1.Design and create a swing based simple calculator program to perform basic arithmetic operations

package programs;

import javax.swing.\*;

import java.awt.event.\*;

class Calculator implements ActionListener

{

JFrame f;

JTextField t;

String arr[]= { "0","1","2","3","4","5","6","7","8","9","/","\*","-","+",".","=","delete","Clear"};

JButton [] button=new JButton[18];

static double a=0,b=0,result=0;

static int operator=0;

Calculator()

{

for(int i=0;i<18;i++)

{

button[i]=new JButton(arr[i]);

}

f=new JFrame("Calculator");

t=new JTextField();

t.setBounds(30,40,280,30);

int x=30;

int k=0;

for(int i=0;i<4;i++) {

x+=70;int y=40;

for(int j=0;j<4;j++) {

button[k].setBounds(y,x,50,40);

k++;y+=70;

}

}

button[16].setBounds(60,380,100,40);

button[17].setBounds(180,380,100,40);

f.add(t);

for(int i=0;i<18;i++) {

f.add(button[i]);

}

f.setLayout(null);

f.setVisible(true);

f.setSize(350,500);

f.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

f.setResizable(false);

for(int i=0;i<18;i++) {

button[i].addActionListener(this);

}

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==button[1])

t.setText(t.getText().concat("1"));

if(e.getSource()==button[2])

t.setText(t.getText().concat("2"));

if(e.getSource()==button[3])

t.setText(t.getText().concat("3"));

if(e.getSource()==button[4])

t.setText(t.getText().concat("4"));

if(e.getSource()==button[5])

t.setText(t.getText().concat("5"));

if(e.getSource()==button[6])

t.setText(t.getText().concat("6"));

if(e.getSource()==button[7])

t.setText(t.getText().concat("7"));

if(e.getSource()==button[8])

t.setText(t.getText().concat("8"));

if(e.getSource()==button[9])

t.setText(t.getText().concat("9"));

if(e.getSource()==button[0])

t.setText(t.getText().concat("0"));

if(e.getSource()==button[14])

t.setText(t.getText().concat("."));

if(e.getSource()==button[13])

{

a=Double.parseDouble(t.getText());

operator=1;

t.setText("");

}

if(e.getSource()==button[12])

{

a=Double.parseDouble(t.getText());

operator=2;

t.setText("");

}

if(e.getSource()==button[11])

{

a=Double.parseDouble(t.getText());

operator=3;

t.setText("");

}

if(e.getSource()==button[10])

{

a=Double.parseDouble(t.getText());

operator=4;

t.setText("");

}

if(e.getSource()==button[15])

{

b=Double.parseDouble(t.getText());

switch(operator)

{

case 1: result=a+b;

break;

case 2: result=a-b;

break;

case 3: result=a\*b;

break;

case 4: result=a/b;

break;

default: result=0;

}

t.setText(""+result);

}

if(e.getSource()==button[16])

t.setText("");

if(e.getSource()==button[17])

{

String s=t.getText();

t.setText("");

for(int i=0;i<s.length()-1;i++)

t.setText(t.getText()+s.charAt(i));

}

}

public static void main(String args[])

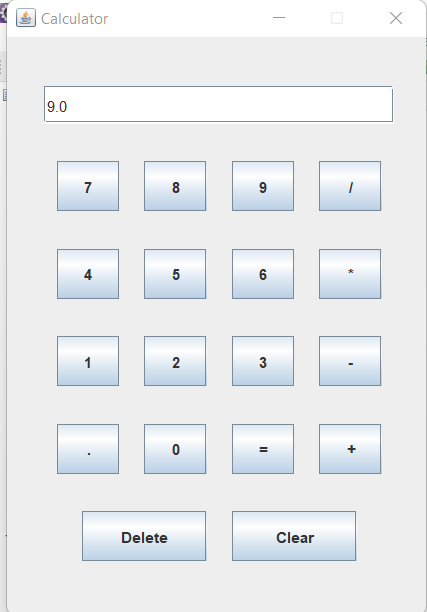
{

new Calculator();

}

}

OUTPUT



1. Design and demonstrate loading of file in a Swing Component.

package com.nhce.advancedjava;

import javax.swing.\*;

import java.awt.event.\*;

import java.io.\*;

public class FileChooserProgram2 extends JFrame implements ActionListener {

JMenuBar mb;

JMenu menu;

JMenuItem menuitem;

JTextArea ta;

FileChooserProgram2() {

mb = new JMenuBar();

mb.setBounds(0, 0, 800, 20);

menu = new JMenu("File");

menuitem = new JMenuItem("Open File");

menu.add(menuitem);

menuitem.addActionListener(this);

mb.add(menu);

ta = new JTextArea(800, 800);

ta.setBounds(0, 20, 800, 800);

add(mb);

add(ta);

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == menuitem) {

JFileChooser fc = new JFileChooser();

int i = fc.showOpenDialog(this);

if (i == JFileChooser.APPROVE\_OPTION) {

File f = fc.getSelectedFile();

String filepath = f.getPath();

try {

BufferedReader br = new BufferedReader(new FileReader(

filepath));

String s1 = "", s2 = "";

while ((s1 = br.readLine()) != null) {

s2 += s1 + "\n";

}

ta.setText(s2);

br.close();

} catch (Exception ex) {

ex.printStackTrace();

}

}

}

}

public static void main(String[] args) {

FileChooserProgram2 om = new FileChooserProgram2();

om.setSize(500, 500);

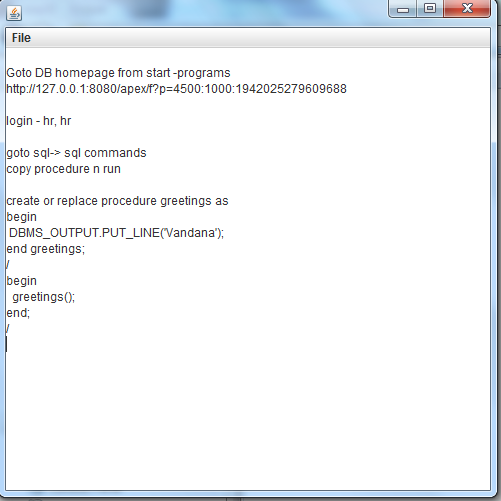
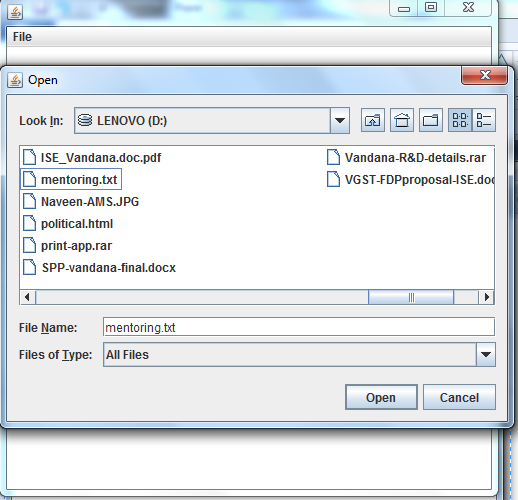
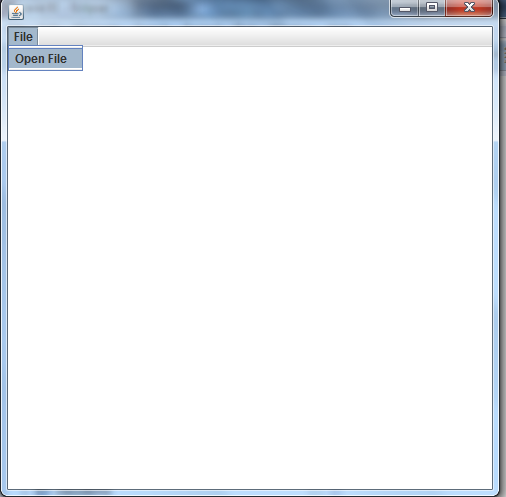
om.setLayout(null);

om.setVisible(true);

om.setDefaultCloseOperation(EXIT\_ON\_CLOSE);

}

}



1. Design and develop a swing based application to count the number of times a specific button is clicked by the user, apply event handling mechanism

package com.nhce.advancedjava;

import java.awt.\*;

import java.awt.event.\*;

public class ButtonCount extends Frame implements WindowListener,ActionListener {

private static final long *serialVersionUID* = 1L;

TextField text = new TextField(20);

Button b;

private int numClicks = 0;

public static void main(String[] args) {

ButtonCount myWindow = new ButtonCount("Button Click Count Window");

myWindow.setSize(350,100);

myWindow.setVisible(true);

}

public ButtonCount(String title) {

super(title);

setLayout(new FlowLayout());

addWindowListener(this);

b = new Button("Click me");

add(b);

add(text);

b.addActionListener(this);

}

public void actionPerformed(ActionEvent e) {

numClicks++;

text.setText("Button Clicked " + numClicks + " times");

}

public void windowClosing(WindowEvent e) {

dispose();

System.*exit*(0);

}

public void windowOpened(WindowEvent e) {}

public void windowActivated(WindowEvent e) {}

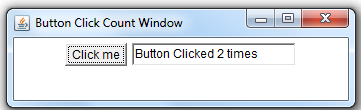
public void windowIconified(WindowEvent e) {}

public void windowDeiconified(WindowEvent e) {}

public void windowDeactivated(WindowEvent e) {}

public void windowClosed(WindowEvent e) {}

}



1. Design, Develop and Implement a JDBC program using statement object to display the Employee information to the console. Assume suitable columns and rows for the Employee table and JDBC drivers.

/\*\*

\* 1. JDK 1.7

\* 2. Eclipse Juno

\* 3. Oracle 10g express edition 4. Pure java

\* thin driver-type 4 – ojdbc14.jar

\* 3. Incase of Mysql DB, use mysql-connector.jar

\* 4. Put this jar in Eclipse build path

\* 5. Incase of mysql, use following driver name and URL for program 4,5,6

**Class.forName("com.mysql.cj.jdbc.Driver");**

**Connection con=DriverManager.getConnection(**

**"jdbc:mysql://localhost:3306/student\_tbl","root","root");**

SQL> connect

Enter user-name: hr

Enter password:

Connected.

SQL> create table employee\_tbl (id number(3), name varchar2(30), department varc

har2(20), salary number(5));

Table created.

SQL> desc employee\_tbl;

Name Null? Type

----------------------------------------- -------- ----------------------------

ID NUMBER(3)

NAME VARCHAR2(30)

DEPARTMENT VARCHAR2(20)

SALARY NUMBER(5)

SQL> insert into employee\_tbl values(123,'Hari Kumar', 'HR', 20000);

1 row created.

SQL> insert into employee\_tbl values(113,'Amit Shah', 'Sales', 20001);

1 row created.

SQL> insert into employee\_tbl values(002,'Sita Aggarwal', 'Admin', 60001);

1 row created.

SQL> commit;

Commit complete.

import java.sql.\*;

class LabProgramEmployeeDisplay4 {

public static void main(String args[]) {

int id = 0, salary = 0;

String name, dept;

try {

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con = DriverManager.getConnection(

"jdbc:oracle:thin:@localhost:1521:xe", "hr", "hr");

Statement stmt = con.createStatement();

ResultSet rs = stmt.executeQuery("select \* from employee\_tbl");

System.out.println("Employee Details");

System.out.println("EmployeeID " + "\t" + "Name "

+ "\t" + "Department " + "\t"

+ "Salary " );

while (rs.next()) {

id = rs.getInt(1);

name = rs.getString(2);

dept = rs.getString(3);

salary = rs.getInt(4);

System.out.println( id + "\t" + name + "\t" + dept + "\t"

+ salary + "\t" );

}

con.close();

} catch (Exception e) {

System.out.println(e);

}

}

}

1. Design, Develop and Implement a JDBC based program using Prepared statement object to perform update operation and display the updated Product data. Assume suitable columns for the Product table and JDBC drivers.

SQL> desc product\_tbl;

Name Null? Type

----------------------------------------- -------- -------------

ID NUMBER(3)

NAME VARCHAR2(30)

PRICE NUMBER(5)

import java.sql.\*;

import java.util.Scanner;

class LabProgramProduct5 {

public static void main(String args[]) {

int id = 0, new\_price = 0, price=0;

String name;

try {

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con = DriverManager.getConnection(

"jdbc:oracle:thin:@localhost:1521:xe", "hr", "hr");

System.out.println("Enter the Product ID and the new Price");

Scanner ibn = new Scanner(System.in);

id=ibn.nextInt();

new\_price=ibn.nextInt();

String updatequery = "update Product\_tbl Set Price="+new\_price+" where id="+ id;

PreparedStatement pst = con.prepareStatement(updatequery);

int result=pst.executeUpdate();

if (result!=0)

{

System.out.println("Product table updated");

}

Statement stmt = con.createStatement();

ResultSet rs = stmt.executeQuery("select \* from Product\_tbl");

System.out.println("Product Details");

System.out.println("ID " + "\t" + "Name " + "\t"

+ "Price " );

while (rs.next()) {

id = rs.getInt(1);

name = rs.getString(2);

price = rs.getInt(3);

System.out.println( id + "\t" + name + "\t" + price + "\t" );

}

con.close();

} catch (Exception e) {

System.out.println(e);

}

}

}

1. Design, Develop and Implement a JDBC based program using callable statement

to execute a stored SQL procedure to display the USN of all students from

student table. Assume suitable columns and rows for the Student table and JDBC

drivers.

/\* type store procedure on Mysql command promt

DELIMITER $$

USE 'mysql\_database'$$

CREATE PROCEDURE selection()

BEGIN

Select \* from student\_tbl;

END$$

DELIMITER ;

\*/



package com.nhce.advancedjava;

import java.sql.\*;

public class Labprogram6sql {

public static void main(String[] args) throws Exception{

Class.forName("com.mysql.cj.jdbc.Driver ");

Connection con=DriverManager.getConnection(

"jdbc:mysql://localhost:3306/database","hr","hr");

CallableStatement cs = con.prepareCall("{call selection()}");

ResultSet rs = cs.executeQuery();

System.out.println("Contents of the result-set");

while( rs.next())

{

int st\_usn=rs.getInt(1);

System.out.println(st\_usn);

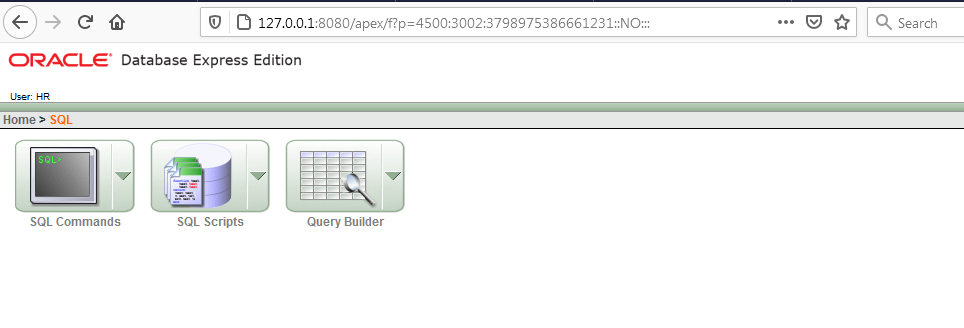
}

}

}

**//FOR ORACLE DB – Goto ORACLE HOME PAGE from All program->Oracle**

**Login with username and password**



**CLICK on SQL commands option**

/\*

\*

create or replace procedure selection(data OUT SYS\_REFCURSOR)

as

begin

OPEN data FOR

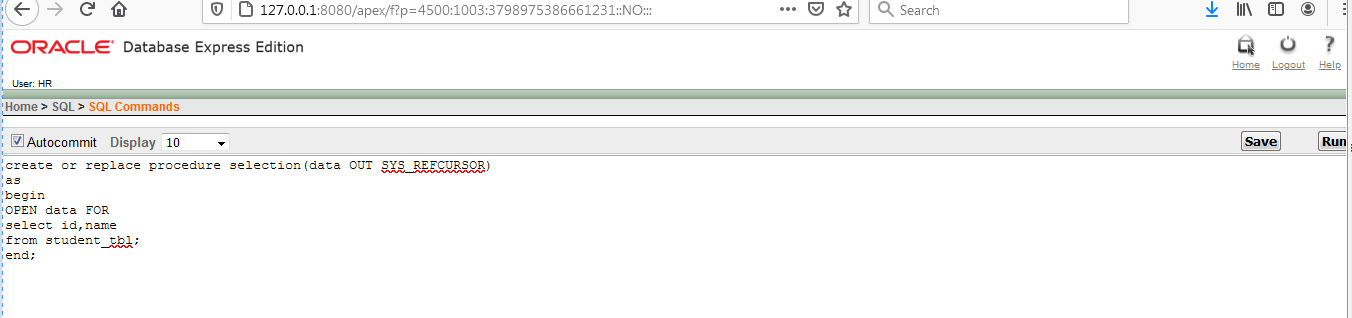
select id,name

from student\_tbl;

end;

\*

\*/



package com.nhce.advancedjava;

import java.sql.\*;

import oracle.jdbc.driver.OracleTypes;

public class Labprogram6 {

public static void main(String[] args) {

try {

Class.forName("oracle.jdbc.driver.OracleDriver");

Connection con = DriverManager.getConnection(

"jdbc:oracle:thin:@localhost:1521:xe", "hr", "hr");

CallableStatement cs = con.prepareCall("{call selection(?)}");

cs.registerOutParameter(1, OracleTypes.CURSOR);

cs.execute();

ResultSet rs = (ResultSet) cs.getObject(1);

if (rs != null) {

System.out.println("Student Details");

System.out.println("Student USN \t\t Student Name");

while (rs.next()) {

int st\_id = rs.getInt(1);

String name = rs.getString(2);

System.out.println(st\_id + "\t\t\t" + name);

}

}

} catch (Exception e) {

e.printStackTrace();

}

}

}

Student Details

Student USN Student Name

144 geeta

1011 Amit

1012 Amitra

1089 Amitra

122 Anu

122 Anu

1 john

12 hari

13 hareesh

Design, Develop and Implement a JDBC based program using callable statement to execute a stored SQL procedure to display the details of a student matching the USN provided, from student table. Assume suitable columns and rows for the Student table and JDBC drivers.

create or replace procedure getusndetails(id IN NUMBER, name OUT VARCHAR2) as

begin

select studentname

into name

from student\_tbl

where rollno=id;

dbms\_output.put\_line(name);

end;

------------------------------------------------------------------------------------------------------------

import java.sql.\*;

import java.util.Scanner;

public class SimpleCallableStatement {

public static void main(String a[]){

Connection con = null;

CallableStatement callSt = null;

String URL = "jdbc:oracle:thin:@localhost:1521:xe";

String USER = "hr";//"root" ;

String PASSWORD = "hr" ; //"mysql";

String DRIVER\_CLASS = "oracle.jdbc.driver.OracleDriver";

try {

Class.forName(DRIVER\_CLASS);

con = DriverManager.getConnection(URL, USER, PASSWORD);

System.out.println("Enter the USN");

Scanner input=new Scanner(System.in);

int input\_usn=Integer.parseInt(input.nextLine());

callSt = con.prepareCall("{call getusndetails(?,?)}");

callSt.setInt(1, input\_usn);

callSt.registerOutParameter(2, Types.VARCHAR);

callSt.execute();

String student\_name = callSt.getString(2);

System.out.println("Executed stored procedure!!!");

System.out.println("Student Name is "+ student\_name);

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} finally{

try{

if(callSt != null) callSt.close();

if(con != null) con.close();

} catch(Exception ex){}

}

}

}

1. Develop a library home page with option to search all books written by a particular author. Use servlet to make DB connectivity and JSP to display the list of books in tabular fashion to the client. Incase of error, JSP page must display error message to client. Assume suitable Data Base and drivers. Table can have columns author name and book title.

SQL\*Plus: Release 10.2.0.1.0 - Production on Wed Apr 27 20:06:02 2022

Copyright (c) 1982, 2005, Oracle. All rights reserved.

SQL> connect

Enter user-name: system

Enter password:

Connected.

SQL> create table library\_tbl( auth\_name varchar(30),book\_title varchar(30));

Table created.

SQL> insert into library\_tbl values('JK Rowling','HarryPotter Series1');

1 row created.

SQL> insert into library\_tbl values('JK Rowling','HarryPotter Series2');

1 row created.

SQL> insert into library\_tbl values('JK Rowling','HarryPotter Series3');

1 row created.

SQL> commit;

Commit complete.

SQL> select book\_title from library\_tbl where auth\_name='JK Rowling';

BOOK\_TITLE

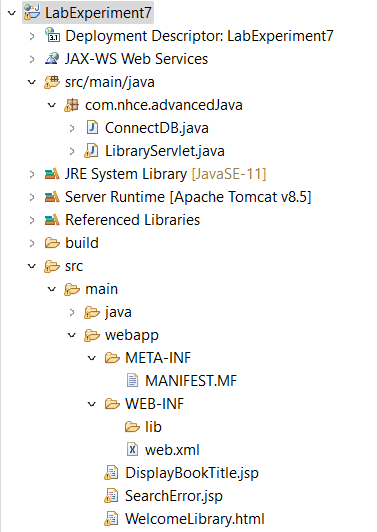
------------------------------

HarryPotter Series1

HarryPotter Series2

HarryPotter Series3

Create a Dynamic web project with structure as below



Place the oracle jar in buildpath of eclipse and also in the lib folder inside tomcat directory

**WelcomeLibrary.html**

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>ServletDemo Programming!!!</title>

</head>

<body>

<h1>Welcome to Library!!</h1>

<h3>Search Book by Author Name</h3>

<form name=*"Form1"* method=*"get"*

action=*"http://localhost:8080/LabExperiment7/LibraryServlet"*>

<table>

<tr>

<td><B>AuthorName</td>

<td><input type=*textbox* name=*"AuthorName"* size=*"25"* value=*""*></td>

</tr>

</table>

<input type=*submit* value=*"Search"*>

</form>

</body>

</html>

**DisplayBookTitle.jsp**

<%@page import=*"java.util.\*"*%>

<%@page errorPage=*"SearchError.jsp"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Library Search Result</title>

</head>

<body>

<h2>Book Title list</h2>

<table border=*'1'* cellpadding=*'10'*>

<%

String authname=(String)request.getAttribute("auth\_name");

ArrayList<String> arr = (ArrayList<String>)request.getAttribute("book\_title");

**if** (arr.size()==0)

out.println(" No Books Found with author " + authname);

**else**

{

out.println("Books written by author " + authname);

%>

<% **for**(**int** i=0;i<arr.size();i++) { %>

<tr>

<td>

<% out.println(i+1);%>

</td>

<td>

<% out.println(arr.get(i));%>

</td>

</tr>

<% }} %>

</table>

</body>

</html>

**SearchError.jsp**

<%@page isErrorPage=*"true"*%>

<html>

<body bgcolor=*"pink"*>

<h2>Result Not Found, Retry Again</h2><br>

<a href=*"WelcomeLibrary.html"*>Click here</a>

</body>

</html>

**web.xml**

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

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"* id=*"WebApp\_ID"* version=*"3.1"*>

<display-name>LabExperiment7</display-name>

<welcome-file-list>

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

</welcome-file-list>

</web-app>

**LibraryServlet.java**

**package** com.nhce.advancedJava;

**import** java.io.IOException;

**import** java.io.PrintWriter;

**import** java.util.ArrayList;

**import** java.util.Enumeration;

**import** javax.servlet.RequestDispatcher;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

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

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

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

/\*\*

\* Servlet implementation class LibraryServlet

\*/

@WebServlet("/LibraryServlet")

**public** **class** LibraryServlet **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

/\*\*

\* **@see** HttpServlet#HttpServlet()

\*/

**public** LibraryServlet() {

**super**();

}

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response)

**throws** ServletException, IOException {

response.setContentType("text/html");

String auth\_name = request.getParameter("AuthorName");

System.***out***.println(auth\_name);

ConnectDB connect = **new** ConnectDB();

ArrayList<String> book\_name = connect.SearchBookTitle(auth\_name);

request.setAttribute("auth\_name", auth\_name);

request.setAttribute("book\_title", book\_name);

RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/DisplayBookTitle.jsp");

dispatcher.forward(request, response);

}

}

**ConnectDB.java**

**package** com.nhce.advancedJava;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.util.ArrayList;

**public** **class** ConnectDB {

**public** ArrayList<String> SearchBookTitle(String book\_author) {

String bk\_title = **null**;

ArrayList<String> books = **new** ArrayList<String>();

**try** {

Class.*forName*("oracle.jdbc.driver.OracleDriver");

Connection con = DriverManager.*getConnection*("jdbc:oracle:thin:@localhost:1521:xe", "system", "tiger");

String selectquery = "select book\_title from library\_tbl where auth\_name=?";

PreparedStatement pst = con.prepareStatement(selectquery);

pst.setString(1, book\_author);

ResultSet rs = pst.executeQuery();

System.***out***.println("BOOK DETAILS");

**while** (rs.next()) {

bk\_title = rs.getString(1);

books.add(bk\_title);

System.***out***.println(bk\_title);

}

con.close();

} **catch** (Exception e) {

System.***out***.println(e);

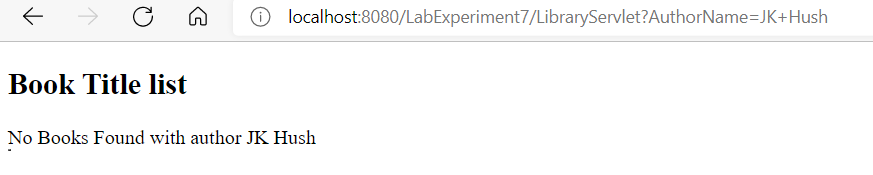
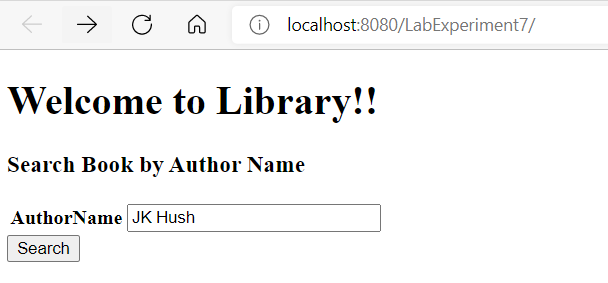
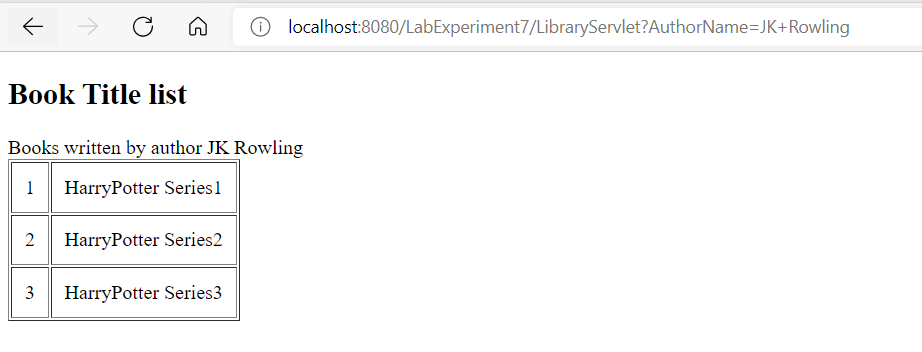
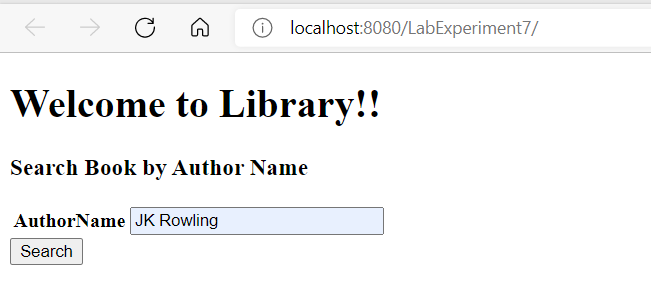
}

**return** books;

}

}

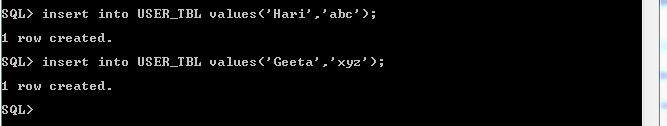
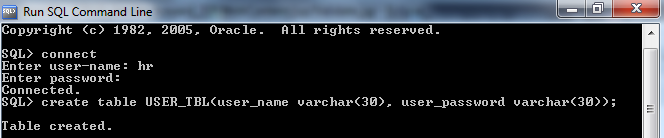
**Output**



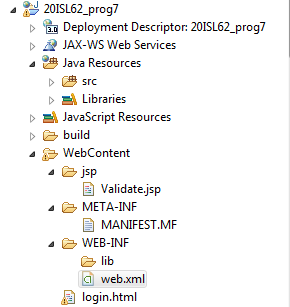
1. Design and develop a user login page and authenticate the user in a JSP page

using database. Assume user name and password to be the column of the USER

database. Establish connectivity using JDBC drivers.



Create a Dynamic web project with structure as below



Place the oracle jar in buildpath of eclipse and also in the lib folder inside tomcat directory

Validate.jsp

<%@ page import =*"java.sql.\*"* %>

<%

**try**{

String username = request.getParameter("username");

String password = request.getParameter("password");

System.out.print("Staring DB connection");

Class.forName("oracle.jdbc.driver.OracleDriver"); // Oracle database connection

Connection conn = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe", "hr", "hr");

System.out.println(username+" "+ password);

PreparedStatement pst = conn.prepareStatement("Select user\_name,user\_password from USER\_TBL where user\_name=? and user\_password=?");

pst.setString(1, username);

pst.setString(2, password);

ResultSet rs = pst.executeQuery();

**if**(rs.next())

{

out.println("Valid login credentials");

out.println("You are welcome " +rs.getString(1));

}

**else**

out.println("Invalid login credentials");

}

**catch**(Exception e){

e.printStackTrace();

out.println("Something went wrong !! Please try again");

}

%>

Login.html

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Login Demo with JSP</title>

</head>

<body>

<form method=*"post"* action=*"jsp/Validate.jsp"*>

<center>

<table border=*"1"* cellpadding=*"5"* cellspacing=*"2"*>

<thead>

<tr>

<th colspan=*"2"*>Login Here</th>

</tr>

</thead>

<tbody>

<tr>

<td>Username</td>

<td><input type=*"text"* name=*"username"* required/></td>

</tr>

<tr>

<td>Password</td>

<td><input type=*"password"* name=*"password"* required/></td>

</tr>

<tr>

<td colspan=*"2"* align=*"center"*><input type=*"submit"* value=*"Login"* />

&nbsp;&nbsp;

<input type=*"reset"* value=*"Reset"* />

</td>

</tr>

</tbody>

</table>

</center>

</form>

</body>

</html>

Web.xml

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

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://java.sun.com/xml/ns/javaee"* xmlns:web=*"http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"* xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"* id=*"WebApp\_ID"* version=*"3.0"*>

<display-name>20ISL66\_Program7</display-name>

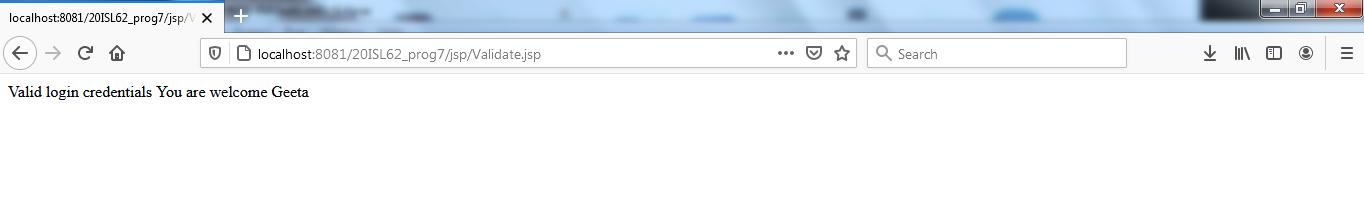
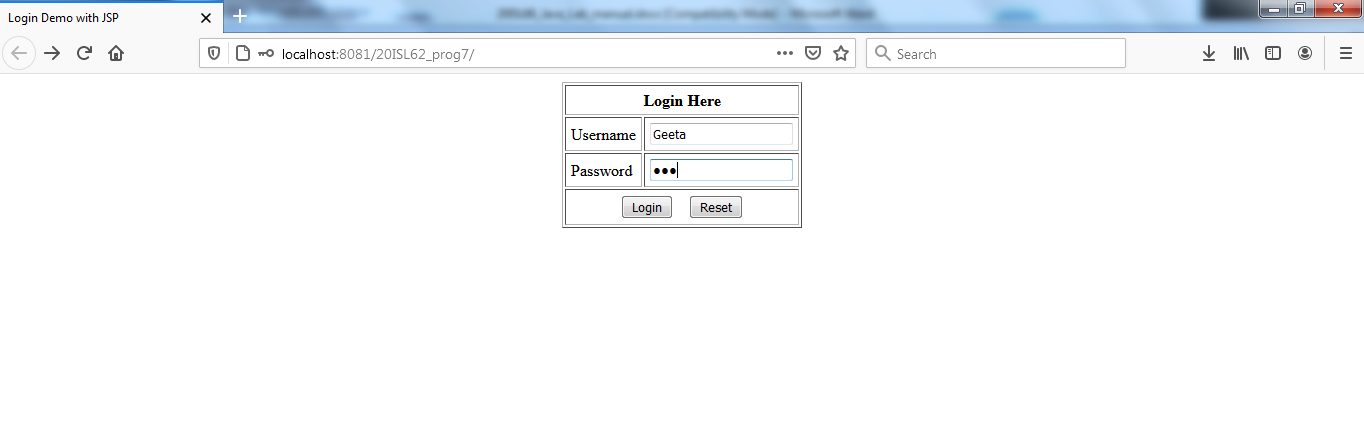
<welcome-file-list>

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

</welcome-file-list>

</web-app>

Output



1. Create a HTML Page, which asks the user to enter a number in a textbox. On

clicking the submit button, it places the request to a Servlet. The Servlet generates all Prime numbers which are less than the given number and adds them to an ArrayList and forwards the control to a JSP page. The JSP page iterates through the ArrayList and prints them in a tabular format. Apply RequestDispatcher methods to achieve the same.

**WelcomeServlet.java**

package com.nhce.program5b;

import java.io.IOException;

import java.util.ArrayList;

import javax.servlet.RequestDispatcher;

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("/WelcomeServlet")

public class WelcomeServlet extends HttpServlet {

public WelcomeServlet() {

super();

}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String str = request.getParameter("Number");

int number= Integer.parseInt(str);

ArrayList<Integer> returnarr= new ArrayList<Integer>();

returnarr= generatePrimeNumbers(number);

request.setAttribute("primenumber", returnarr);

RequestDispatcher dispatcher = getServletContext().getRequestDispatcher("/jsp/NumberDisplay.jsp");

dispatcher.forward(request, response);

}

ArrayList generatePrimeNumbers(int number)

{

ArrayList<Integer> arr = new ArrayList<Integer>();

for(int i=1; i < number; i++){

boolean isPrime = true;

for(int j=2; j < i ; j++){

if(i % j == 0){

isPrime = false;

break;

}

}

// print the number

if(isPrime){

System.out.print(i + " ");

arr.add(i);

}

}

return arr;

}

}

**NumberHome.html**

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Servlet forwarding Program</title>

</head>

<body>

<form name=*"Form1"* method=*"post"*

action=*"http://localhost:8080/Servlet-JSP-Program/WelcomeServlet"*>

<B>Enter a number:

<input type=*textbox* name=*"Number"* size=*"25"* value=*""*>

<input type=*submit* value=*"Submit"*>

</form>

</body>

</html>

**NumberDisplay.jsp**

<%@page import=*"java.util.\*"*%>

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=windows-1255"*>

<title>Insert title here</title>

</head>

<body>

<h2>PrimeNumber list</h2>

<table>

<%

ArrayList<Integer> arr = (ArrayList<Integer>)request.getAttribute("primenumber");

%>

<% **for**(**int** i=1;i<arr.size();i++) { %>

<tr>

<td><% out.println(i);%> </td>

<td><% out.println(arr.get(i));%> </td>

</tr>

<% } %>

</table>

</body>

</html>

**Web.xml**

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

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://java.sun.com/xml/ns/javaee"* xmlns:web=*"http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"* xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"* id=*"WebApp\_ID"* version=*"3.0"*>

<display-name>Servlet-JSP-Program</display-name>

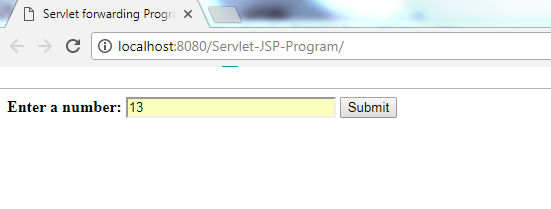
<welcome-file-list>

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

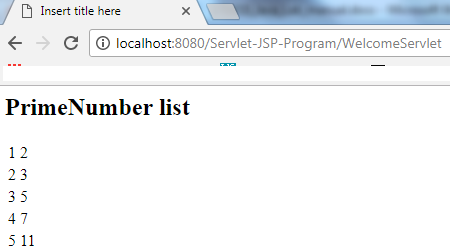
</welcome-file-list>

</web-app>

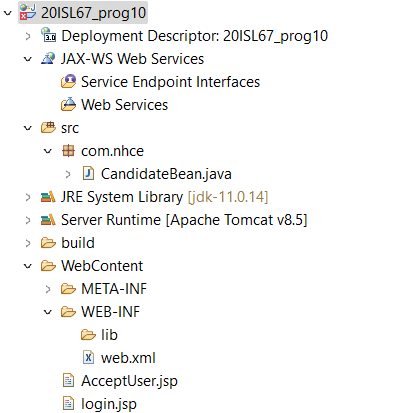
**Output**



**Response**



1. Design a Seminar registration form with Candidate Name, Email id, Favourite Courses enrolled. Apply usebean tag in JSP to receive and send the data to Candidate Java Bean and reply with successful registration and display the registered data.



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>SEMINAR REGISTRATION PAGE</title>

</head>

<body>

<h1> SEMINAR REGISTRATION FORM </h1>

<%-- The form data will be passed to acceptuser.jsp

for validation on clicking submit

--%>

<form method =*"POST"* action=*"AcceptUser.jsp"*>

Enter Candidate Name : <input type=*"text"* name=*"candidateName"*><br/><br/>

Enter EmailID : <input type=*"text"* name =*"candidate\_Emailid"*><br/><br/>

Choose Favourite Course Elements<br/>

<input type=*"checkbox"* name=*"candidate\_Course"* value=*"Java"*>Java <br/>

<input type=*"checkbox"* name=*"candidate\_Course"* value=*"Python"*>Python <br/>

<input type=*"checkbox"* name=*"candidate\_Course"* value=*"C"*>C<br/>

<input type=*"checkbox"* name=*"candidate\_Course"* value=*"C++"*>C++<br/>

<input type=*"checkbox"* name=*"candidate\_Course"* value=*"GoLang "*>GoLang<br/>

<br/><br/>

<input type =*"submit"* value=*"SUBMIT"*>

</form>

</body>

</html>

AcceptUser.jsp

<%@ page language=*"java"* %>

<%@ page import=*"java.util.\*"* %>

<%!

%>

<jsp:useBean id=*"formHandler"* class=*"com.nhce.CandidateBean"*>

<jsp:setProperty name=*"formHandler"* property=*"\*"*/>

</jsp:useBean>

<%

**if** (formHandler.validate()) {

%>

<html>

<body>

<font size=*5*>STUDENT REGISTRATION SUCCESSFUL!!</font>

<p>Candidate Name: <jsp:getProperty name=*"formHandler"* property=*"candidateName"*/></p>

<p>Candidate Emailid : <jsp:getProperty name=*"formHandler"* property=*"candidate\_Emailid"*/></p>

<b>Favourite Courses:</b>

<br>

<%

String[] faveTech = formHandler.getcandidate\_Course();

**if** (!faveTech[0].equals("1")) {

out.println("<ul>");

**for** (**int** i=0; i<faveTech.length; i++)

out.println("<li>"+faveTech[i]);

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

} **else** out.println("Nothing was selected");

%>

</body>

</html>

<%

} **else** {

out.println("SEMINAR REGISTRATION FAILED! Fill the fields");

}

%>

CandiadteBean.java

**package** com.nhce;

**public** **class** CandidateBean {

**private** String candidateName;

**private** String candidate\_Emailid;

**private** String[] candidate\_Course;

**public** CandidateBean()

{

candidateName="";

candidate\_Emailid="";

candidate\_Course = **new** String[] { "1" };

}

**public** **boolean** validate() {

**boolean** bool=**true**;

**if** (candidateName.equals("")) {

candidateName="";

bool=**false**;

}

**if** (candidate\_Emailid.equals("")) {

candidate\_Emailid="";

bool=**false**;

}

**return** bool;

}

**public** String getcandidateName() {

**return** candidateName;

}

**public** **void** setcandidateName(String candidateName) {

**this**.candidateName = candidateName;

}

**public** String getcandidate\_Emailid() {

**return** candidate\_Emailid;

}

**public** **void** setcandidate\_Emailid(String candidate\_Emailid) {

**this**.candidate\_Emailid = candidate\_Emailid;

}

**public** String[] getcandidate\_Course() {

**return** candidate\_Course;

}

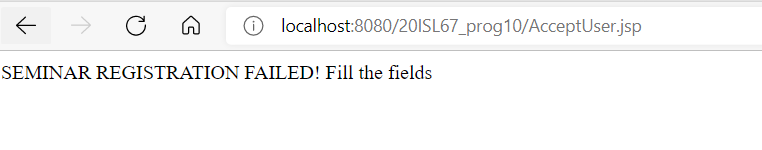
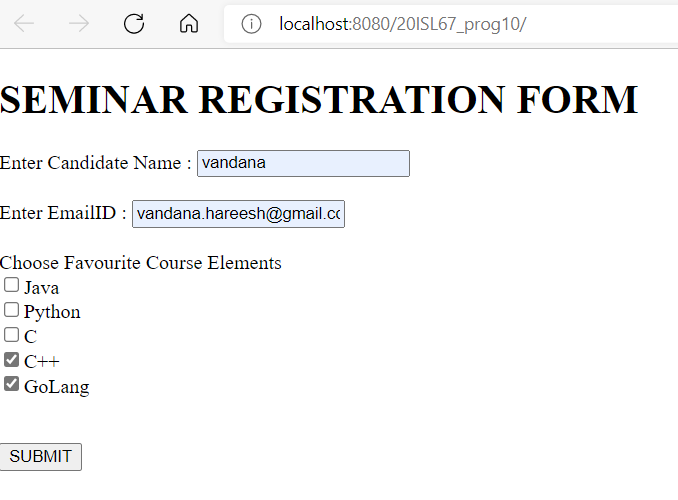
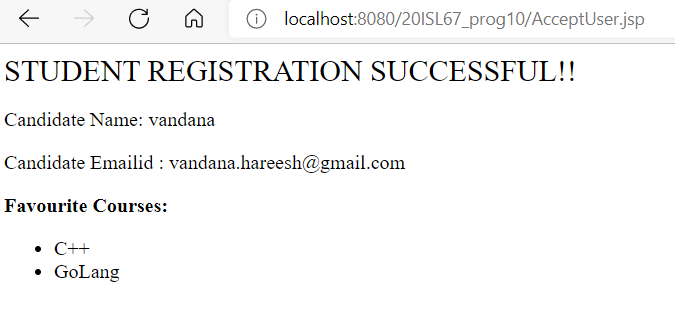
**public** **void** setcandidate\_Course(String[] candidate\_Course) {

**this**.candidate\_Course = candidate\_Course;

}

}

OUTPUT



11.Create a Spring based examination conduction application to demonstrate the concept of dependency injection. Bean USER has the properties id, name, email id. Bean ANSWER has attributes id, answer, postedDate. Bean QUESTION has id, name, answer.

Put Spring core jars ion java project buildpath

Anwer.java

**package** com.nhce;

**import** java.util.Date;

**public** **class** Answer {

**private** **int** id;

**private** String answer;

**private** Date postedDate;

**public** Answer() {}

**public** Answer(**int** id, String answer, Date postedDate) {

**super**();

**this**.id = id;

**this**.answer = answer;

**this**.postedDate = postedDate;

}

**public** String toString(){

**return** "Id:"+id+" Answer:"+answer+" Posted Date:"+postedDate;

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getAnswer() {

**return** answer;

}

**public** **void** setAnswer(String answer) {

**this**.answer = answer;

}

**public** Date getPostedDate() {

**return** postedDate;

}

**public** **void** setPostedDate(Date postedDate) {

**this**.postedDate = postedDate;

}

}

User.java

**package** com.nhce;

**public** **class** User {

**private** **int** id;

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getEmail() {

**return** email;

}

**public** **void** setEmail(String email) {

**this**.email = email;

}

**private** String name,email;

**public** User() {}

**public** User(**int** id, String name, String email) {

**super**();

**this**.id = id;

**this**.name = name;

**this**.email = email;

}

**public** String toString(){

**return** "Id:"+id+" Name:"+name+" Email Id:"+email;

}

}

Question.java

package com.nhce;

import java.util.Iterator;

import java.util.Map;

import java.util.Set;

import java.util.Map.Entry;

public class Question {

private int id;

private String name;

private Map<Answer,User> answers;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public Map<Answer, User> getAnswers() {

return answers;

}

public void setAnswers(Map<Answer, User> answers) {

this.answers = answers;

}

public void displayInfo(){

System.out.println("question id:"+id);

System.out.println("question name:"+name);

System.out.println("Answers....");

Set<Entry<Answer, User>> set=answers.entrySet();

Iterator<Entry<Answer, User>> itr=set.iterator();

while(itr.hasNext()){

Entry<Answer, User> entry=itr.next();

Answer ans=entry.getKey();

User user=entry.getValue();

System.out.println("Answer Information:");

System.out.println(ans);

System.out.println("Posted By:");

System.out.println(user);

}

}

}

Test.java

package com.nhce;

import org.springframework.beans.factory.BeanFactory;

import org.springframework.beans.factory.xml.XmlBeanFactory;

import org.springframework.core.io.ClassPathResource;

import org.springframework.core.io.Resource;

public class Test {

public static void main(String[] args) {

Resource r=new ClassPathResource("applicationContext.xml");

BeanFactory factory=new XmlBeanFactory(r);

Question q=(Question)factory.getBean("question");

q.displayInfo();

}

}

applicationContext.xml

<?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"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd"*>

<bean id=*"answer1"* class=*"com.nhce.Answer"*>

<property name=*"id"* value=*"1"*></property>

<property name=*"answer"* value=*"Java is a Programming Language"*></property>

<property name=*"postedDate"* value=*"12/12/2001"*></property>

</bean>

<bean id=*"answer2"* class=*"com.nhce.Answer"*>

<property name=*"id"* value=*"2"*></property>

<property name=*"answer"* value=*"Java is a Platform"*></property>

<property name=*"postedDate"* value=*"12/12/2003"*></property>

</bean>

<bean id=*"user1"* class=*"com.nhce.User"*>

<property name=*"id"* value=*"1"*></property>

<property name=*"name"* value=*"Arun Kumar"*></property>

<property name=*"email"* value=*"arun@gmail.com"*></property>

</bean>

<bean id=*"user2"* class=*"com.nhce.User"*>

<property name=*"id"* value=*"2"*></property>

<property name=*"name"* value=*"Varun Kumar"*></property>

<property name=*"email"* value=*"Varun@gmail.com"*></property>

</bean>

<bean id=*"question"* class=*"com.nhce.Question"*>

<property name=*"id"* value=*"1"*></property>

<property name=*"name"* value=*"What is Java?"*></property>

<property name=*"answers"*>

<map>

<entry key-ref=*"answer1"* value-ref=*"user1"*></entry>

<entry key-ref=*"answer2"* value-ref=*"user2"*></entry>

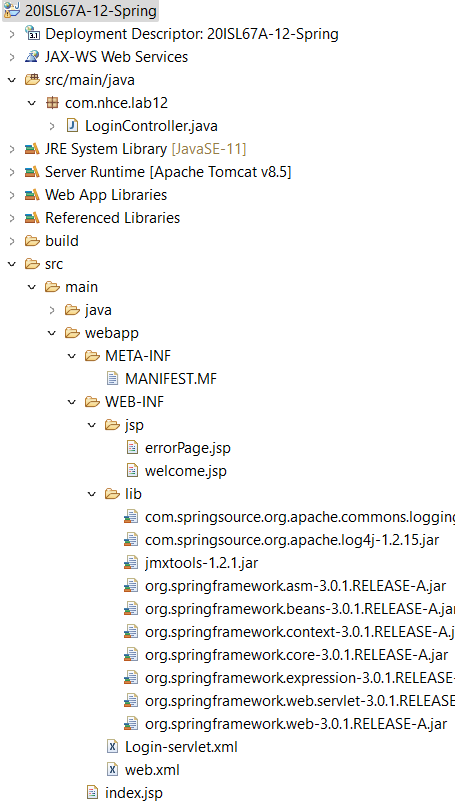
</map>

</property>

</bean>

</beans>

1. Design and develop a user login based web application based on Spring Model view Controller (MVC) components.



### Create a dynamic web project

### Include core and web module related spring jars in build path and inside lib folder

### Create a Java class LoginController under the com.nhce.lab12 package.

### Create Spring configuration files web.xml and Login-servlet.xml under the webapp/WEB-INF folder.

### Create view file welcome.jsp, errorPage.jsp under jsp folder inside WEB-INF in webapp

### Index.jsp , landing page can be inside webapp folder

web.xml

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

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"* id=*"WebApp\_ID"* version=*"3.1"*>

<display-name>20ISL67A-12-Spring</display-name>

<servlet>

<servlet-name>Login</servlet-name>

<servlet-class>

org.springframework.web.servlet.DispatcherServlet

</servlet-class>

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

</servlet>

<servlet-mapping>

<servlet-name>Login</servlet-name>

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

</servlet-mapping>

</web-app>

index.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"* pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<title>Spring MVC login example.</title>

</head>

<body>

<form action=*"Login.html"* method=*"post"*>

UserName:<input type=*"text"* name=*"userName"*/>

<br/><br/>

Password:<input type=*"password"* name=*"password"*/>

<br/><br/>

<input type=*"submit"* value=*"Login"*/>

</form>

</body>

</html>

Login-servlet.xml

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

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

xmlns:context=*"http://www.springframework.org/schema/context"*

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

xsi:schemaLocation=*"*

*http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context.xsd*

*http://www.springframework.org/schema/mvc*

*http://www.springframework.org/schema/mvc/spring-mvc.xsd"*>

<context:component-scan base-package=*"com.nhce.lab12"* />

<bean class=

*"org.springframework.web.servlet.view.InternalResourceViewResolver"*>

<property name=*"prefix"* value=*"/WEB-INF/jsp/"* />

<property name=*"suffix"* value=*".jsp"* />

</bean>

</beans>

errorPage.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<title>Spring MVC login example.</title>

</head>

<body>

<h2>${message}</h2><br/>

<jsp:include page=*"/index.jsp"*></jsp:include>

</body>

</html>

welcome.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"* pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE >

<html>

<head>

<title>Spring MVC login example.</title>

</head>

<body>

<h2>${message}</h2>

</body>

</html>

LoginController.java

package com.nhce.lab12;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.servlet.ModelAndView;

@Controller

public class LoginController {

@RequestMapping("/Login")

public ModelAndView Login(HttpServletRequest request,

HttpServletResponse response) {

String userName=request.getParameter("userName");

String password=request.getParameter("password");

String message;

if(userName != null &&

!userName.equals("")

&& userName.equals("ISE") &&

password != null &&

!password.equals("") &&

password.equals("123")){

message = "Welcome " +userName + ".";

return new ModelAndView("welcome",

"message", message);

}else{

message = "Wrong username or password.";

return new ModelAndView("errorPage",

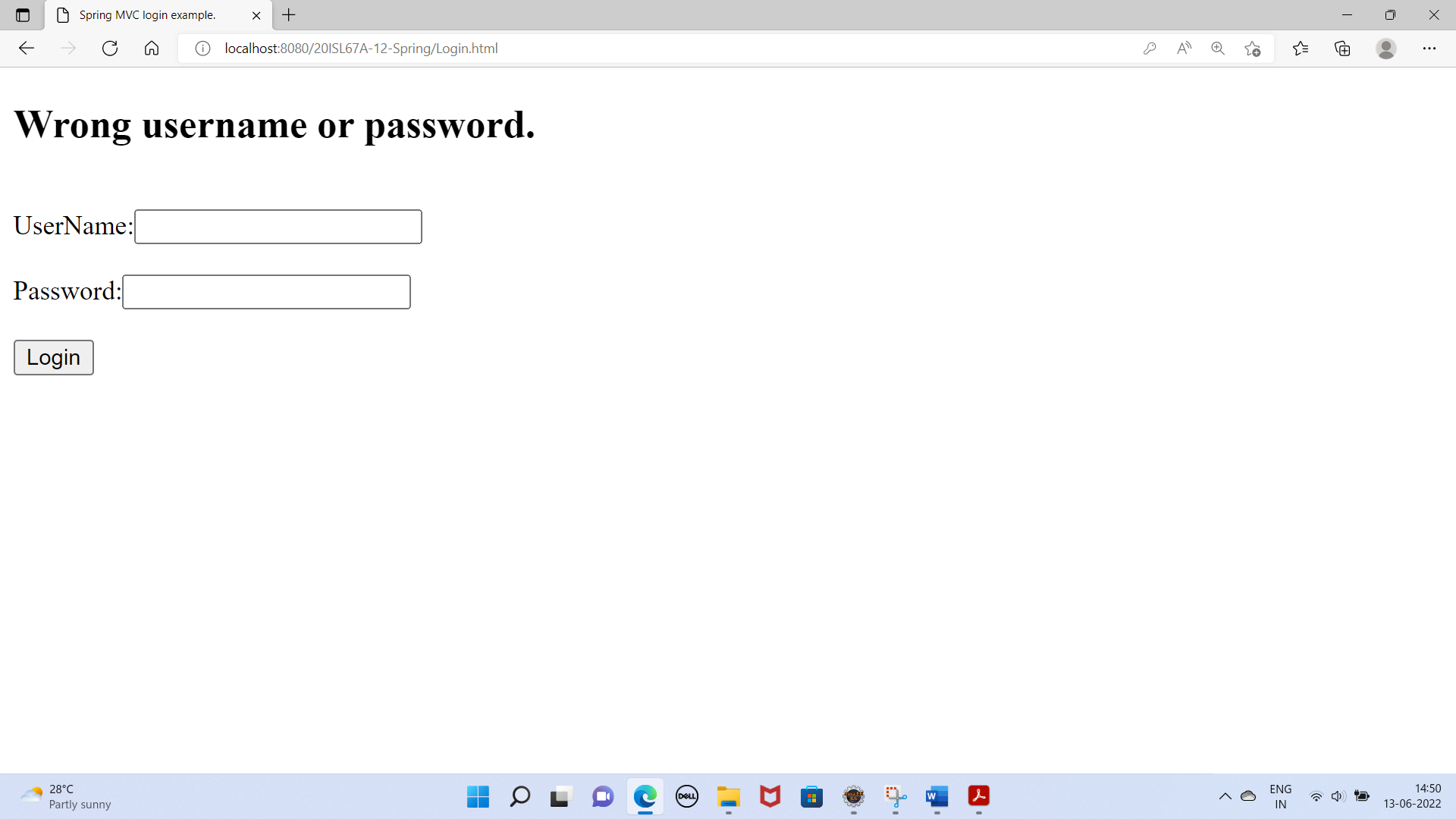
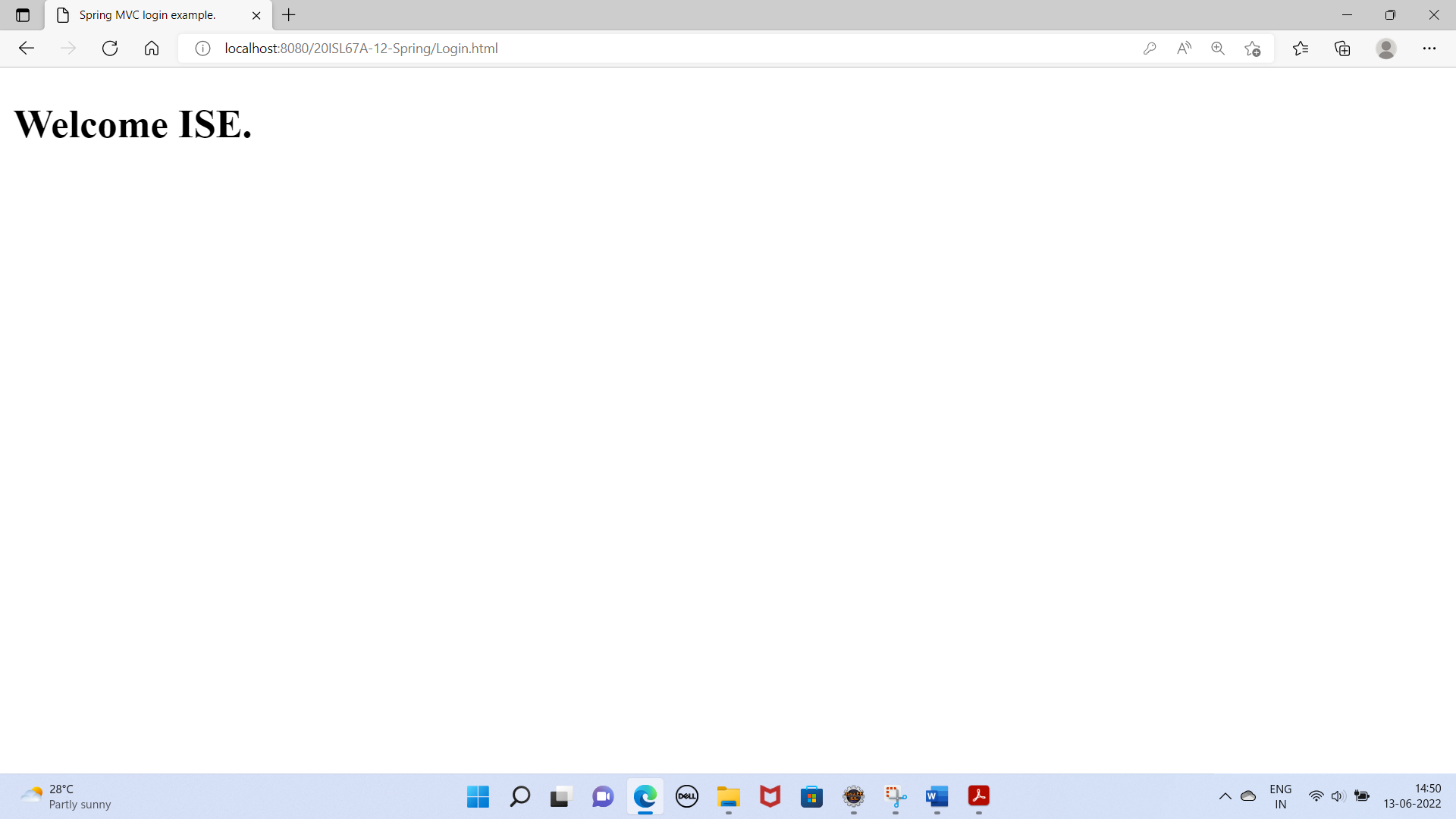
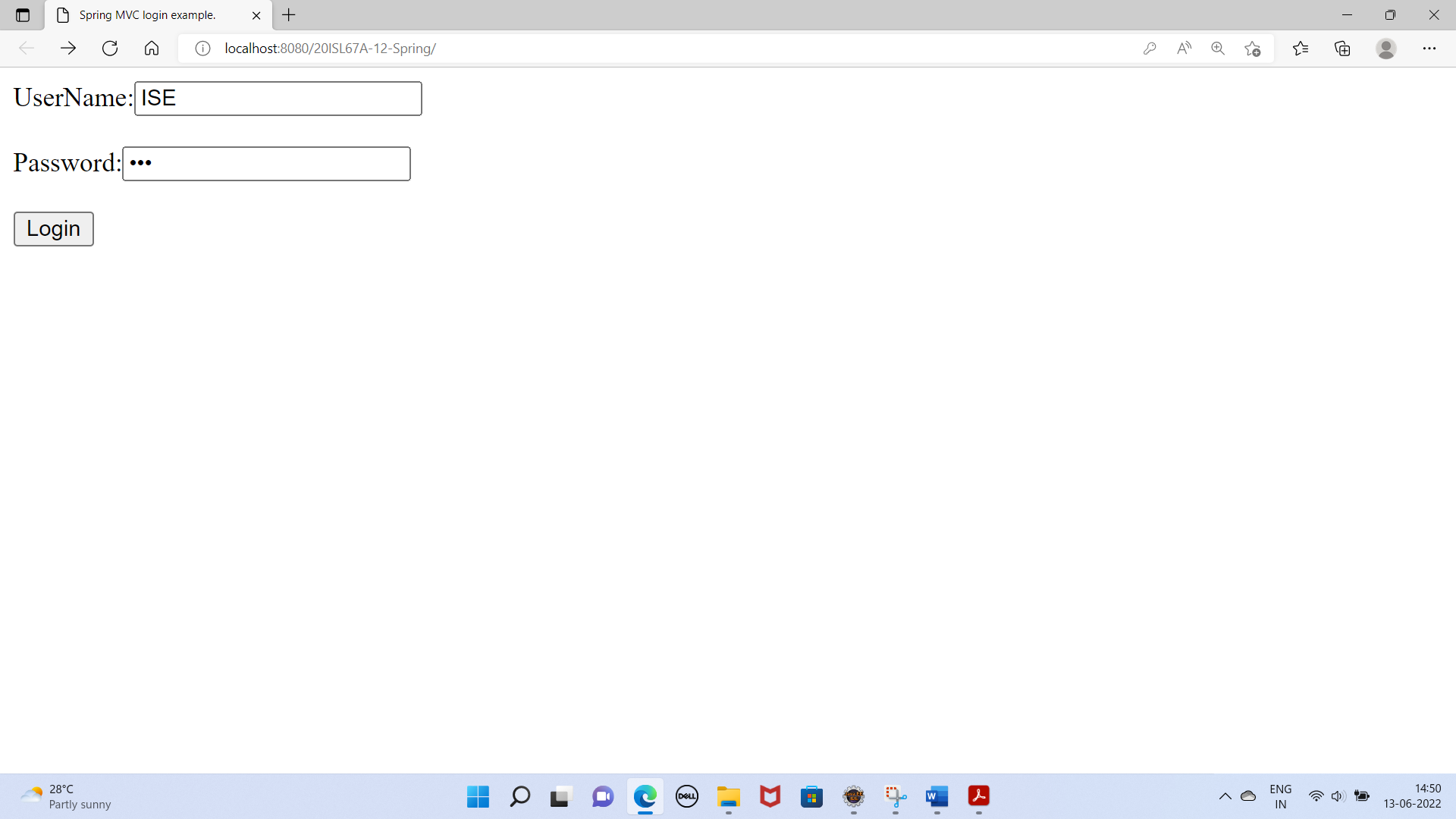
"message", message);

}

}

}

OUTPUT



Extra Question

Design and develop a program to demonstrate the MVC pattern using spring

framework. Implement the Model view Controller components to interact with

spring framework.

|  |  |
| --- | --- |
| Create a dynamic web projectInclude core and web module related spring jars in build pathCreate a Java class HelloController under the com.nhce package.Create Spring configuration files web.xml and HelloMVC-servlet.xml under the WebContent/WEB-INF folder.Create view file hello.jsp under WebContent |  |

### HelloController.java

**package** com.nhce;

**import** org.springframework.stereotype.Controller;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestMethod;

**import** org.springframework.ui.ModelMap;

@Controller

**public** **class** HelloController {

@RequestMapping("/")

**public** String display()

{

**return** "index";

}

}

HelloMVC-servlet.xml

<?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:context=*"http://www.springframework.org/schema/context"*

xmlns:mvc=*"http://www.springframework.org/schema/mvc"*

xsi:schemaLocation=*"*

*http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context.xsd*

*http://www.springframework.org/schema/mvc*

*http://www.springframework.org/schema/mvc/spring-mvc.xsd"*>

<!-- Provide support for component scanning -->

<context:component-scan base-package=*"com.nhce"* />

</beans>

Web.xml

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

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://java.sun.com/xml/ns/javaee"* xmlns:web=*"http://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd"* xsi:schemaLocation=*"http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"* id=*"WebApp\_ID"* version=*"3.0"*>

<display-name>SpringMVCHello</display-name>

<servlet>

<servlet-name>HelloMVC</servlet-name>

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

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

</servlet>

<servlet-mapping>

<servlet-name>HelloMVC</servlet-name>

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

</servlet-mapping>

</web-app>

index.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

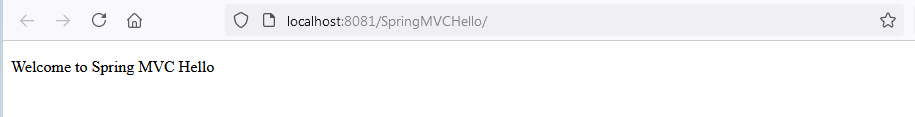
</head>

<body>

<p>Welcome to Spring MVC Hello </p>

</body>

</html>



Extra Program

1. Design and create swing based program to display the coordinates of a mouse

pointer in a label component.

package com.nhce.advancedjava;

import javax.swing.JFrame;

import javax.swing.JLabel;

import java.awt.event.MouseEvent;

import java.awt.event.MouseMotionAdapter;

public class MouseCoordinateProgram1 extends JFrame {

private JLabel coords;

public MouseCoordinateProgram1() {

setTitle("Mouse move events");

coords = new JLabel("");

coords.setBounds(50, 100, 250, 20);

add(coords);

setSize(400, 400);

setLayout(null);

setVisible(true);

addMouseMotionListener(new MouseMotionAdapter() {

@Override

public void mouseMoved(MouseEvent e) {

int x = e.getX();

int y = e.getY();

String text = String.*format*("x: %d, y: %d", x, y);

coords.setText(text);

}

});

}

public static void main(String[] args) {

new MouseCoordinateProgram1();

}

}

