**Experiment-1**

**Code:**

**KeyEventDemo.java**

import java.awt.\*;

import java.applet.\*;

import java.awt.event.\*;

public class KeyEventDemo extends Applet implements KeyListener {

String msg = "Assign1: A JAVA program to demo status of key";

public void init() {

addKeyListener(this);

setFocusable(true); // Important to ensure key events are received

}

public void keyPressed(KeyEvent k) {

showStatus("Key Pressed");

repaint();

}

public void keyReleased(KeyEvent k) {

showStatus("Key Released");

repaint();

}

public void keyTyped(KeyEvent k) {

showStatus("Key Typed");

repaint();

}

public void paint(Graphics g) {

g.drawString(msg, 10, 20);

}

}

**Index.html**

<html>

<body>

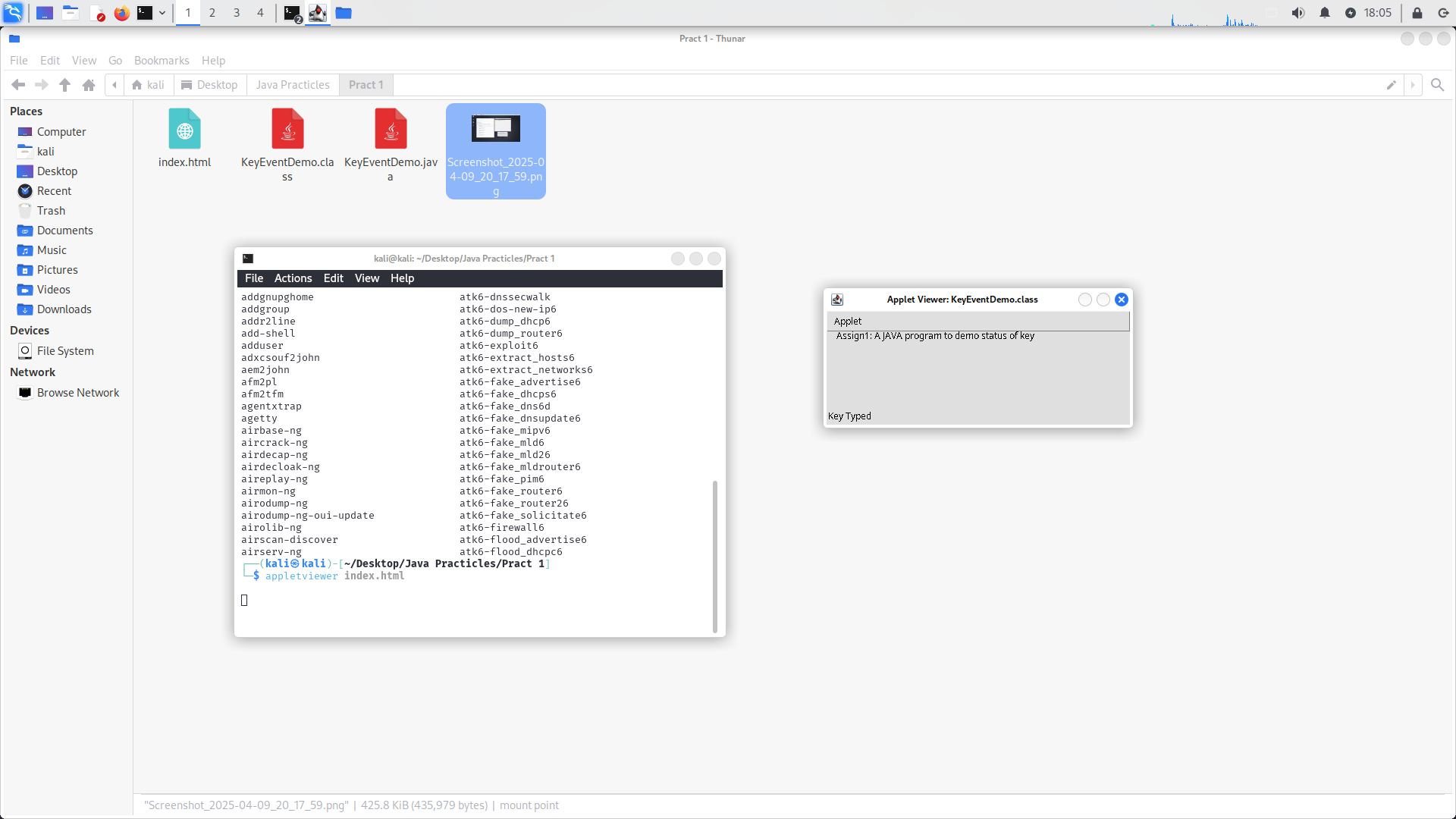
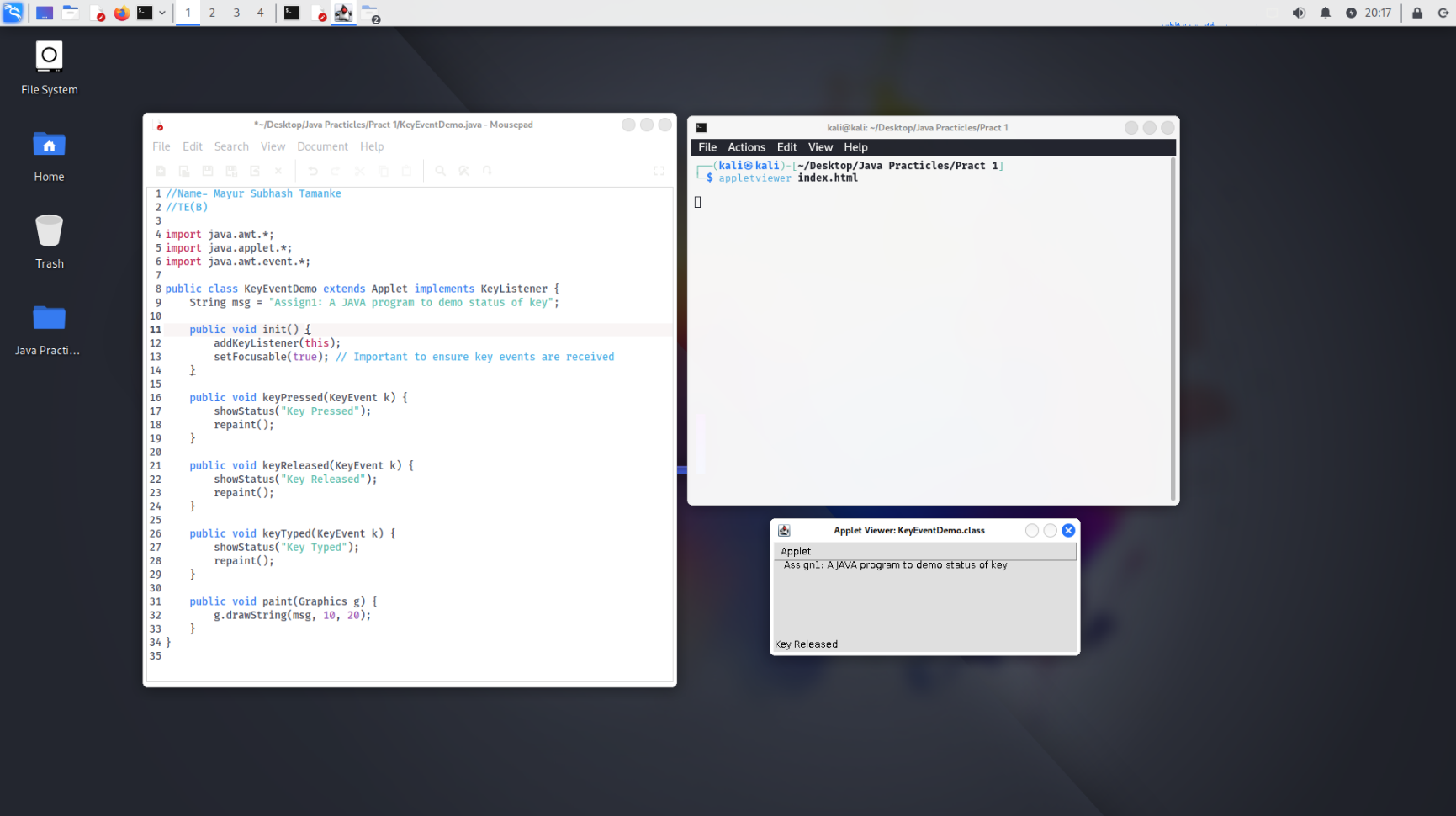
<applet code="KeyEventDemo.class" width="400" height="100">

</applet>

</body>

</html>

**Output:**



**Experiment-2**

**Code:**

**Exp2.java**

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) {

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);

Exp2 mouseFrame = new Exp2();

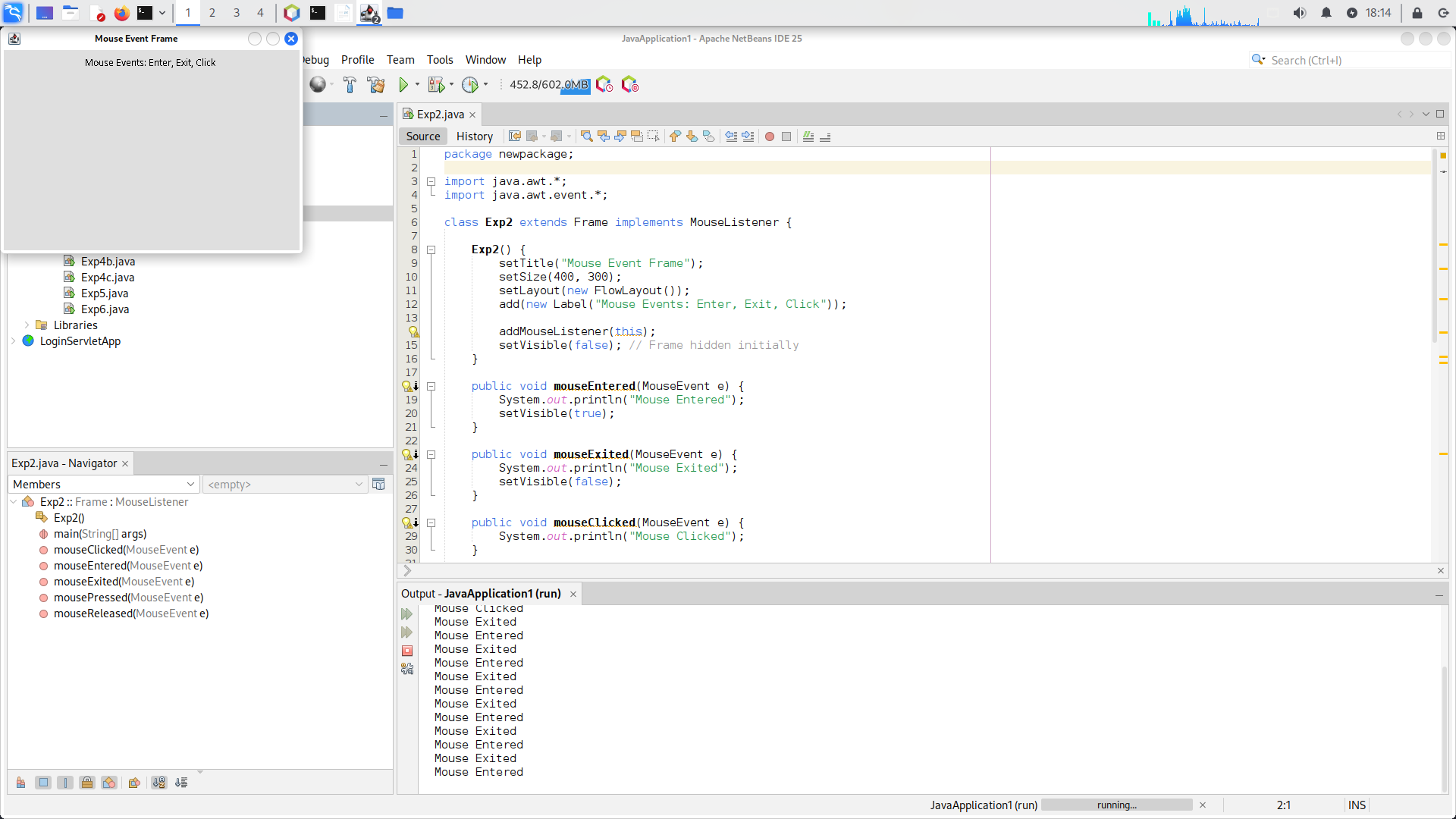
baseFrame.addMouseListener(new MouseAdapter() {

public void mouseEntered(MouseEvent e) {

mouseFrame.setVisible(true);

} }); } }

**Output:**



**Experiment-3**

**Code:**

**Exp3.java**

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(7, 2, 10, 10));

add(new JLabel("Enter marks for subjects:", SwingConstants.CENTER));

add(new JLabel()); // spacer

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);

setVisible(true);

}

void showResult() {

int total = 0;

boolean pass = true;

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

String input = markFields[i].getText().trim();

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

JOptionPane.showMessageDialog(this, "Please 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,

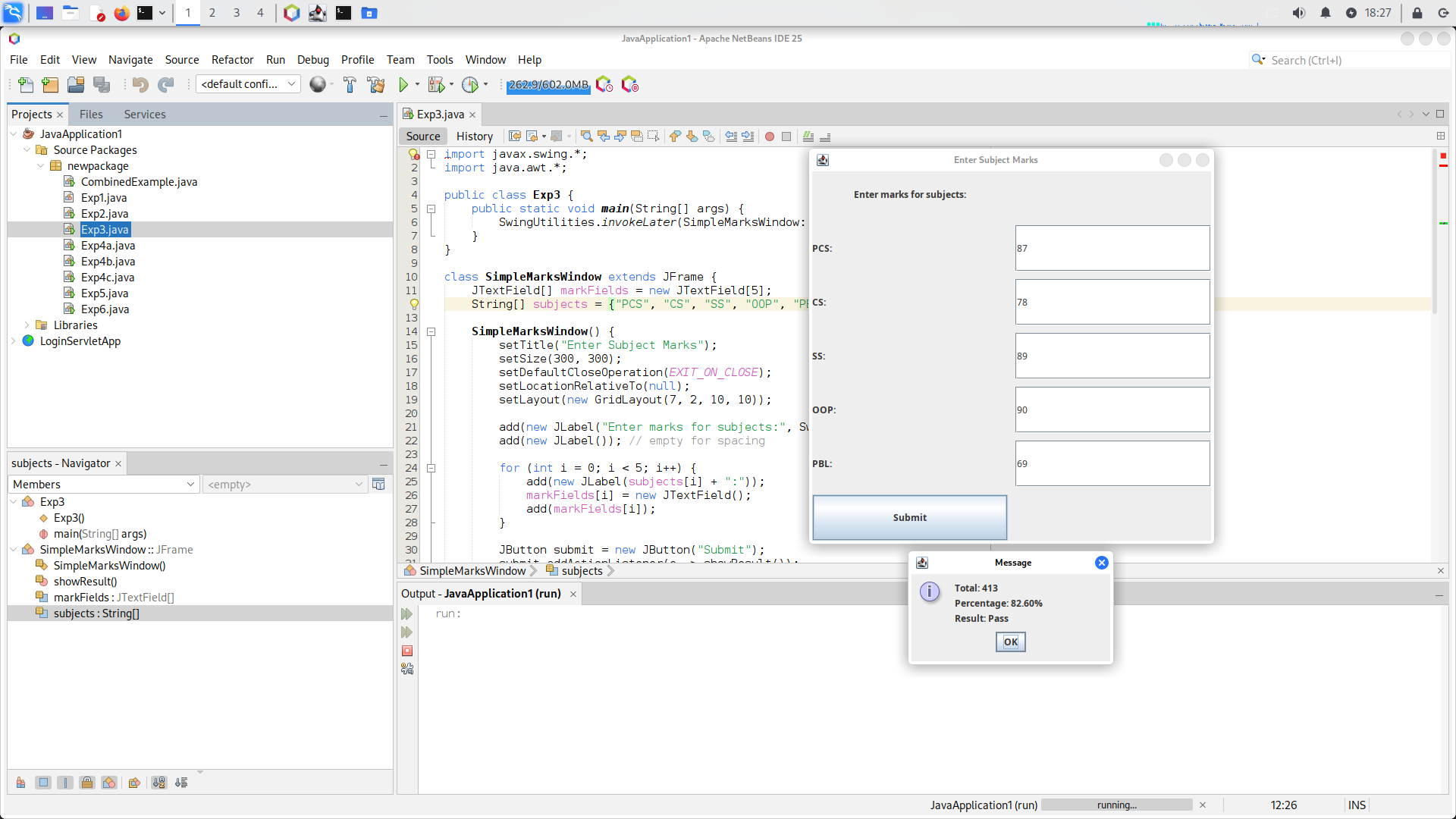
"Total: " + total +

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

"\nResult: " + result);

} }

**Output:**



**Experiment-4**

**Code:**

**Exp4.java**

import java.sql.\*;

public class Exp4 {

public static void main(String[] args) {

String url = "jdbc:mysql://localhost:3306/testdb"; // MySQL JDBC URL

String user = "root";

String password = "mayur123";

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

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

String insert = "INSERT INTO students (id, name, age) VALUES (?, ?, ?)";

try (PreparedStatement ps = con.prepareStatement(insert)) {

ps.setInt(1, 3);

ps.setString(2, "Ram");

ps.setInt(3, 18);

ps.executeUpdate();

}

try (Statement stmt = con.createStatement();

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

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

while (rs.next()) {

System.out.printf("%d | %s | %d%n",

rs.getInt("id"),

rs.getString("name"),

rs.getInt("age"));

}

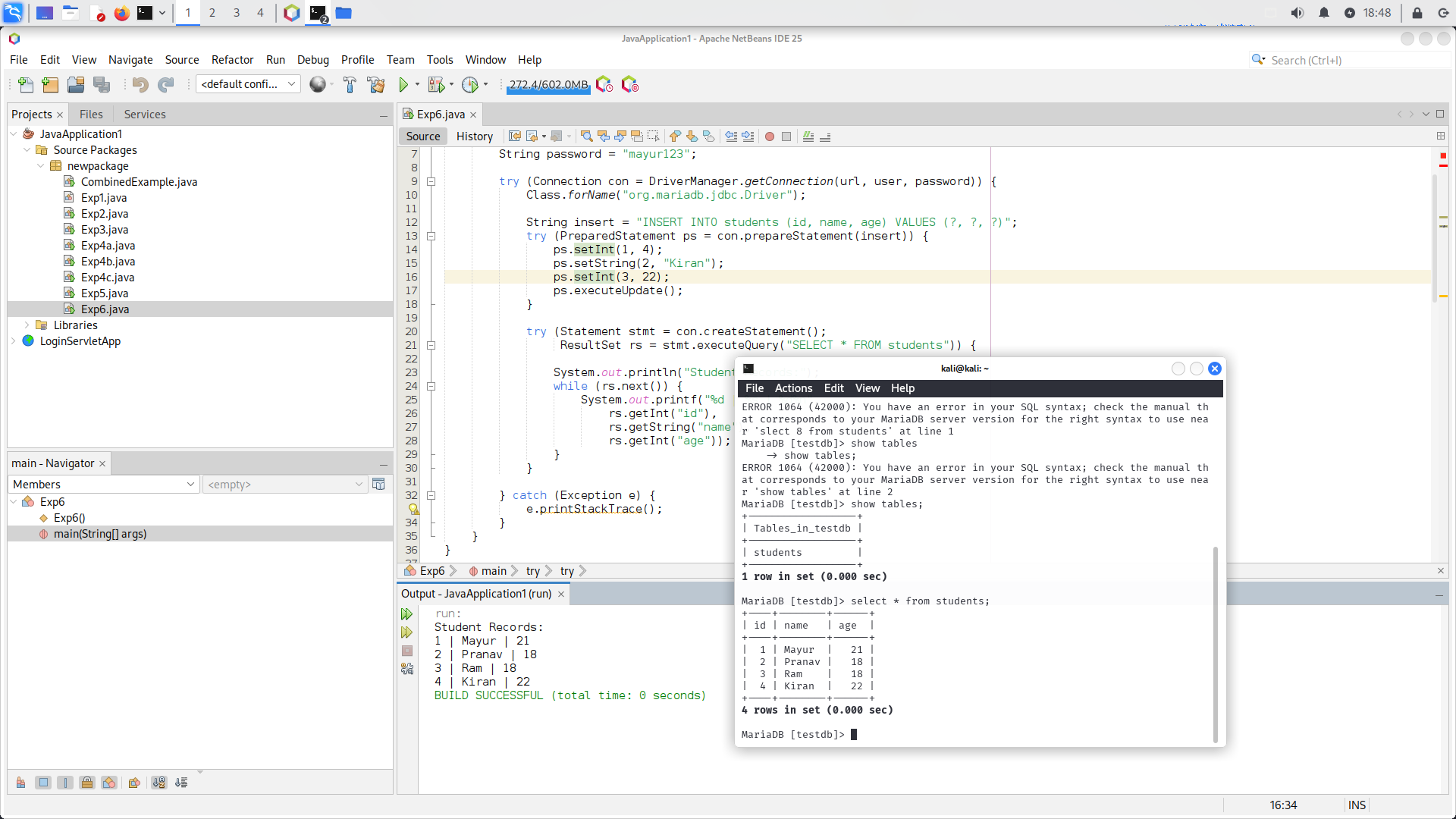
}

} catch (Exception e) {

e.printStackTrace();

}}}

**Output:**



**Experiment-5**

**Code:**

**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) {

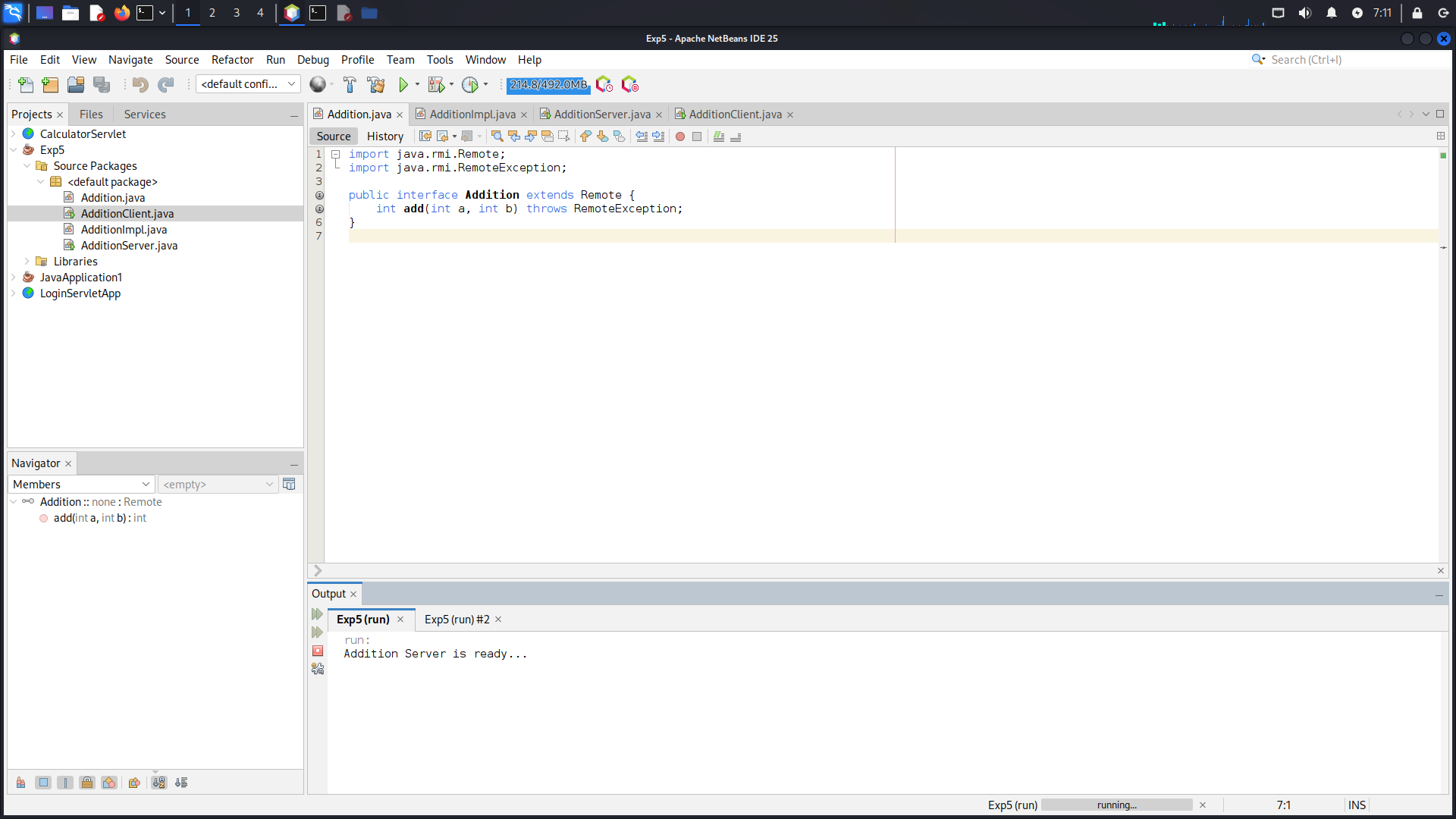
e.printStackTrace();

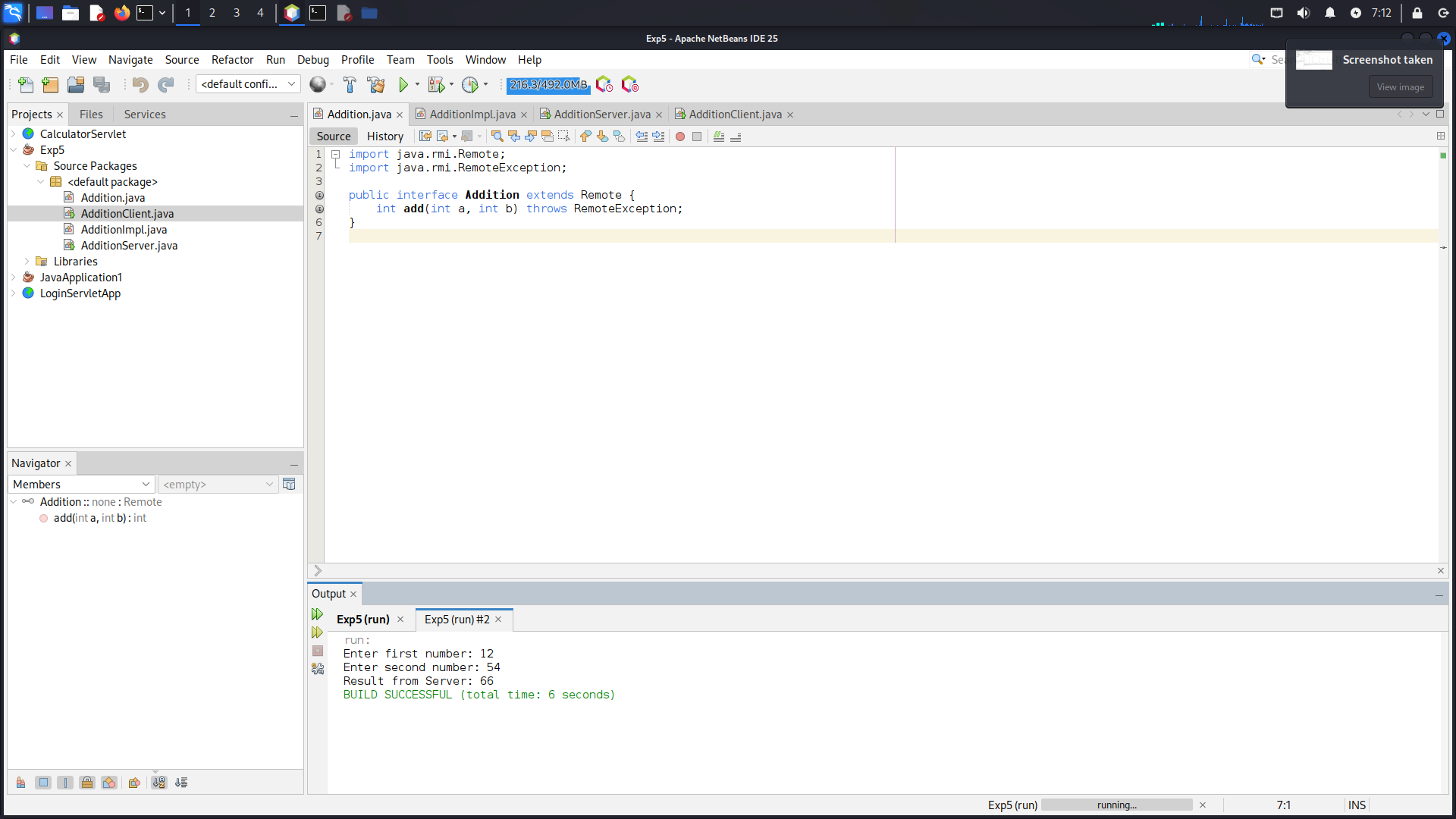
}

}

}

**Output:**





**Experiment-6**

**Code:**

**Exp6.java**

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) {

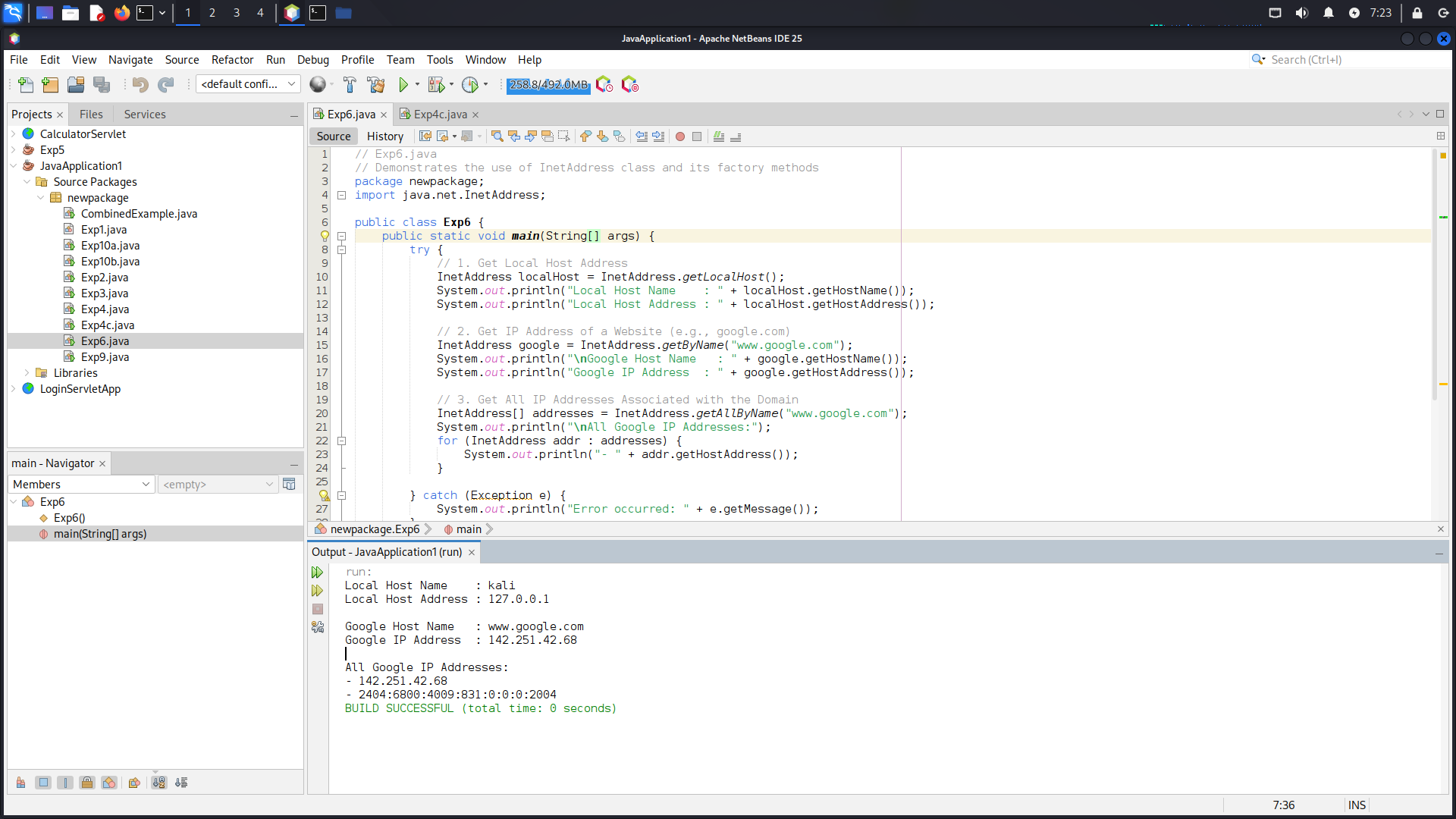
System.out.println("Error occurred: " + e.getMessage());

}

}

}

**Output:**



**Experiment-7**

**Code:**

**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 ("SITS".equals(username) && "SITS".equals(password)) {

out.println("<h1>Welcome to " + username + "</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='#e600e6'>

<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>

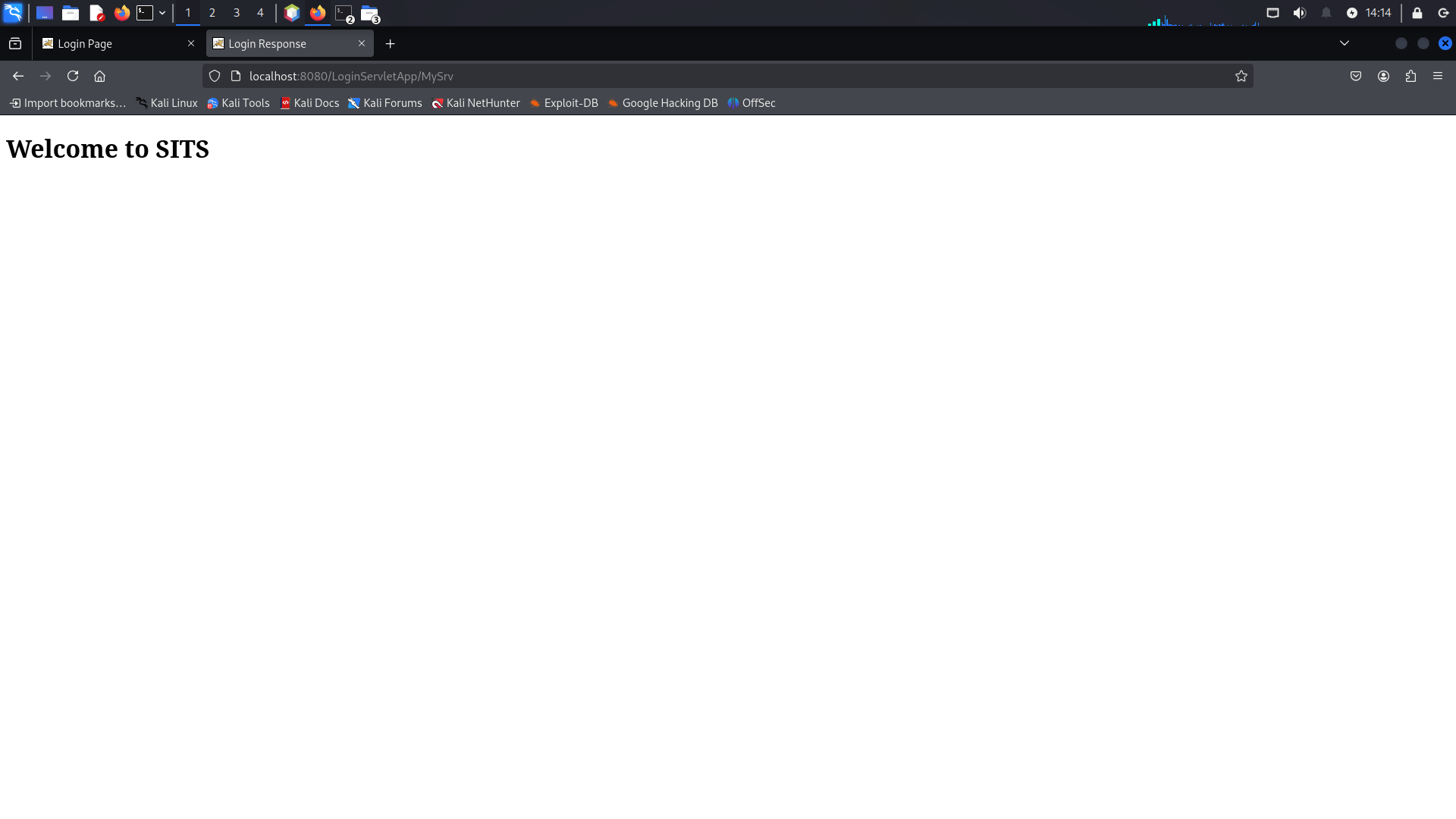
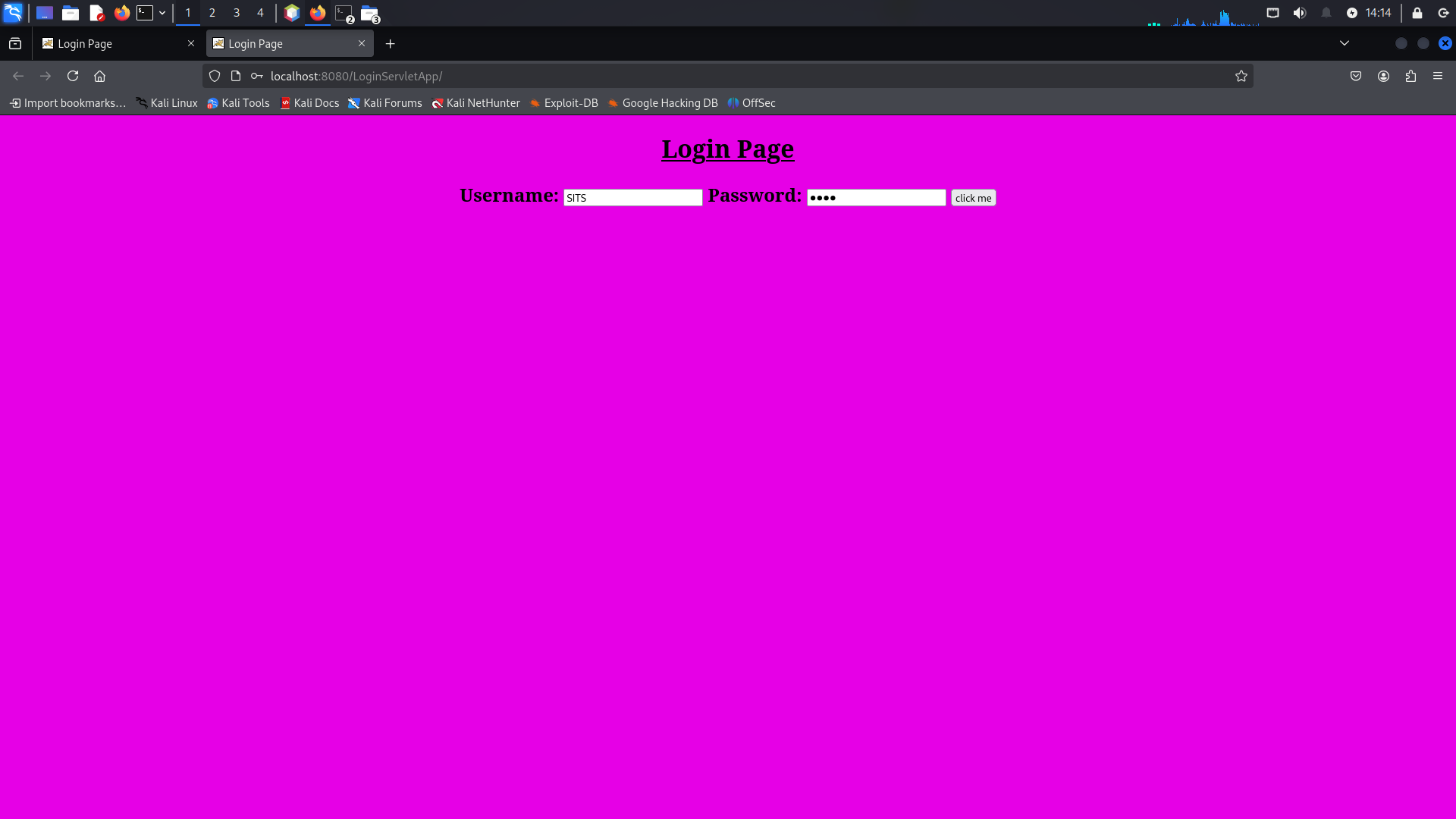
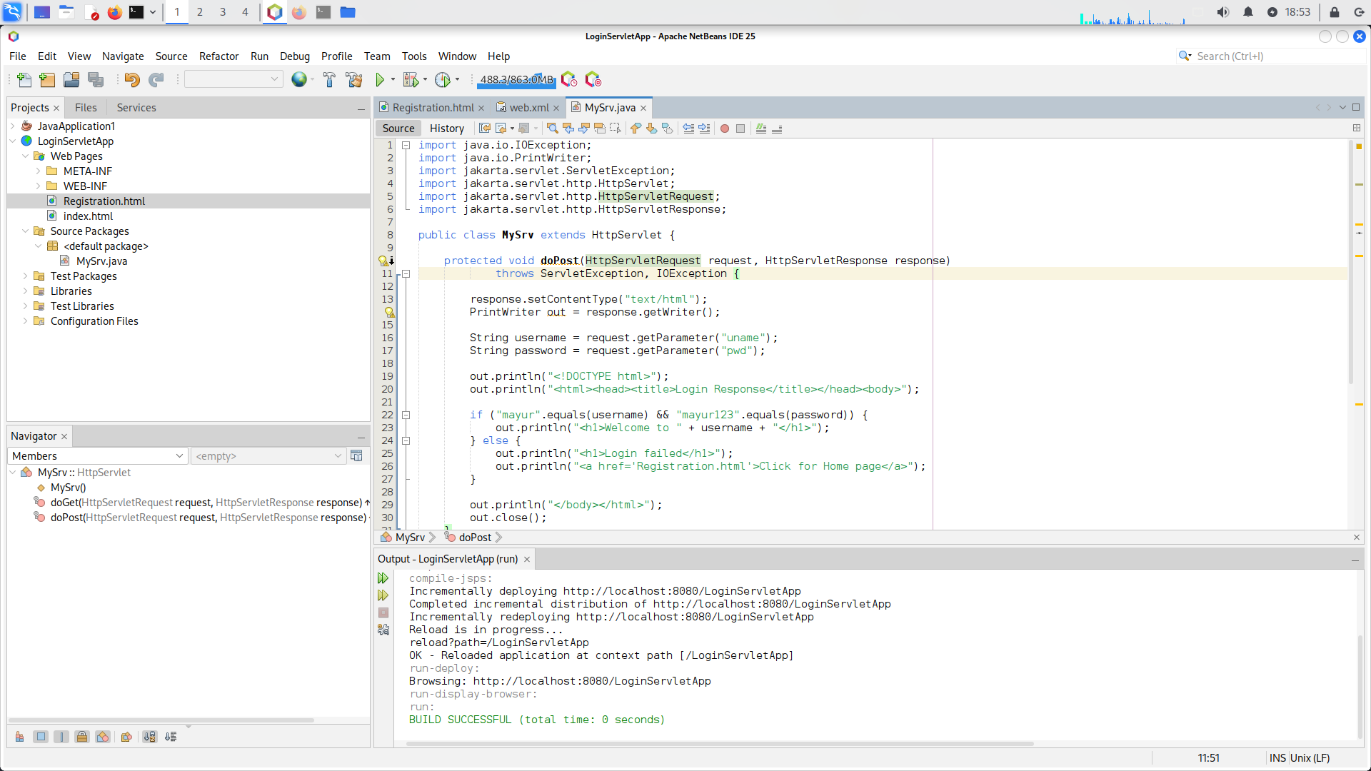
<welcome-file-list>

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

</welcome-file-list>

</web-app>

**Output:**



**Experiment-8**

**Code:**

**Exp8jdbc.java**

import java.sql.\*;

public class Exp8jdbc{

public static void main(String[] args) {

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

String user = "root";

String password = "password";

try {

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

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

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

Statement st = con.createStatement();

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);

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

while (rs.next()) {

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

}

con.close();

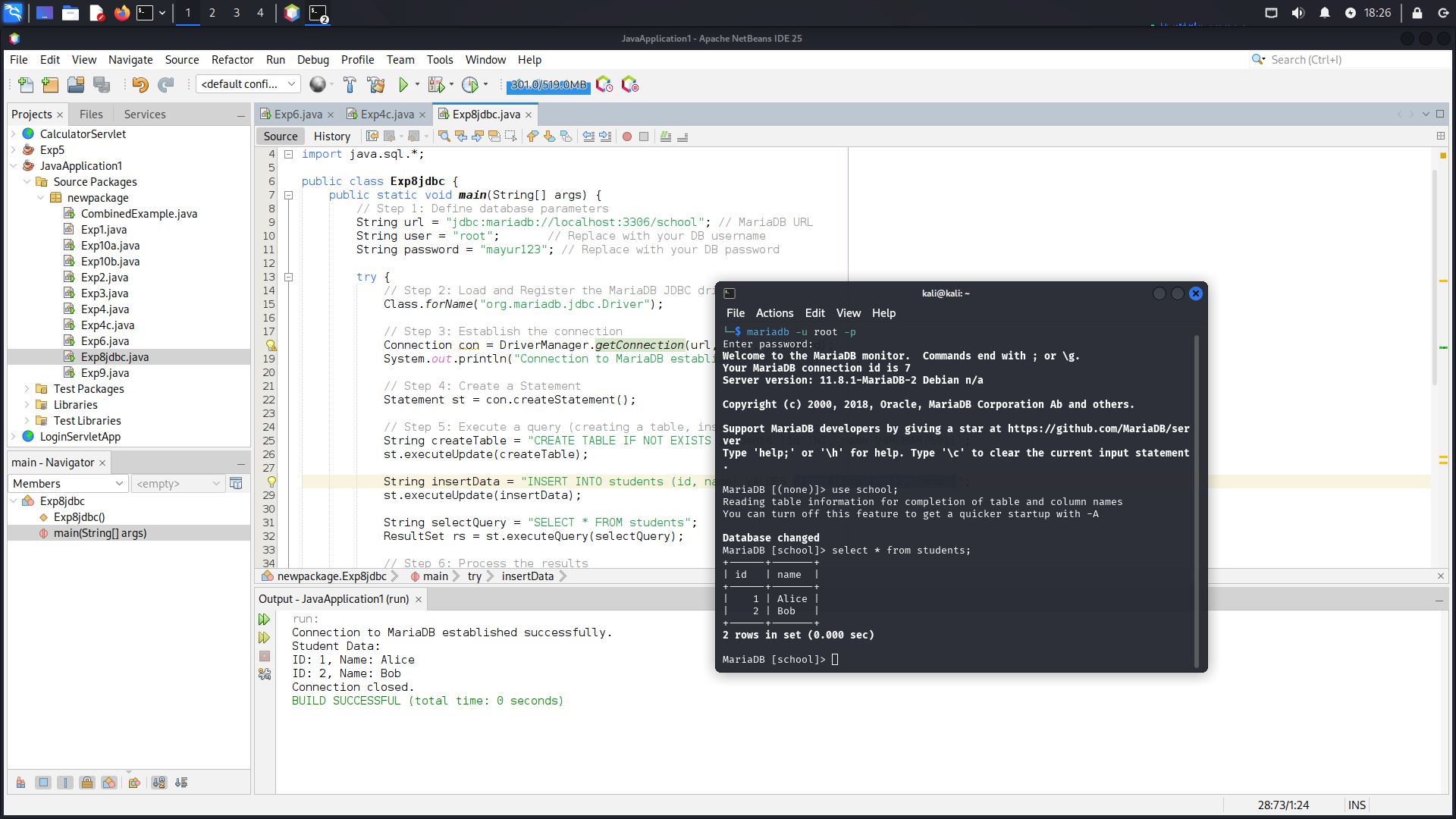
System.out.println("Connection closed.");

} catch (Exception e) {

e.printStackTrace();

}}}

**Output:**

****

**Experiment-9**

**Code:**

**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>

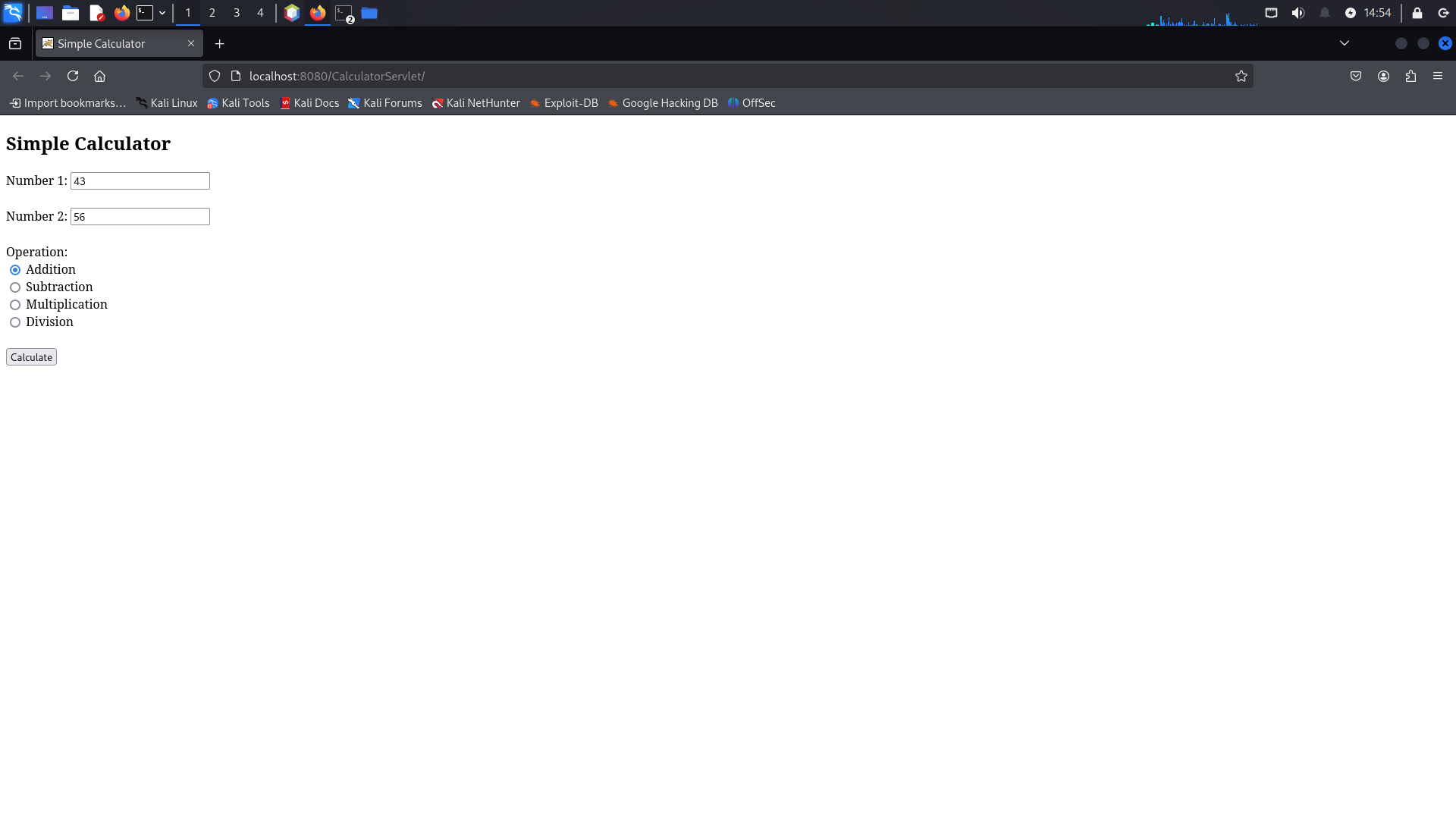
<welcome-file-list>

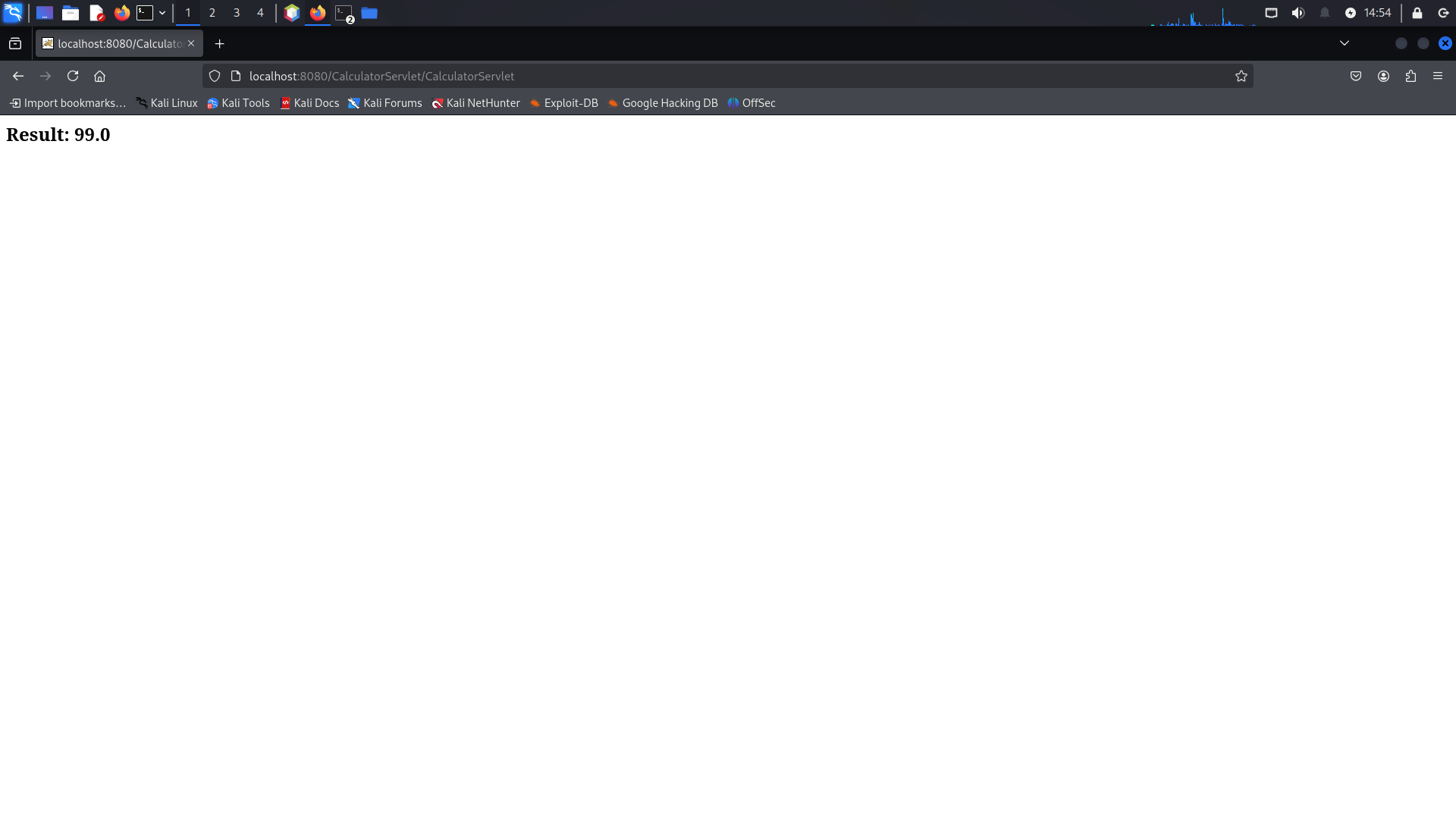
<welcome-file>calculator.html</welcome-file>

</welcome-file-list>

</web-app>

**Output:**





**Experiment-10**

**Code:**

**Index.jsp**

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<!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>

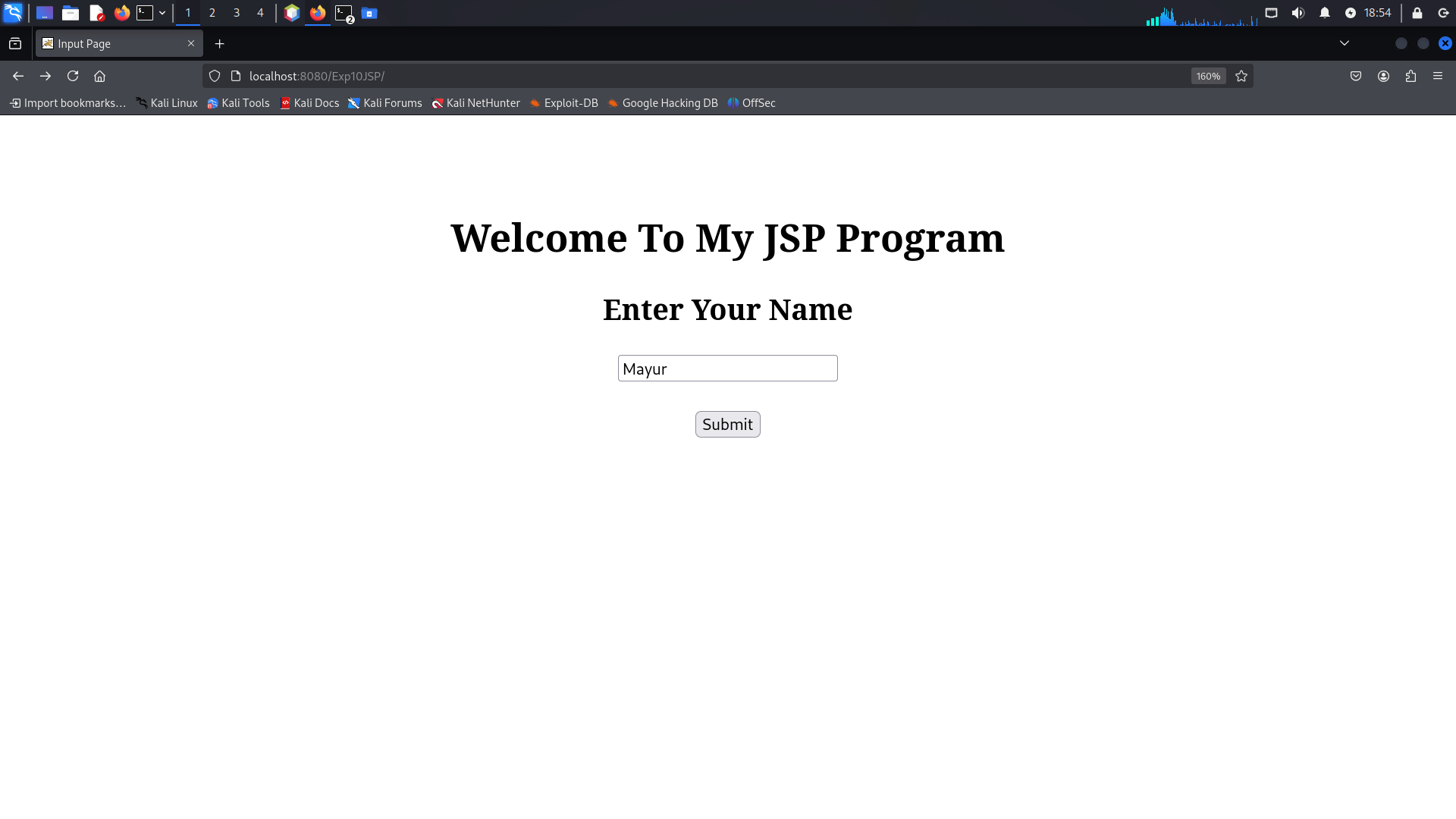
<input type="submit" value="Submit">

</form>

</body>

</html>

**Output:**

****

**Experiment-11**

**Code:**

**Exp11.java**

package newpackage;

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 Exp11 extends Frame implements ActionListener, ItemListener {

Dialog dialog;

Label l;

Exp11() {

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) {

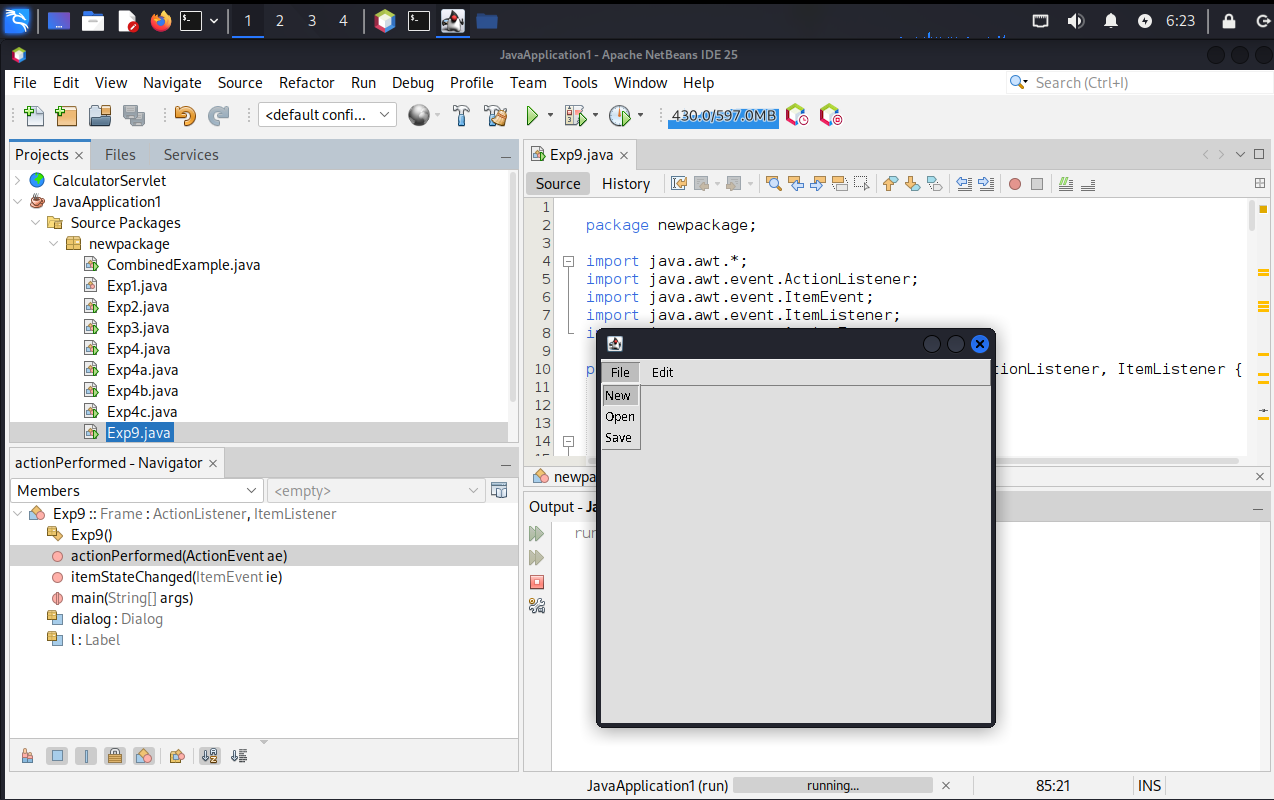
Exp11 md = new Exp11();

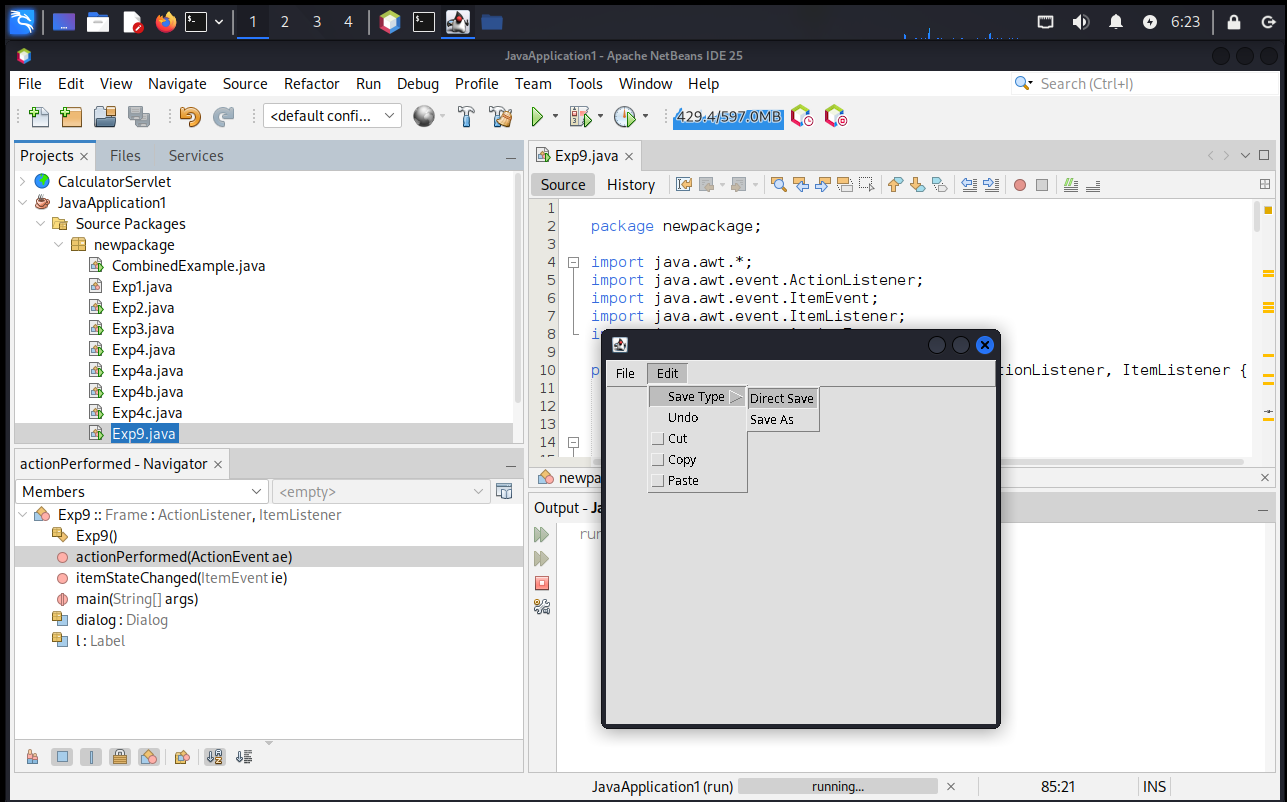
md.setVisible(true);

md.setSize(400, 400);

}}

**Output:**





**Experiment-12a**

**Code:**

**Exp12a.java**

package newpackage;

import java.awt.\*;

public class Exp12a {

public static void main(String args[]) {

Frame f = new Frame("Grid Demo");

f.setSize(500, 500);

f.setLayout(new GridLayout(5, 5)); // 5x5 grid

Font font = new Font("TimesRoman", Font.BOLD, 25);

f.setFont(font);

for (int i = 0; i < 25; i++) {

Label lbl = new Label(String.valueOf(i + 1), Label.CENTER);

// Simple color logic using HSB (Hue, Saturation, Brightness)

Color color = Color.getHSBColor(i / 25.0f, 0.8f, 0.9f);

lbl.setBackground(color);

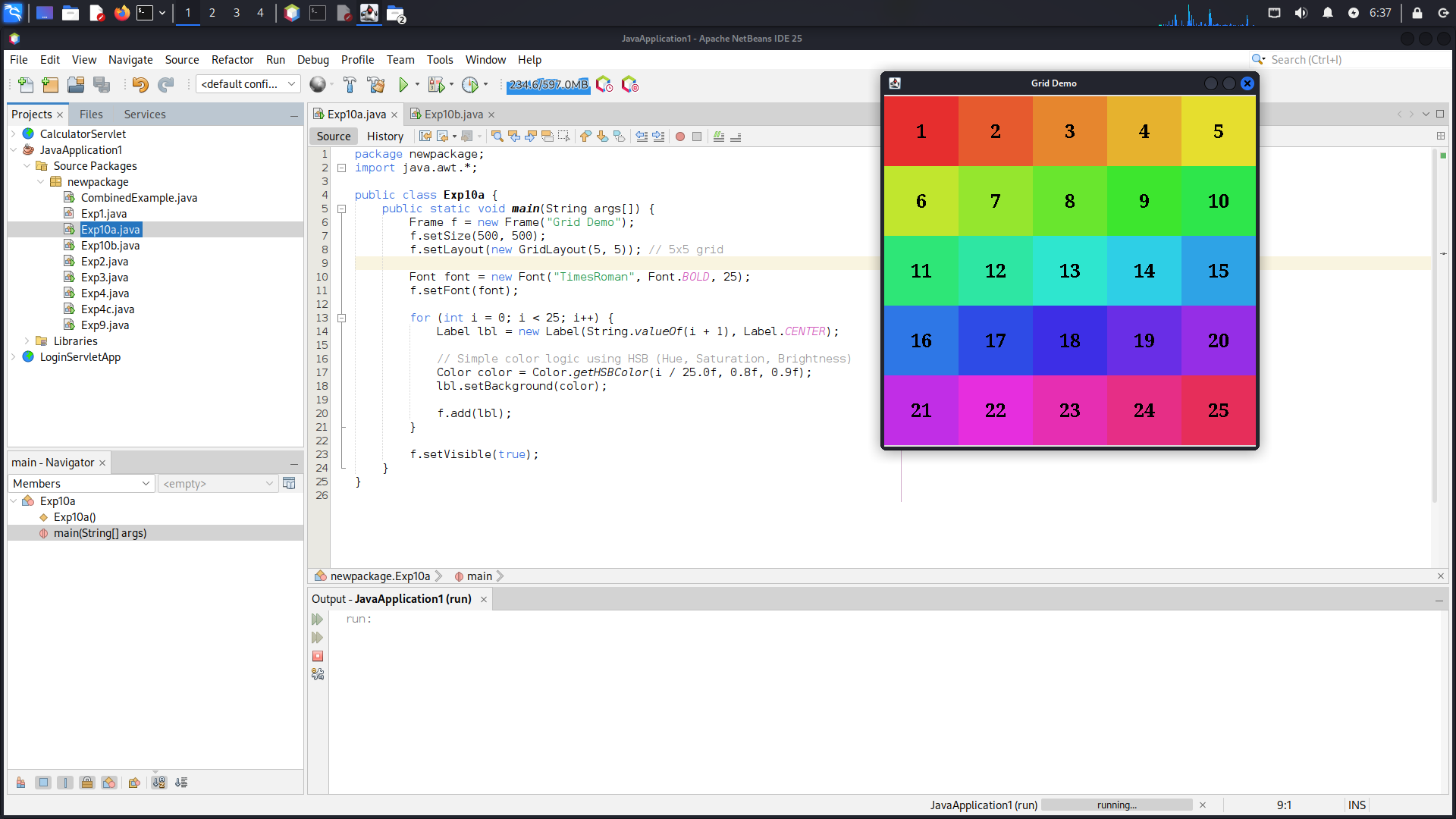
f.add(lbl);

}

f.setVisible(true);

}}

**Output:**



**Experiment-12b**

**Code:**

**Exp12b.java**

import java.awt.\*;

public class Exp12b {

public static void main(String args[]) {

Frame f = new Frame("BorderLayout Demo");

f.setSize(400, 400);

f.setLayout(new BorderLayout());

// Creating buttons

Button northButton = new Button("North");

Button southButton = new Button("South");

Button eastButton = new Button("East");

Button westButton = new Button("West");

Button centerButton = new Button("Center");

// Adding buttons to specific regions

f.add(northButton, BorderLayout.NORTH);

f.add(southButton, BorderLayout.SOUTH);

f.add(eastButton, BorderLayout.EAST);

f.add(westButton, BorderLayout.WEST);

f.add(centerButton, BorderLayout.CENTER);

f.setVisible(true);

}

}

**Output:**

