())) {

System.out.println("----Password"+user1.getPassword());

System.out.println("----Uname"+user1.getUname());

System.out.println("------Entered Password"+useroep.getPassword());

status = true;

} else { status=false; }

session.getTransaction().commit();

} catch(Exception ex) { System.err.println("Exception::"+ex); }

finally{ System.out.println("Succesfull 123456789 "+status); }

System.out.println("Status "+status);

return status;

}

@Override

public List getsubmissions() {

List result = null;

Asubmitted asub = new Asubmitted();

try{ Configuration configuration = new Configuration().configure();

ServiceRegistry serviceRegistry;

serviceRegistry = new StandardServiceRegistryBuilder()

.applySettings(configuration.getProperties()).build();

SessionFactory sessionFactory = configuration.buildSessionFactory(serviceRegistry);

Session session = sessionFactory.openSession();

session.beginTransaction();

result = session.createQuery("from domain.Asubmitted").list();

session.getTransaction().commit();

}catch(Exception e) { e.printStackTrace(); }

return result;

}}

**FacultyService.java**

package service;

import domain.Aspost;

import domain.Asubmitted;

import java.util.List;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.boot.registry.StandardServiceRegistryBuilder;

import org.hibernate.cfg.Configuration;

import org.hibernate.service.ServiceRegistry;

public class FacultyService {

public boolean postasmnt(Aspost aspost) {

boolean isatatus = false;

try{ Configuration configuration = new Configuration().configure();

ServiceRegistry serviceRegistry;

serviceRegistry = new StandardServiceRegistryBuilder()

.applySettings(configuration.getProperties()).build();

SessionFactory sessionFactory = configuration.buildSessionFactory(serviceRegistry);

Session session = sessionFactory.openSession();

session.beginTransaction();

session.save(aspost);

session.getTransaction().commit();

isatatus = true;

}catch(Exception e) { e.printStackTrace(); }

return isatatus;

}

public List getsubmissions() {

List result = null;

Asubmitted asub = new Asubmitted();

try{ Configuration configuration = new Configuration().configure();

ServiceRegistry serviceRegistry;

serviceRegistry = new StandardServiceRegistryBuilder()

.applySettings(configuration.getProperties()).build();

SessionFactory sessionFactory = configuration.buildSessionFactory(serviceRegistry);

Session session = sessionFactory.openSession();

session.beginTransaction();

result = session.createQuery("from domain.Asubmitted").list();

session.getTransaction().commit();

}catch(Exception e) { e.printStackTrace(); }

return result;

}}

**StudentService.java**

package service;

import domain.Asubmitted;

import java.util.List;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.boot.registry.StandardServiceRegistryBuilder;

import org.hibernate.cfg.Configuration;

import org.hibernate.service.ServiceRegistry;

public class StudentService {

public boolean submitasmt(Asubmitted as) {

boolean isatatus = false;

try{ Configuration configuration = new Configuration().configure();

ServiceRegistry serviceRegistry;

serviceRegistry = new StandardServiceRegistryBuilder()

.applySettings(configuration.getProperties()).build();

SessionFactory sessionFactory = configuration.buildSessionFactory(serviceRegistry);

Session session = sessionFactory.openSession();

session.beginTransaction();

session.save(as);

session.getTransaction().commit();

isatatus = true;

}catch(Exception e) { e.printStackTrace(); }

return isatatus;

}

public List getassignments() {

List result = null;

try{ Configuration configuration = new Configuration().configure();

ServiceRegistry serviceRegistry;

serviceRegistry = new StandardServiceRegistryBuilder()

.applySettings(configuration.getProperties()).build();

SessionFactory sessionFactory = configuration.buildSessionFactory(serviceRegistry);

Session session = sessionFactory.openSession();

session.beginTransaction();

result = session.createQuery("from domain.Aspost").list();

session.getTransaction().commit();

}catch(Exception e) { e.printStackTrace(); }

return result;

} }

**Output**





