**Practical : 1**

**Write a program to demonstrate status of key on an Applet window such as KeyPressed, KeyReleased, KeyUp, KeyDown.**

**Program :**

**KeyEventDemo.java**

import java.awt.\*;

import java.applet.\*;

import java.awt.event.\*;

public class KeyEventDemo extends Applet implements KeyListener

{

String msg = "";

public void init()

{

addKeyListener(this);

}

public void keyReleased(KeyEvent k)

{

showStatus("Key Released");

repaint();

}

public void keyTyped(KeyEvent k)

{

showStatus("Key Typed");

repaint();

}

public void keyPressed(KeyEvent k)

{

showStatus("Key Pressed");

repaint();

}

public void paint(Graphics g)

{

g.drawString(msg, 10, 10);

}

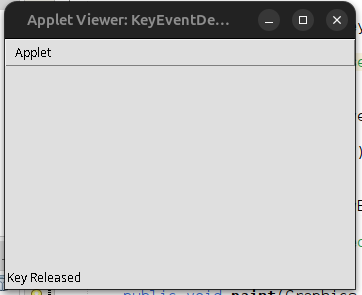
}

**Exp1.html**

<applet code="KeyEventDemo" height="400" width="400">

</applet>

**Output :**



**Practical : 2**

**Write a program to create a frame using AWT. Implement mouseClicked, mouseEntered() and mouseExited() events. Frame should become visible when the mouse enters it.**

**Program :**

import java.awt.\*;

import java.awt.event.\*;

class Exp2 extends Frame implements MouseListener {

Exp2() {

setTitle("Mouse Event Frame");

setSize(400, 300);

setLayout(new FlowLayout());

add(new Label("Mouse Events: Enter, Exit, Click"));

addMouseListener(this);

setVisible(false); // Frame hidden initially

}

public void mouseEntered(MouseEvent e) {

System.out.println("Mouse Entered");

setVisible(true);

}

public void mouseExited(MouseEvent e) {

System.out.println("Mouse Exited");

setVisible(false);

}

public void mouseClicked(MouseEvent e) {

System.out.println("Mouse Clicked");

}

public void mousePressed(MouseEvent e) {}

public void mouseReleased(MouseEvent e) {}

public static void main(String[] args) {

// Helper frame that stays visible so you can trigger the events

Frame baseFrame = new Frame("Trigger Area");

baseFrame.setSize(300, 200);

baseFrame.setLayout(new FlowLayout());

baseFrame.add(new Label("Move mouse here to show the frame"));

baseFrame.setVisible(true);

// Frame that appears/disappears based on mouse

Exp2 mouseFrame = new Exp2();

// Add mouse listener to the base frame to trigger visibility

baseFrame.addMouseListener(new MouseAdapter() {

public void mouseEntered(MouseEvent e) {

mouseFrame.setVisible(true);

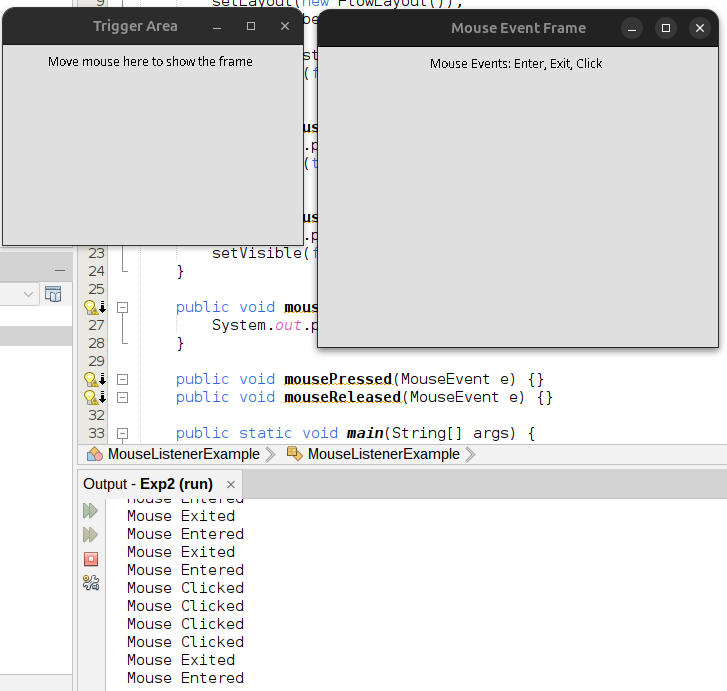
}

});

}

}

**Output :**



**Practical : 3**

**Develop a GUI which accepts the information regarding the marks for all the subjects of a student in the examination. Display the result for a student in a separate window.**

**Program :**

import javax.swing.\*;

import java.awt.\*;

public class Exp3 {

public static void main(String[] args) {

SwingUtilities.invokeLater(SimpleMarksWindow::new);

}

}

class SimpleMarksWindow extends JFrame {

JTextField[] markFields = new JTextField[5];

String[] subjects = {"PCS", "CS", "SS", "OOP", "PBL"};

SimpleMarksWindow() {

setTitle("Enter Subject Marks");

setSize(300, 300);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

setLayout(new GridLayout(6, 2, 10, 10)); ff

for (int i = 0; i < 5; i++) {

add(new JLabel(subjects[i] + ":"));

markFields[i] = new JTextField();

add(markFields[i]);

}

JButton submit = new JButton("Submit");

submit.addActionListener(e -> showResult());

add(submit);

add(new JLabel()); // for grid alignment

setVisible(true);

}

void showResult() {

int total = 0;

boolean pass = true;

for (JTextField field : markFields) {

String input = field.getText().trim();

if (!input.matches("\\d+")) {

JOptionPane.showMessageDialog(this, "Enter valid integer marks.");

return;

}

int mark = Integer.parseInt(input);

if (mark < 0 || mark > 100) {

JOptionPane.showMessageDialog(this, "Marks must be between 0 and 100.");

return;

}

if (mark < 35) pass = false;

total += mark;

}

double percentage = total / 5.0;

String result = pass ? "Pass" : "Fail";

JOptionPane.showMessageDialog(this,

"Total: " + total +

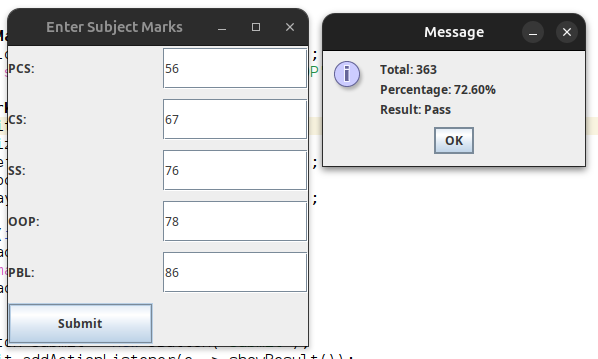
"\nPercentage: " + String.format("%.2f", percentage) + "%" +

"\nResult: " + result);

}

}

**Output :**



**Practical : 4**

**Write a program to insert and retrieve the data from the database using JDBC.**

**Program :**

**StudentDB.java**

import java.sql.\*;

public class StudentDB {

public static void main(String[] args) {

String url = "jdbc:mysql://localhost:3306/studentdb";

String user = "root";

String password = "root123";

try {

// 1. Load driver and establish connection

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

Connection con = DriverManager.getConnection(url, user, password);

// 2. Create table if not exists

Statement stmt = con.createStatement();

stmt.executeUpdate("CREATE TABLE IF NOT EXISTS students (name VARCHAR(50), marks INT)");

// 3. Insert sample data

stmt.executeUpdate("INSERT INTO students VALUES ('Rahul', 85)");

stmt.executeUpdate("INSERT INTO students VALUES ('Priya', 92)");

// 4. Retrieve and display data

ResultSet rs = stmt.executeQuery("SELECT \* FROM students");

System.out.println("Student Records:");

while(rs.next()) {

System.out.println(rs.getString("name") + "\t" + rs.getInt("marks"));

}

// 5. Close connection

con.close();

} catch(Exception e) {

System.out.println(e);

}

}

}

**MySQL :**

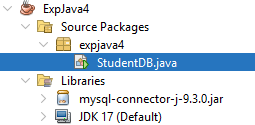
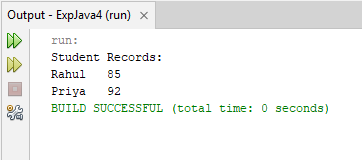
SHOW TABLES;

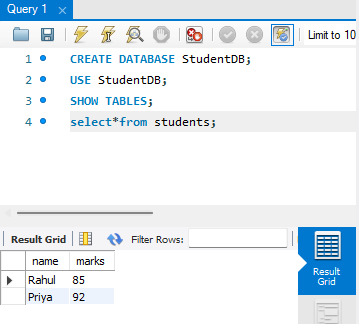
USE studentdb;

CREATE DATABASE StudentDB;

select\*from students;

**Output :**

****

****

**Practical : 5**

**Write program with suitable example to develop your remote interface, implement your RMI server, implement application that create your server.**

**Program :**

**Addition.java (Remote Interface)**

import java.rmi.Remote;

import java.rmi.RemoteException;

public interface Addition extends Remote {

int add(int a, int b) throws RemoteException;

}

**AdditionImpl.java (Server Implementation)**

import java.rmi.server.UnicastRemoteObject;

import java.rmi.RemoteException;

public class AdditionImpl extends UnicastRemoteObject implements Addition {

public AdditionImpl() throws RemoteException {

super();}

public int add(int a, int b) throws RemoteException {

return a + b;}}

**AdditionServer.java (RMI Server)**

import java.rmi.registry.LocateRegistry;

import java.rmi.registry.Registry;

public class AdditionServer {

public static void main(String[] args) {

try {

AdditionImpl obj = new AdditionImpl();

Registry registry = LocateRegistry.createRegistry(1099);

registry.rebind("AddService", obj);

System.out.println("Addition Server is ready...");

} catch (Exception e) {

e.printStackTrace();

}}}

**AdditionClient.java (RMI Client)**

import java.rmi.registry.LocateRegistry;

import java.rmi.registry.Registry;

import java.util.Scanner;

public class AdditionClient {

public static void main(String[] args) {

try {

// Getting input from the user

Scanner sc = new Scanner(System.in);

System.out.print("Enter first number: ");

int num1 = sc.nextInt();

System.out.print("Enter second number: ");

int num2 = sc.nextInt();

Registry registry = LocateRegistry.getRegistry("localhost", 1099);

Addition stub = (Addition) registry.lookup("AddService");

int result = stub.add(num1, num2);

System.out.println("Result from Server: " + result);

} catch (Exception e) {

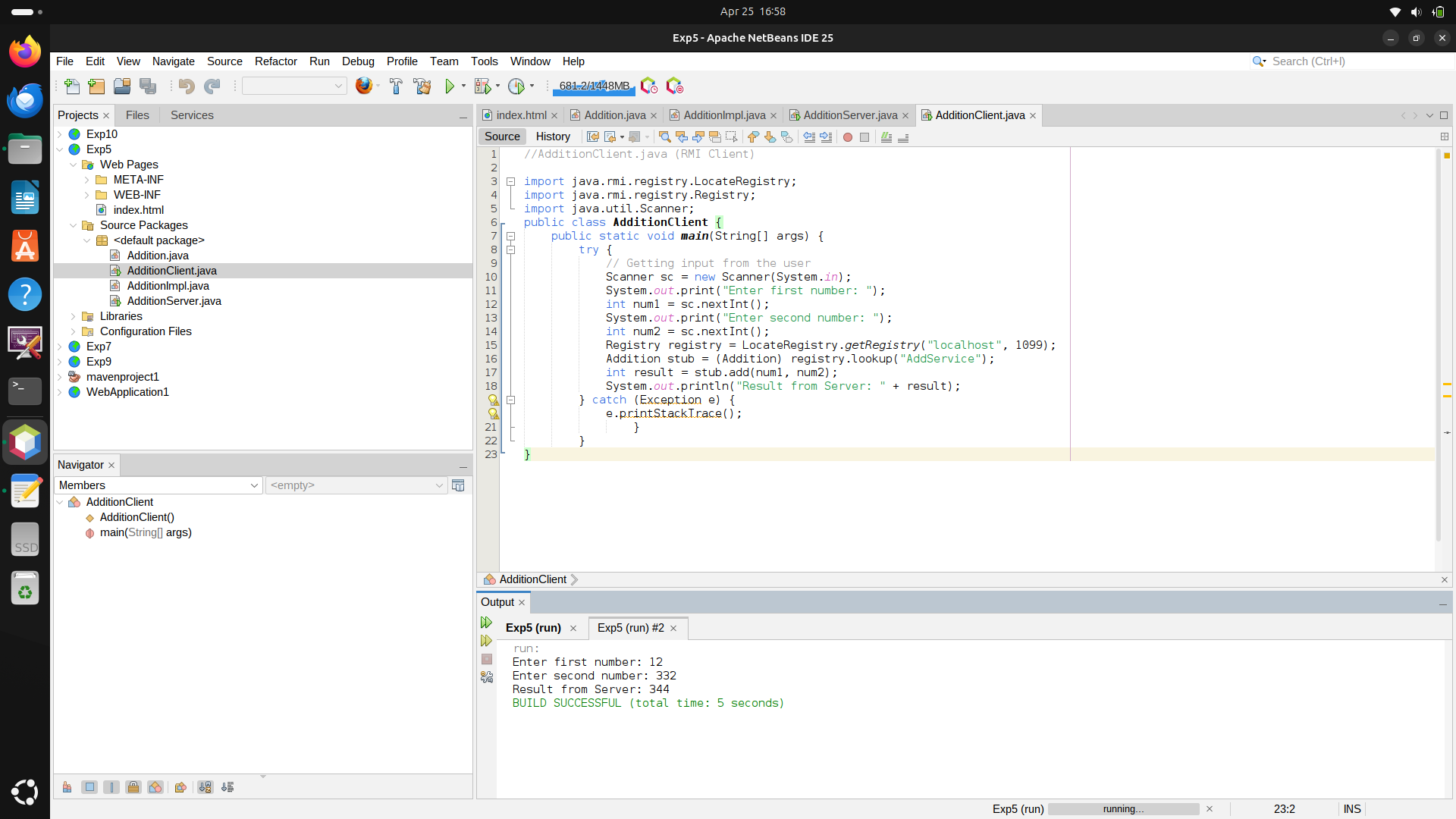
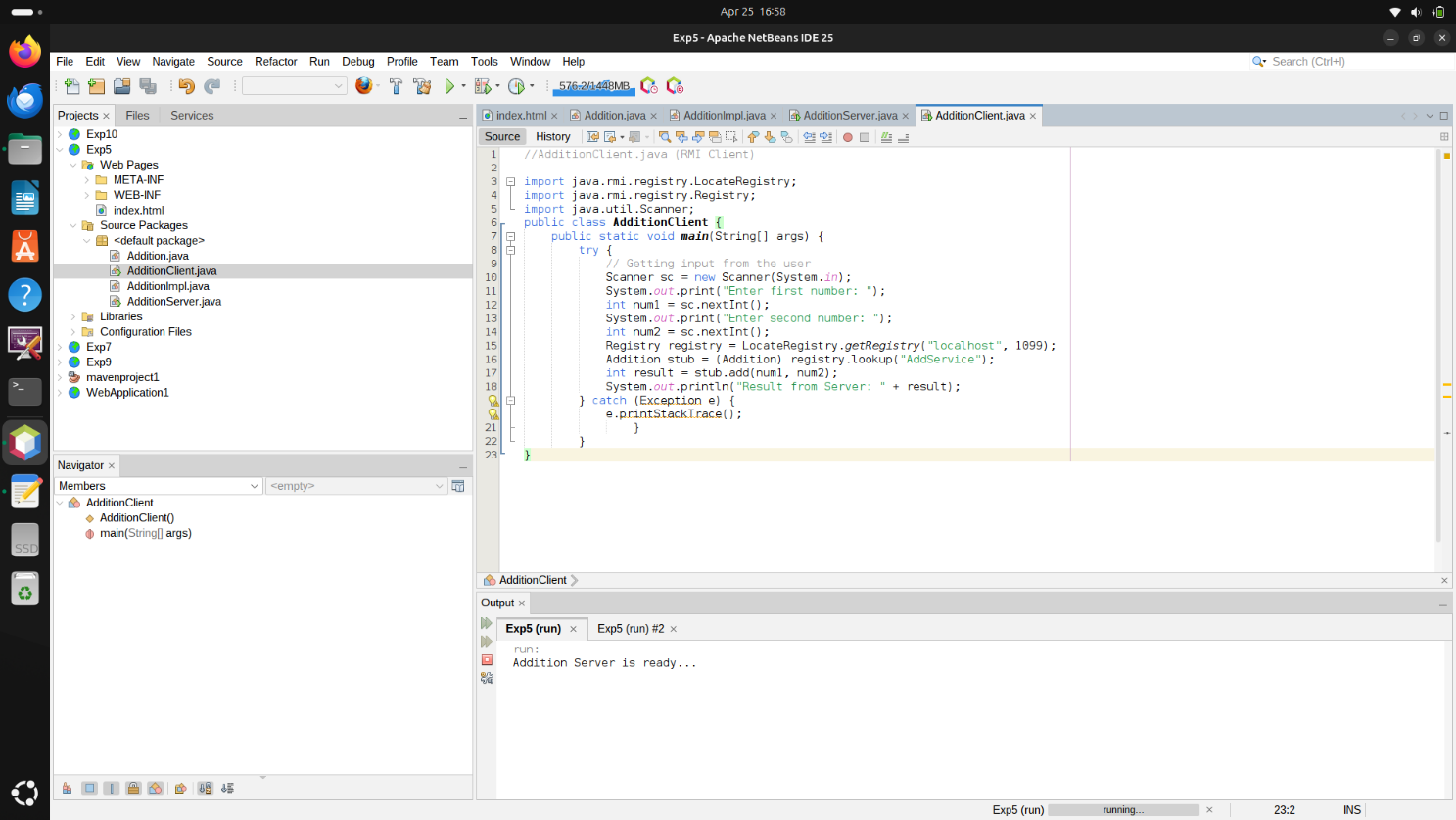
e.printStackTrace();

}

}

}

**Output :**



**Practical : 6**

**Write a program to demonstrate the use of InetAddress class and its factory methods.**

**Program :**

import java.net.InetAddress;

public class Exp6 {

public static void main(String[] args) {

try {

// 1. Get Local Host Address

InetAddress localHost = InetAddress.getLocalHost();

System.out.println("Local Host Name : " + localHost.getHostName());

System.out.println("Local Host Address : " + localHost.getHostAddress());

// 2. Get IP Address of a Website (e.g., google.com)

InetAddress google = InetAddress.getByName("www.google.com");

System.out.println("\nGoogle Host Name : " + google.getHostName());

System.out.println("Google IP Address : " + google.getHostAddress());

// 3. Get All IP Addresses Associated with the Domain

InetAddress[] addresses = InetAddress.getAllByName("www.google.com");

System.out.println("\nAll Google IP Addresses:");

for (InetAddress addr : addresses) {

System.out.println("- " + addr.getHostAddress());

}

} catch (Exception e) {

System.out.println("Error occurred: " + e.getMessage());

}

}

}

**Output :**

Local Host Name : DESKTOP-US539TC

Local Host Address : 26.253.5.171

Google Host Name : www.google.com

Google IP Address : 142.251.220.4

All Google IP Addresses:

- 142.251.220.4

- 2404:6800:4009:80a:0:0:0:2004

**Practical : 7**

**Write a servlet program to send username and password using html forms and authenticate the user.**

**Program :**

**MySrv.java**

import java.io.IOException;

import java.io.PrintWriter;

import jakarta.servlet.ServletException;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

public class MySrv extends HttpServlet {

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html");

PrintWriter out = response.getWriter();

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

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

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

out.println("<html><head><title>Login Response</title></head><body>");

if ("shubh".equals(username) && "shubham123".equals(password)) {

out.println("<h1>Welcome " + username + " to the Server!</h1>");

} else {

out.println("<h1>Login failed</h1>");

out.println("<a href='Registration.html'>Click for Home page</a>");

}

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

out.close();

}

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

doPost(request, response);

}

}

**Registration.html**

<!DOCTYPE html>

<html>

<head>

<title>Login Page</title>

</head>

<body bgcolor='#00FFFF'>

<form action='MySrv' method="post">

<center>

<h1><u>Login Page</u></h1>

<h2>

Username: <input type="text" name="uname" />

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

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

</h2>

</center>

</form>

</body>

</html>

**web.xml**

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

<web-app version="6.1" xmlns="https://jakarta.ee/xml/ns/jakartaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee https://jakarta.ee/xml/ns/jakartaee/web-app\_6\_1.xsd">

<servlet>

<servlet-name>MySrv</servlet-name>

<servlet-class>MySrv</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>MySrv</servlet-name>

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

</servlet-mapping>

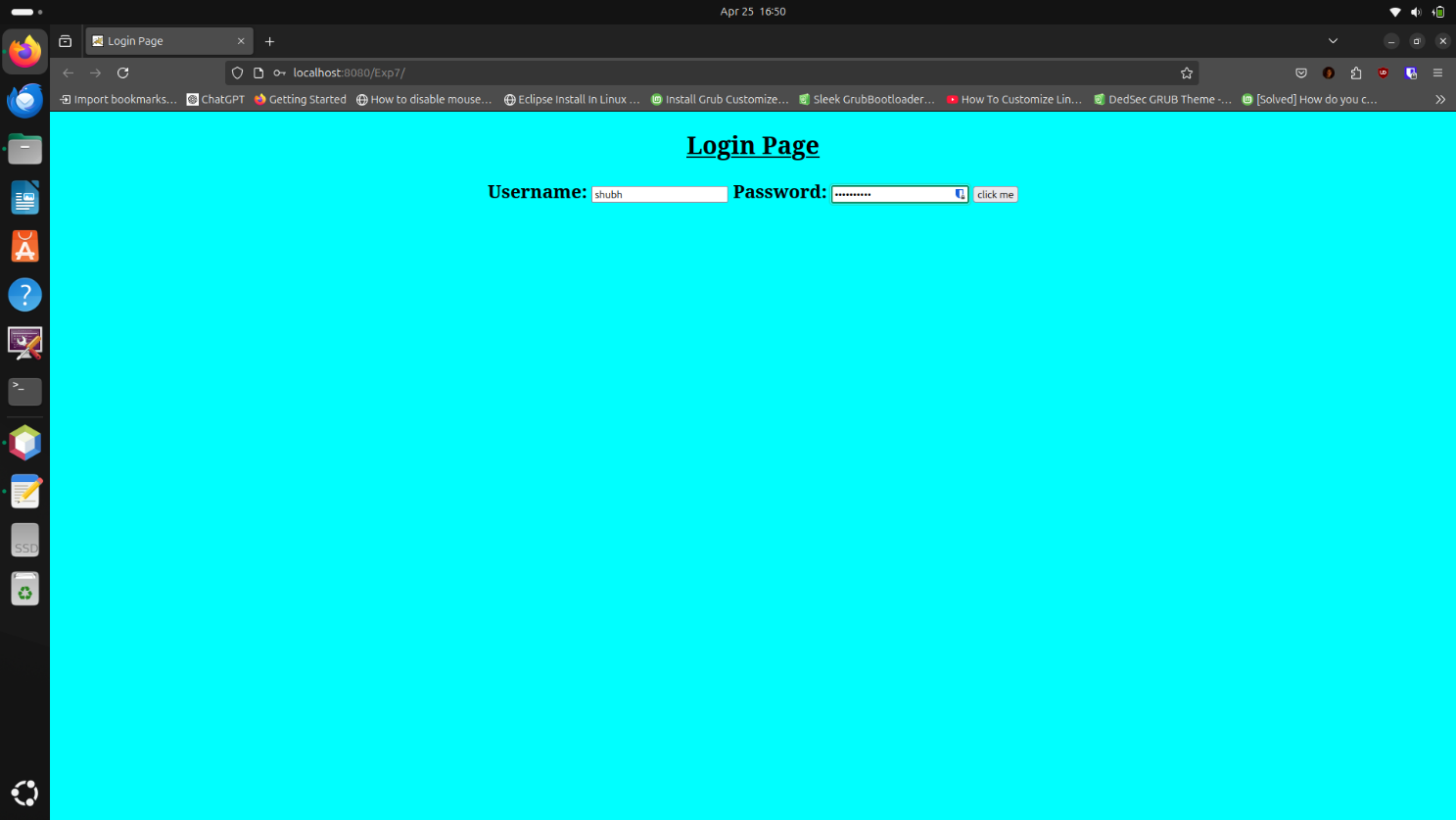
<welcome-file-list>

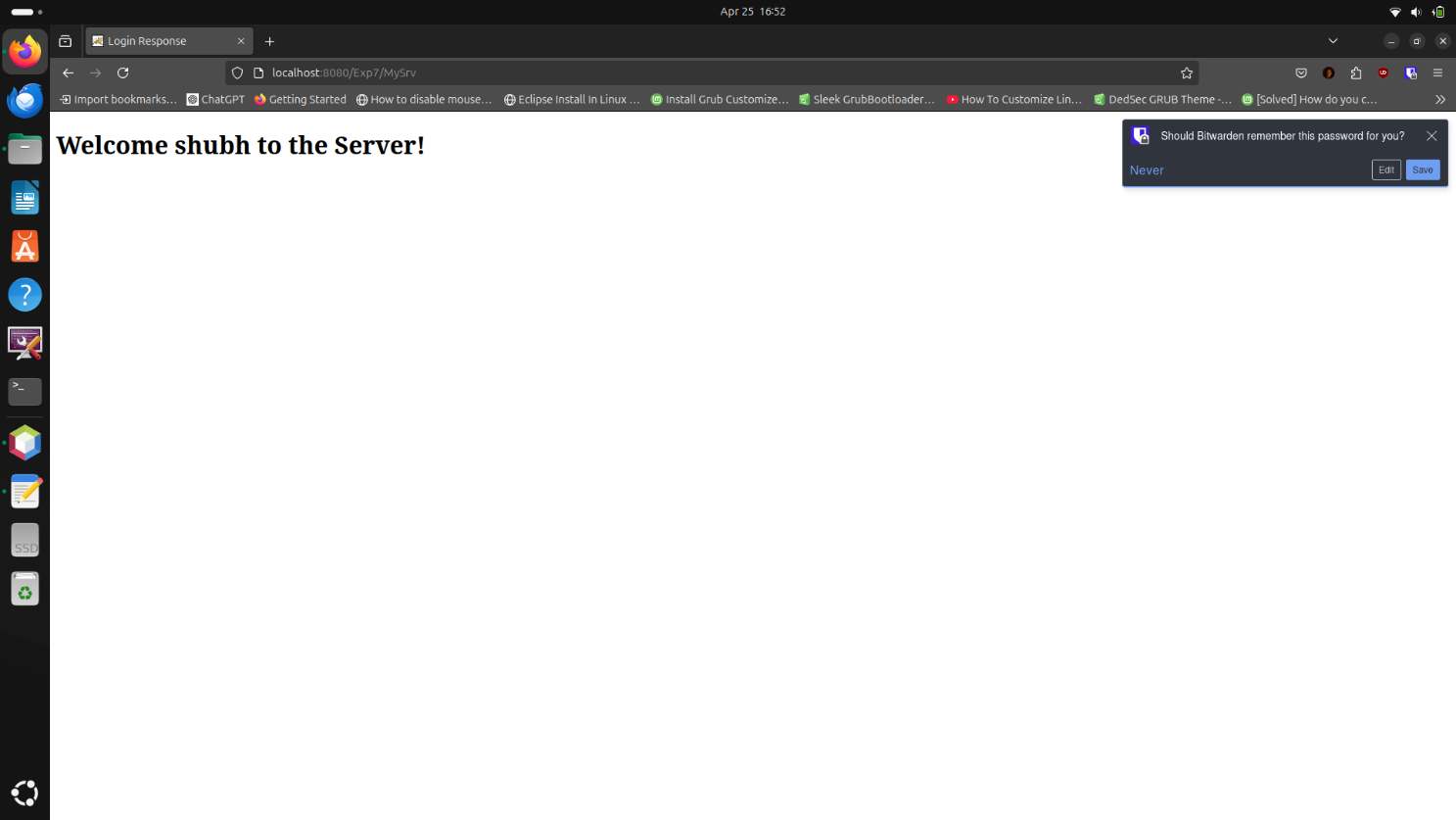
<welcome-file>Registration.html</welcome-file>

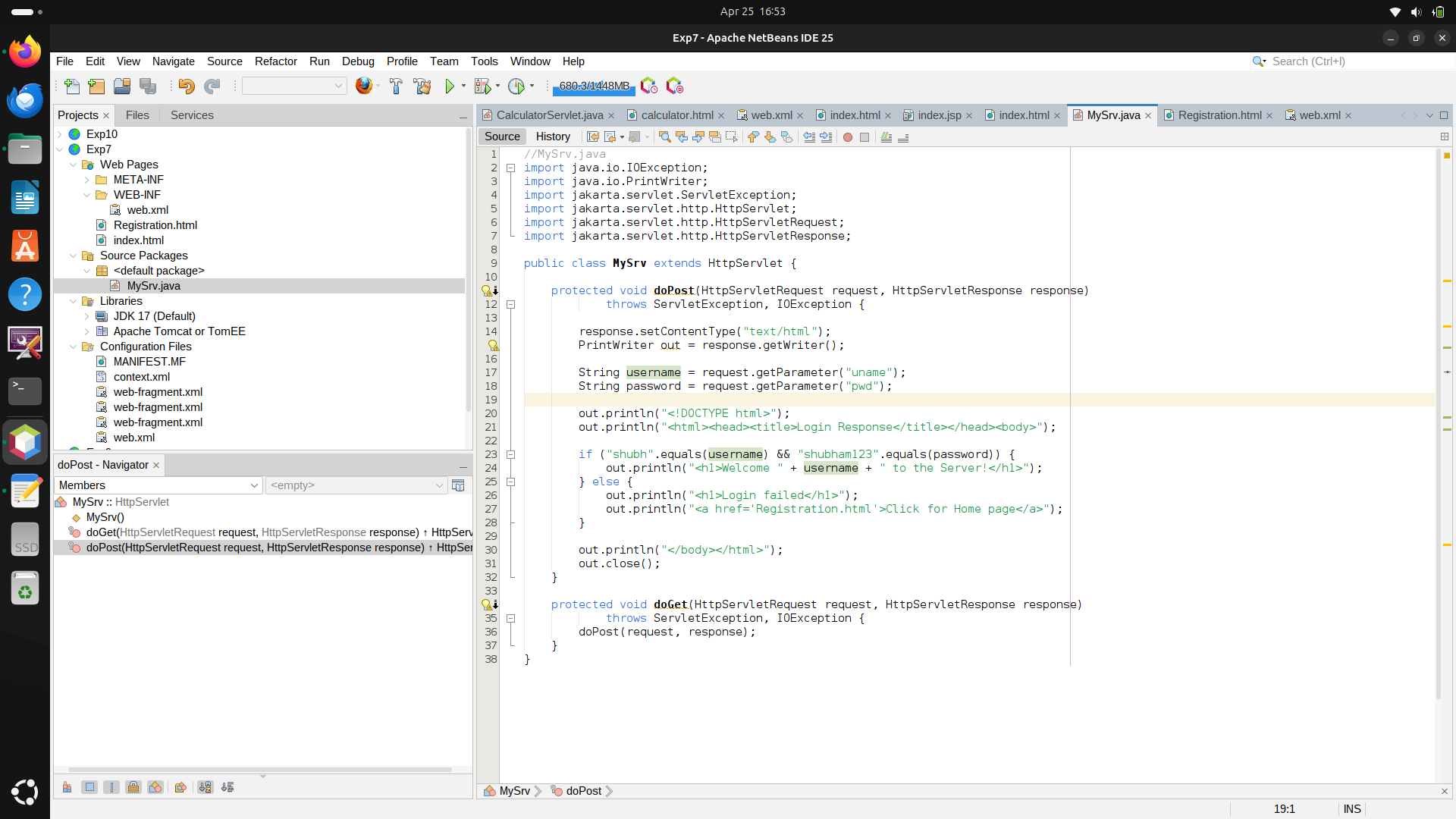
</welcome-file-list>

</web-app>

**Output :**







**Practical : 8**

**Write a database application that uses any JDBC driver**

**Program :**

**StudentDB.java –**

import java.sql.\*;

public class JdbcExample {

public static void main(String[] args) {

// Step 1: Define database parameters

String url = "jdbc:mysql://localhost:3306/Exp8"; // Replace 'testdb' with your database name

String user = "root"; // Replace with your DB username

String password = "root123"; // Replace with your DB password

try {

// Step 2: Load and Register the JDBC driver

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

// Step 3: Establish the connection

Connection con = DriverManager.getConnection(url, user, password);

System.out.println("Connection established successfully.");

// Step 4: Create a Statement

Statement st = con.createStatement();

// Step 5: Execute a query (creating a table, inserting data, retrieving)

String createTable = "CREATE TABLE IF NOT EXISTS students (id INT, name VARCHAR(50))";

st.executeUpdate(createTable);

String insertData = "INSERT INTO students (id, name) VALUES (1, 'Alice'), (2, 'Bob')";

st.executeUpdate(insertData);

String selectQuery = "SELECT \* FROM students";

ResultSet rs = st.executeQuery(selectQuery);

// Step 6: Process the results

System.out.println("Student Data:");

while (rs.next()) {

System.out.println("ID: " + rs.getInt("id") + ", Name: " + rs.getString("name"));

}

// Step 7: Close the connection

con.close();

System.out.println("Connection closed.");

} catch (Exception e) {

e.printStackTrace();

}

}

}

**MySQL :**

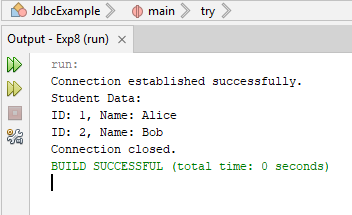
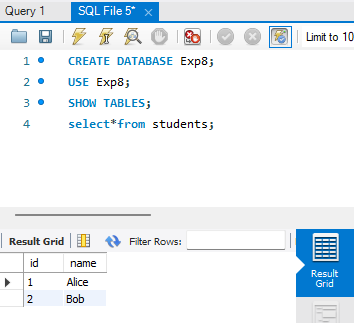
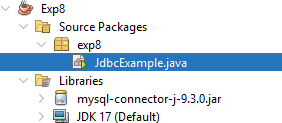
CREATE DATABASE Exp8;

USE Exp8;

SHOW TABLES;

select\*from students;

**Output :**

**Practical : 9**

**Write a servlet program to implement simple calculator.**

**Program :**

**CalculatorServlet.java**

import java.io.\*;

import jakarta.servlet.\*;

import jakarta.servlet.http.\*;

public class CalculatorServlet extends HttpServlet {

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

response.setContentType("text/html");

PrintWriter out = response.getWriter();

// Get parameters from form

int num1 = Integer.parseInt(request.getParameter("num1"));

int num2 = Integer.parseInt(request.getParameter("num2"));

String op = request.getParameter("operation");

double result = 0;

switch (op) {

case "add": result = num1 + num2; break;

case "sub": result = num1 - num2; break;

case "mul": result = num1 \* num2; break;

case "div":

if (num2 != 0)

result = (double) num1 / num2;

else

out.println("<h3>Division by zero error!</h3>");

break;

default:

out.println("<h3>Invalid Operation</h3>");

return;

}

out.println("<h2>Result: " + result + "</h2>");

}}

**Calculator.html**

<!DOCTYPE html>

<html>

<head><title>Simple Calculator</title></head>

<body>

<h2>Simple Calculator</h2>

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

Number 1: <input type="text" name="num1"><br><br>

Number 2: <input type="text" name="num2"><br><br>

Operation:<br>

<input type="radio" name="operation" value="add" checked> Addition<br>

<input type="radio" name="operation" value="sub"> Subtraction<br>

<input type="radio" name="operation" value="mul"> Multiplication<br>

<input type="radio" name="operation" value="div"> Division<br><br>

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

</form>

</body>

</html>

**web.xml**

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

<web-app 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-app\_3\_0.xsd" version="3.0">

<display-name>SimpleCalculatorApp</display-name>

<servlet>

<servlet-name>CalculatorServlet</servlet-name>

<servlet-class>CalculatorServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>CalculatorServlet</servlet-name>

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

</servlet-mapping>

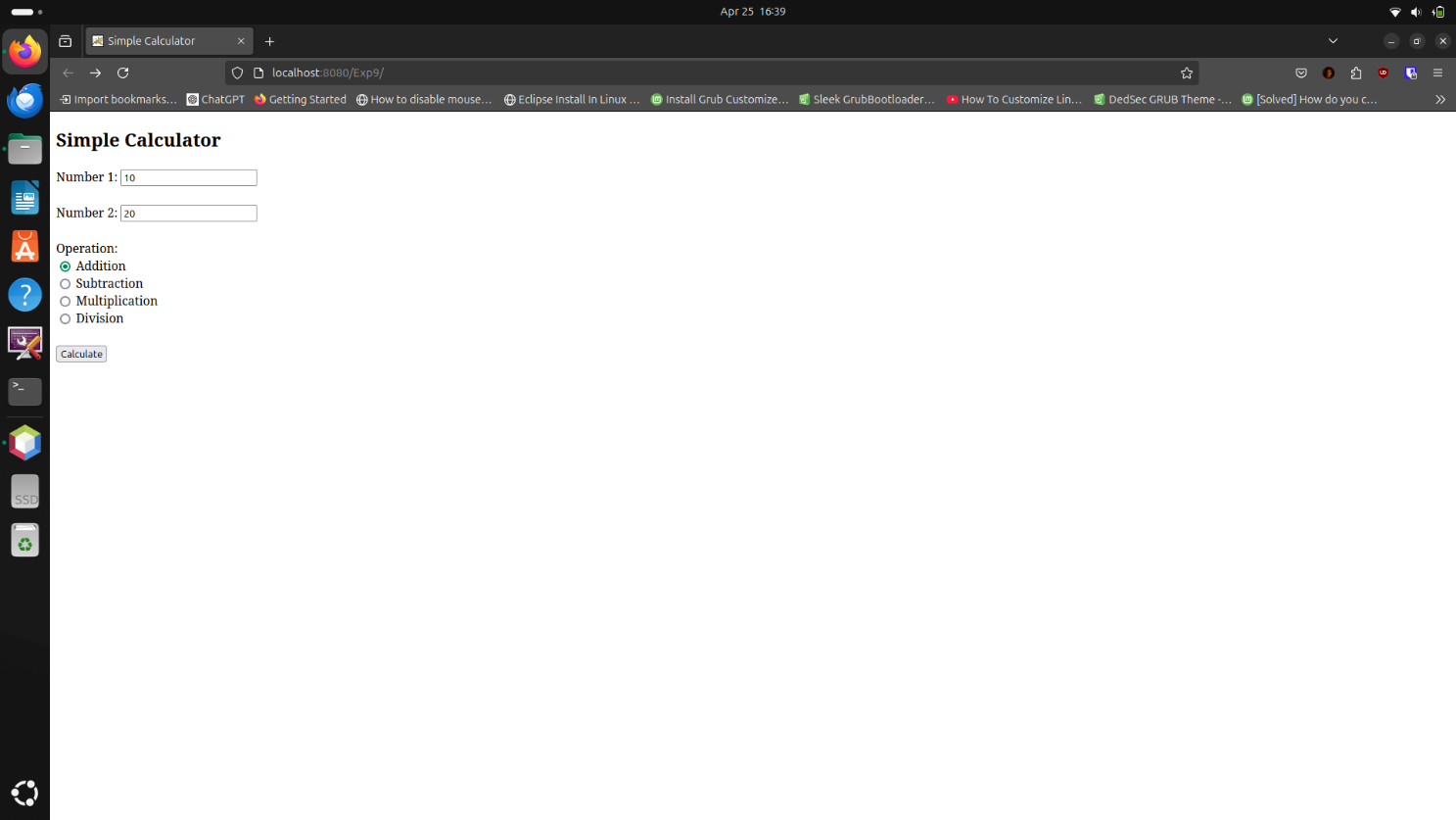
<welcome-file-list>

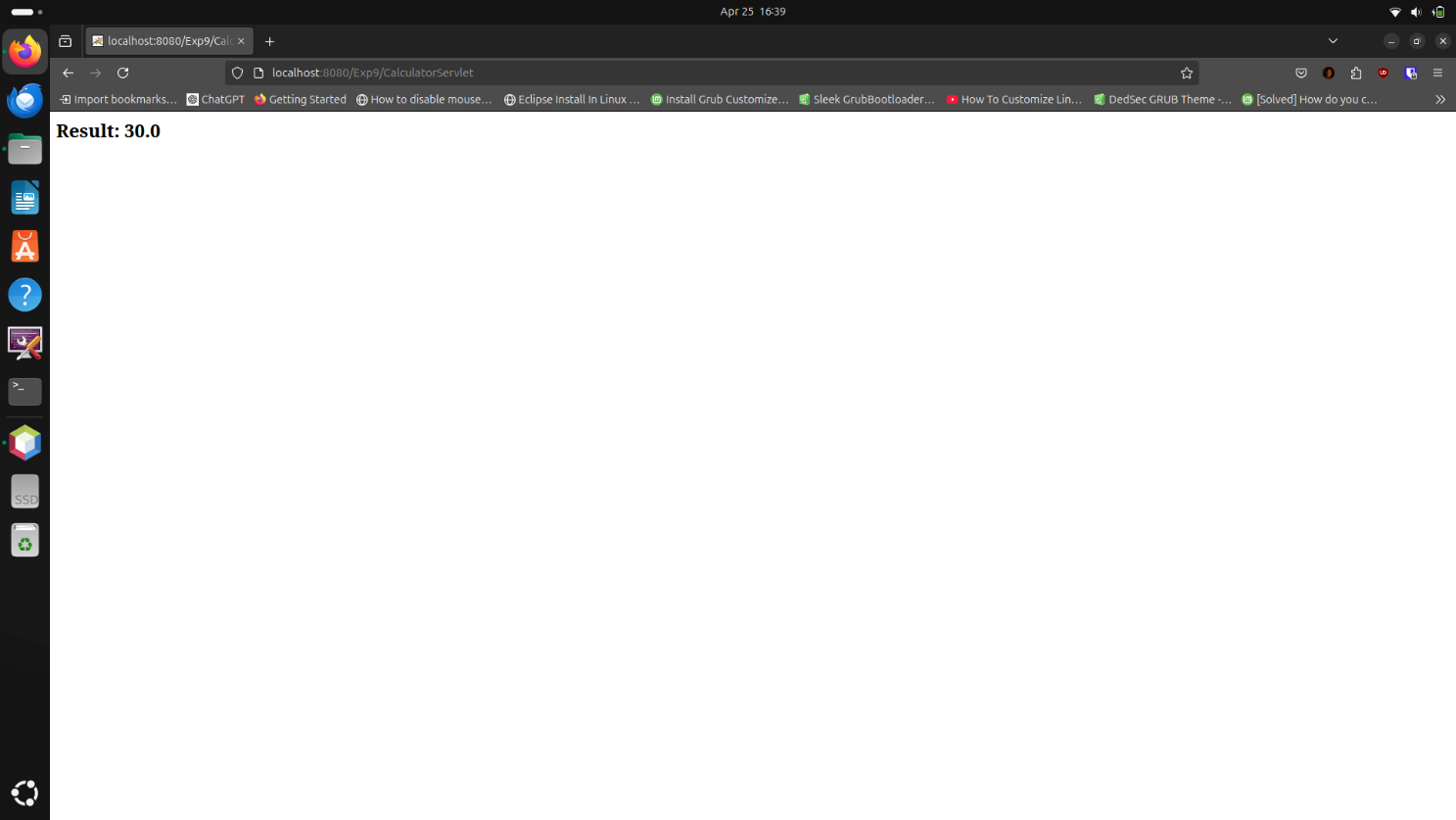
<welcome-file>calculator.html</welcome-file>

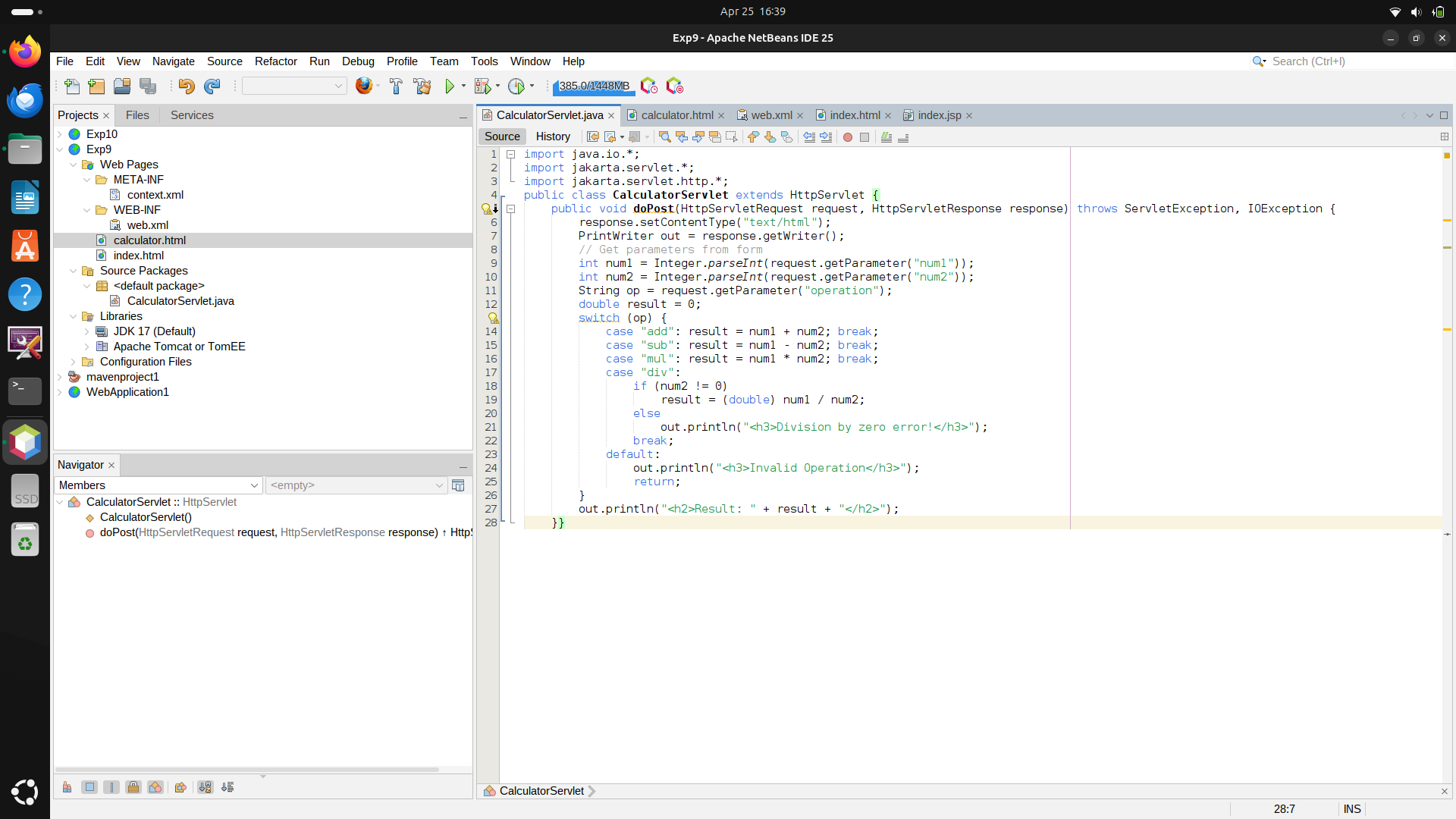
</welcome-file-list>

</web-app>

**Output :**







**Practical : 10**

**Write a simple JSP page to display a simple message (It may be a simple HTML page)**

**Program :**

**index.jsp**

<!DOCTYPE html>

<html>

<head>

<title>Input Page</title>

</head>

<body style="text-align: center; padding-top: 50px;">

<h2>Enter Your Name</h2>

<form action="#">

<input type="text" name="username" placeholder="Your Name" required>

<br><br>

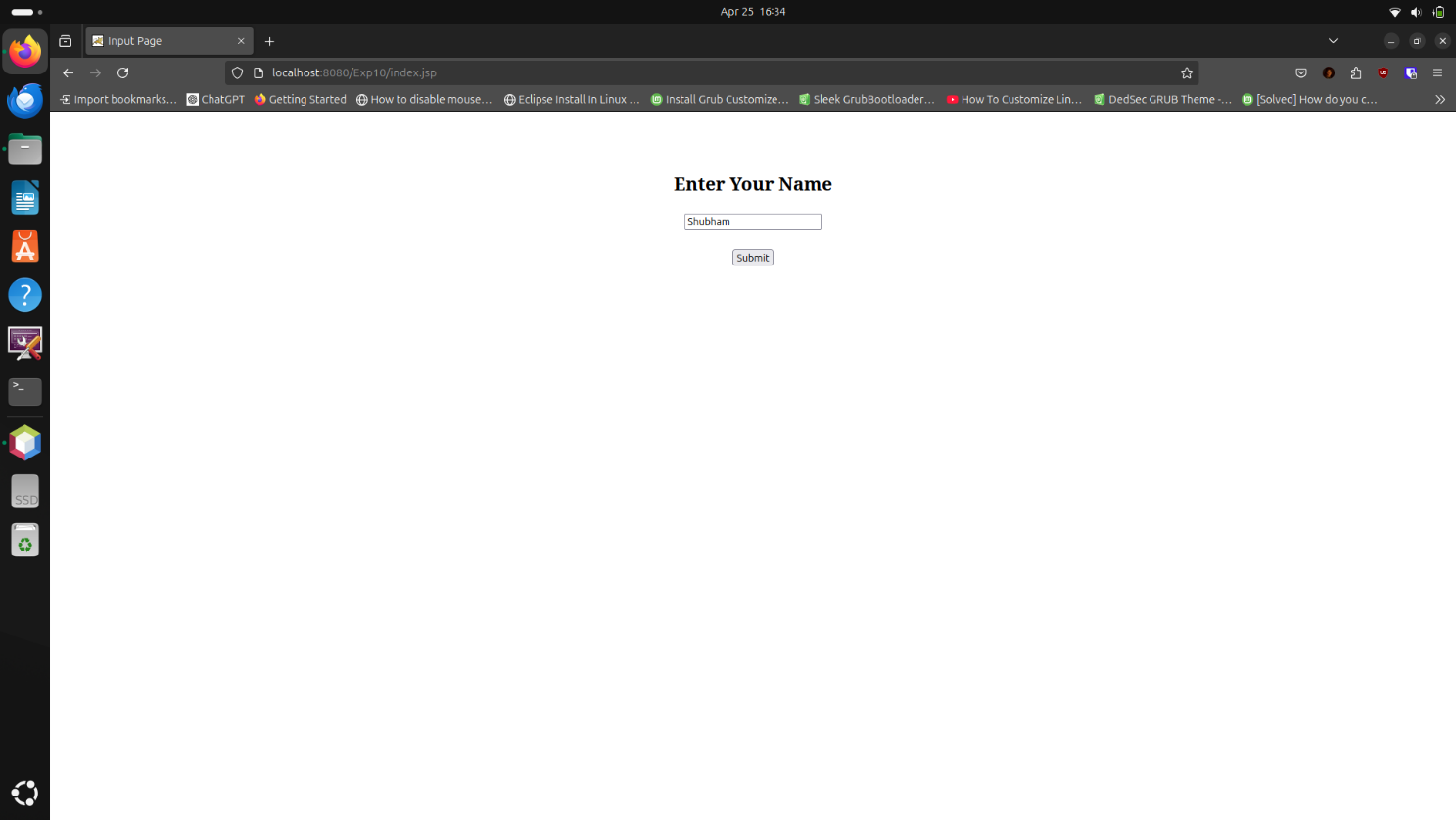
<input type="submit" value="Submit">

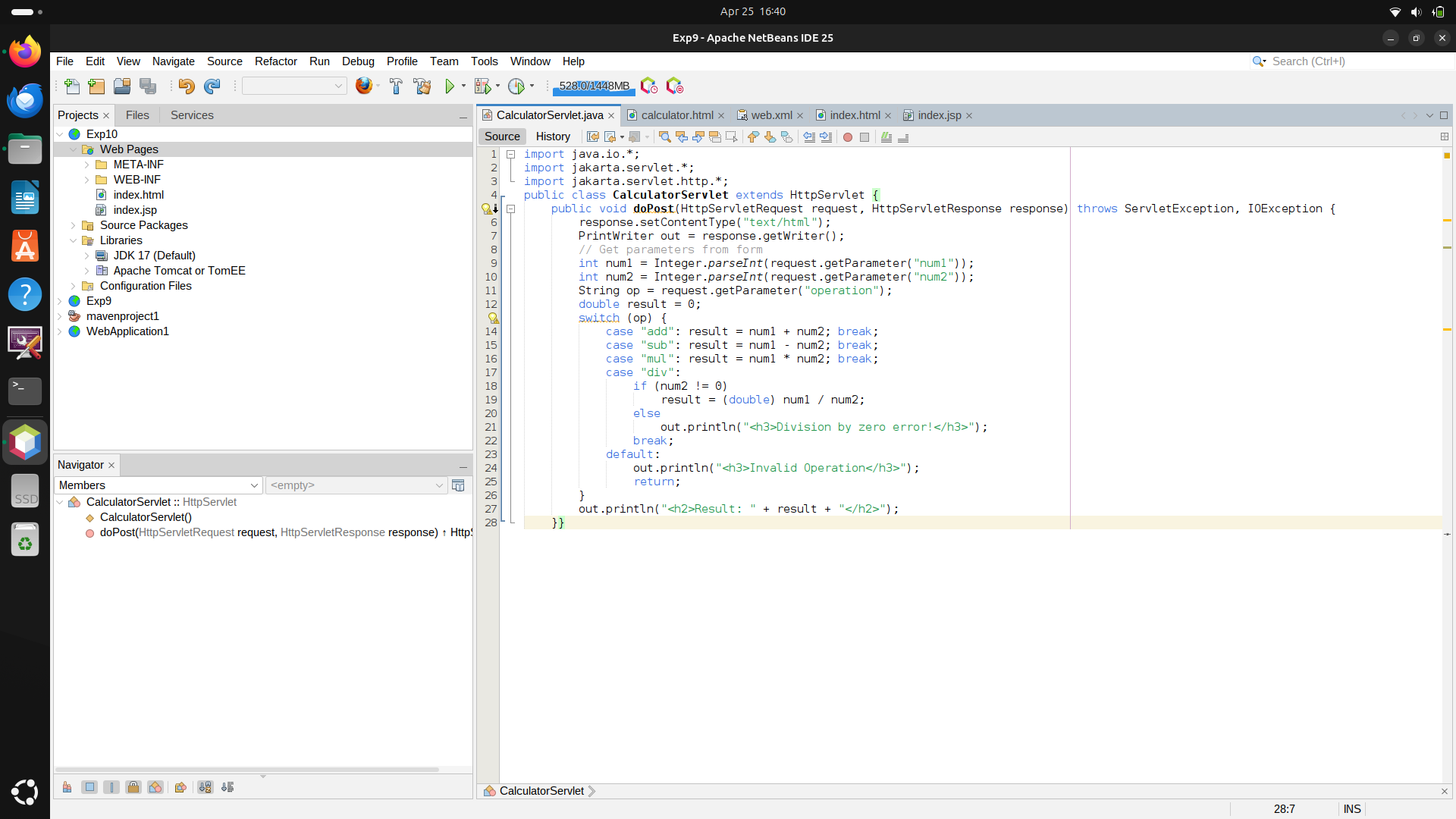
</form>

</body>

</html>

**Output :**





**Practical : 11**

**Write a program to demonstrate different layout.**

**Program :**

import java.awt.\*;

public class LayoutDemo extends Frame {

public LayoutDemo() {

// Panel with BorderLayout

Panel borderPanel = new Panel(new BorderLayout());

borderPanel.add(new Button("North"), BorderLayout.NORTH);

borderPanel.add(new Button("West"), BorderLayout.WEST);

borderPanel.add(new Button("Center"), BorderLayout.CENTER);

// Panel with FlowLayout

Panel flowPanel = new Panel(new FlowLayout());

for (int i = 1; i <= 5; i++)

flowPanel.add(new Button("Button " + i));

// Panel with GridLayout

Panel gridPanel = new Panel(new GridLayout(2, 3));

for (int i = 1; i <= 6; i++)

gridPanel.add(new Button("Grid " + i));

// Frame layout

setLayout(new GridLayout(3, 1));

add(borderPanel);

add(flowPanel);

add(gridPanel);

setSize(500, 400);

setVisible(true);

}

public static void main(String[] args) {

new LayoutDemo();

}

}

**Output :**



**Practical : 12**

**Write a program using AWT to create a menu bar where menu bar contains menu items such as file, edit, view and create a submenu under the file menu: new and open**

**Program :**

import java.awt.\*;

public class MenuDemo extends Frame {

public MenuDemo() {

MenuBar mb = new MenuBar();

Menu file = new Menu("File");

file.add(new MenuItem("New"));

file.add(new MenuItem("Open"));

mb.add(file);

mb.add(new Menu("Edit"));

mb.add(new Menu("View"));

setMenuBar(mb);

setSize(300, 200);

setVisible(true);

}

public static void main(String[] args) {

new MenuDemo();

}

}

//

import java.awt.\*;

import java.awt.event.ActionListener;

import java.awt.event.ItemEvent;

import java.awt.event.ItemListener;

import java.awt.event.ActionEvent;

public class Exp12 extends Frame implements ActionListener, ItemListener {

Dialog dialog;

Label l;

Exp12() {

MenuBar mBar = new MenuBar();

setMenuBar(mBar);

Menu file = new Menu("File");

MenuItem new\_file = new MenuItem("New");

MenuItem open\_file = new MenuItem("Open");

MenuItem save\_file = new MenuItem("Save");

new\_file.addActionListener(this);

open\_file.addActionListener(this);

save\_file.addActionListener(this);

file.add(new\_file);

file.add(open\_file);

file.add(save\_file);

mBar.add(file);

Menu edit = new Menu("Edit");

MenuItem undo\_edit = new MenuItem("Undo");

CheckboxMenuItem cut\_edit = new CheckboxMenuItem("Cut");

CheckboxMenuItem copy\_edit = new CheckboxMenuItem("Copy");

CheckboxMenuItem paste\_edit = new CheckboxMenuItem("Paste");

undo\_edit.addActionListener(this);

cut\_edit.addItemListener(this);

copy\_edit.addItemListener(this);

paste\_edit.addItemListener(this);

Menu sub = new Menu("Save Type");

MenuItem sub1\_sum = new MenuItem("Direct Save");

MenuItem sub2\_sum = new MenuItem("Save As");

sub.add(sub1\_sum);

sub.add(sub2\_sum);

edit.add(sub);

edit.add(undo\_edit);

edit.add(cut\_edit);

edit.add(copy\_edit);

edit.add(paste\_edit);

mBar.add(edit);

dialog = new Dialog(this, false);

dialog.setSize(200, 200);

dialog.setTitle("Dialog Box");

Button b = new Button("Close");

b.addActionListener(this);

dialog.setLayout(new FlowLayout());

dialog.add(b);

l = new Label();

dialog.add(l);

}

public void actionPerformed(ActionEvent ae) {

String selected\_item = ae.getActionCommand();

switch (selected\_item) {

case "New": l.setText("New"); break;

case "Open": l.setText("Open"); break;

case "Save": l.setText("Save"); break;

case "Undo": l.setText("Undo"); break;

case "Cut": l.setText("Cut"); break;

case "Copy": l.setText("Copy"); break;

case "Paste": l.setText("Paste"); break;

case "Close": dialog.dispose(); return;

default: l.setText("Invalid Input");

}

dialog.setVisible(true);

}

public void itemStateChanged(ItemEvent ie) {

this.repaint();

}

public static void main(String[] args) {

Exp12 md = new Exp12();

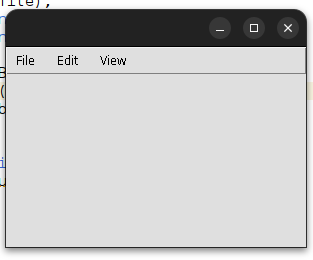
md.setVisible(true);

md.setSize(400, 400);

}

}

**Output :**



**How to run these all programs**

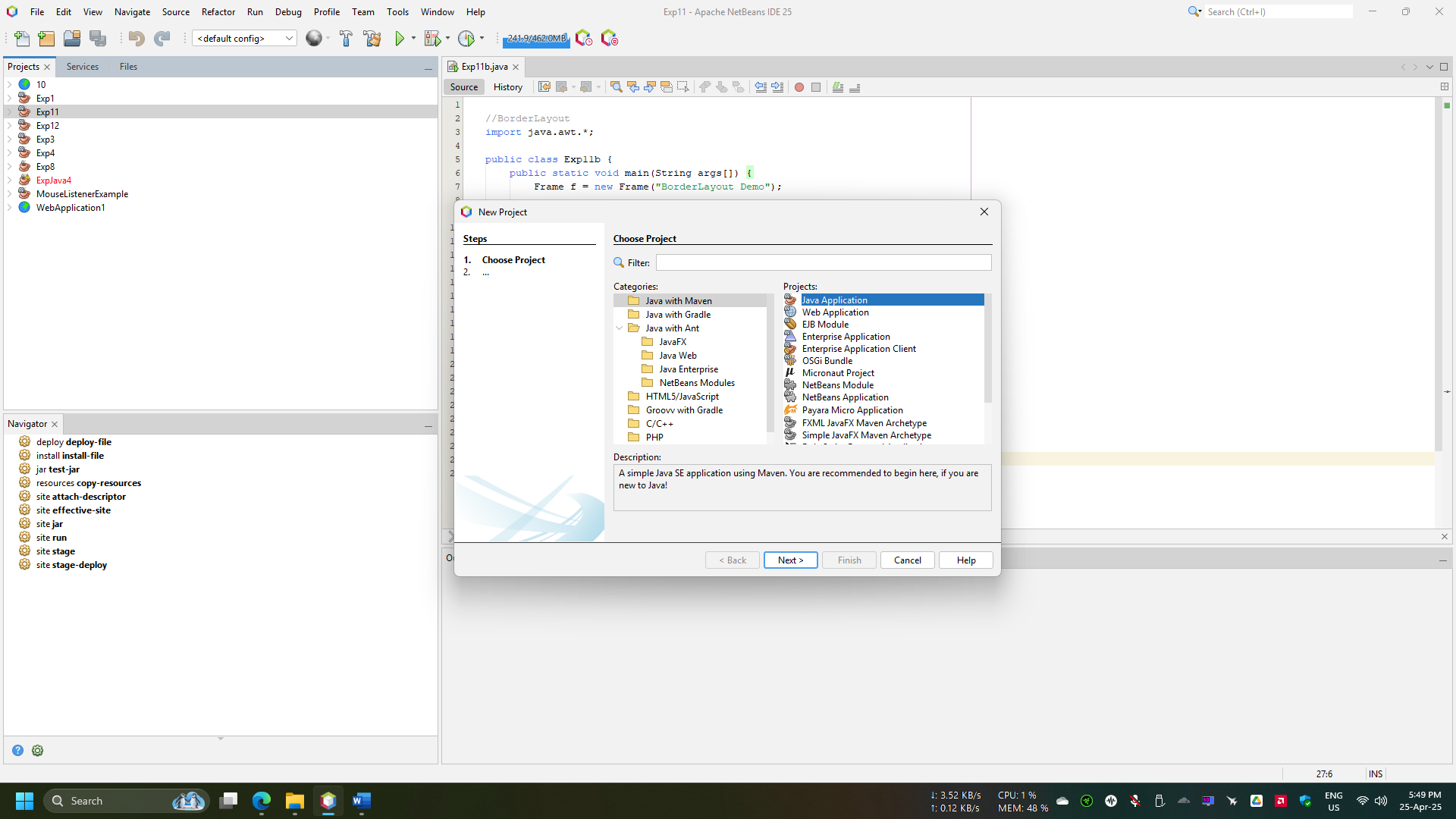
**For applet(1st ) program**

**Change the Java version to JDK 8 and run these commands in the program folder –**

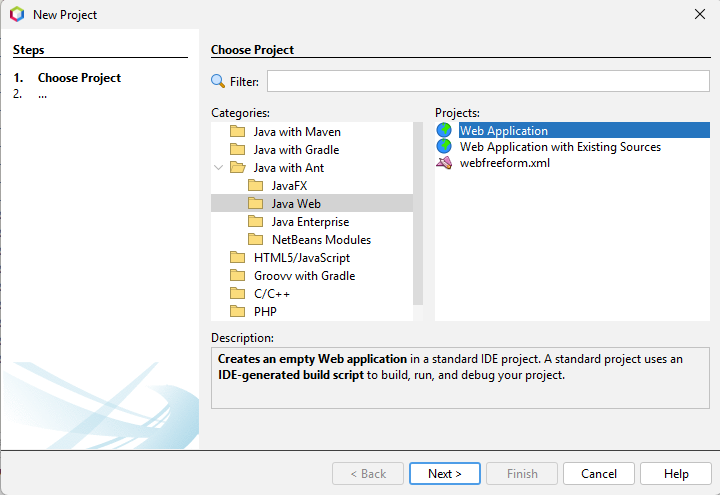
javac 1stprogram.java

appletviewer 1stprogram.html

**For Simple Java Programs**



**For JSP Servlet and other Server side or to open in Web browser programs**

****