Experiment-1

Code:

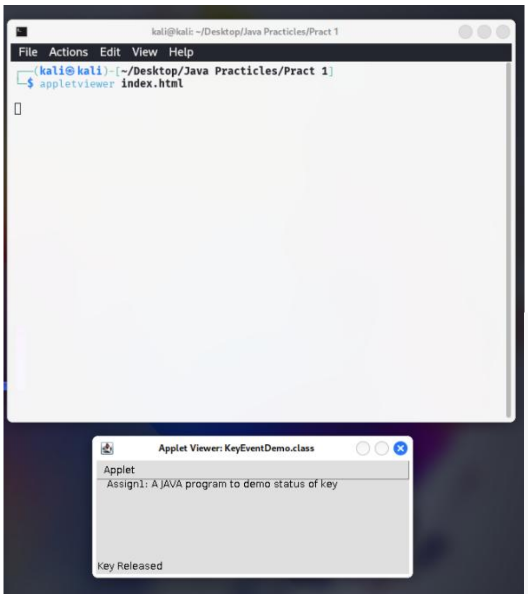
KeyEventDemo.java

```
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

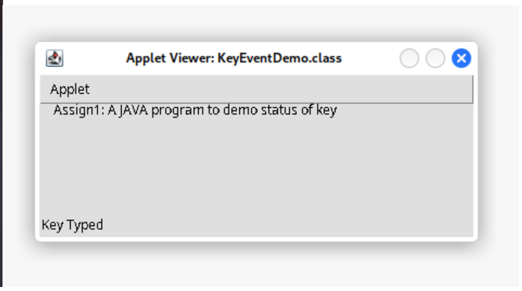
```
<html>
   <body>
      <applet code="KeyEventDemo.class" width="400" height="100">
      </applet>
   </body>
</html>
```

Output:



```
kali@kali: ~/Desktop/Java Practicles/Pract 1
┌──(kali㉿kali)-[~/Desktop/Java Practicles/Pract 1]
└─$ appletviewer index.html
```

```
Applet Viewer: KeyEventDemo.class
Applet
 Assign1: A JAVA program to demo status of key

Key Released
```



```
Applet Viewer: KeyEventDemo.class
Applet
 Assign1: A JAVA program to demo status of key

Key Typed
```

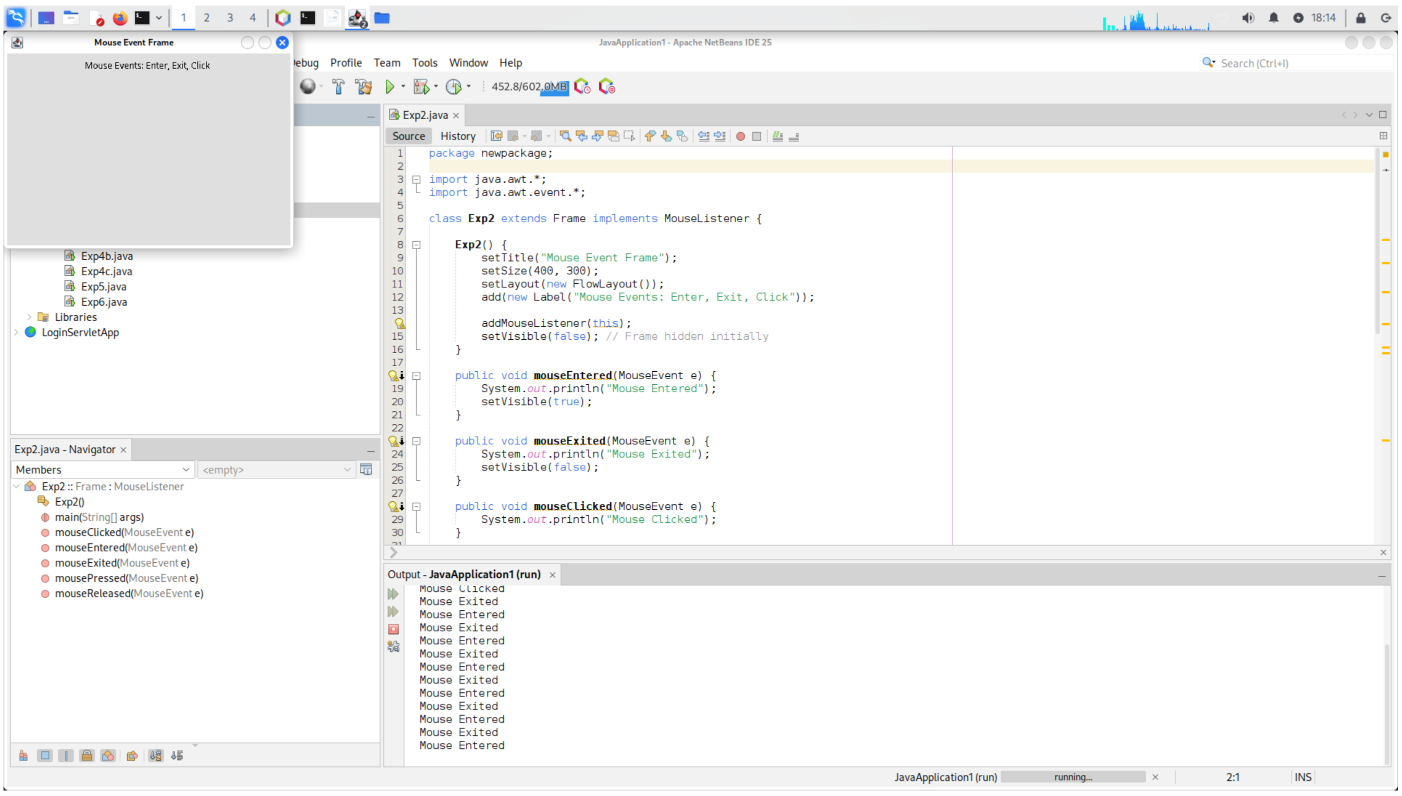
Experiment-2

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:



Mouse Event Frame

Mouse Events: Enter, Exit, Click

```java
package newpackage;

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");
    }
```

Output - JavaApplication1 (run)

```
Mouse Clicked
Mouse Exited
Mouse Entered
Mouse Exited
Mouse Entered
Mouse Exited
Mouse Entered
Mouse Exited
Mouse Entered
Mouse Exited
Mouse Entered
Mouse Exited
Mouse Entered
```

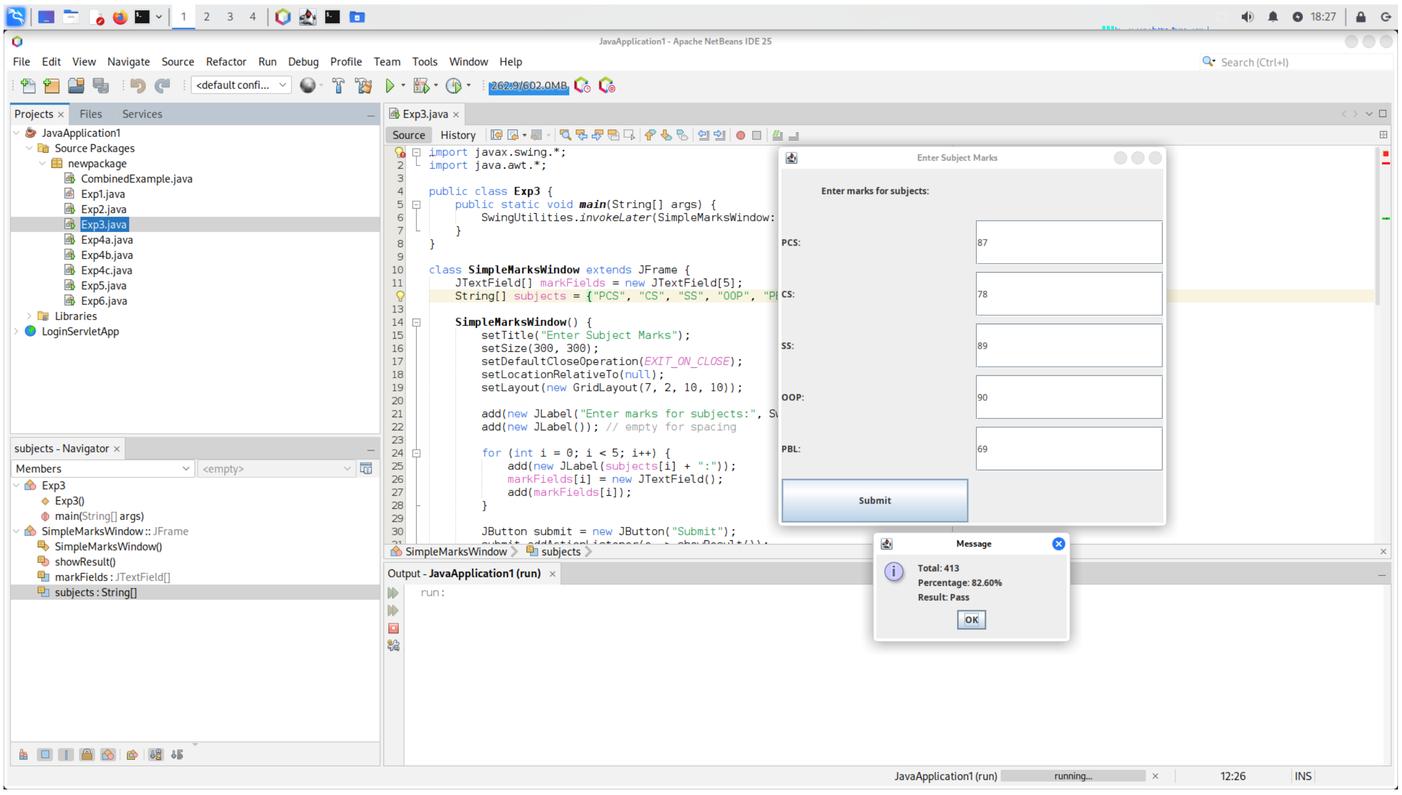
Experiment-3

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:



```
import javax.swing.*;
import java.awt.*;

public class Exp3 {
    public static void main(String[] args) {
        SwingUtilities.invokeLater(SimpleMarksWindow:
    }
}

class SimpleMarksWindow extends JFrame {
    JTextField[] markFields = new JTextField[5];
    String[] subjects = {"PCS", "CS", "SS", "OOP", "PI

    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:", S
        add(new JLabel()); // empty for spacing

        for (int i = 0; i < 5; i++) {
            add(new JLabel(subjects[i] + ":"));
            markFields[i] = new JTextField();
            add(markFields[i]);
        }

        JButton submit = new JButton("Submit");
```

```
run:
```

Enter Subject Marks

Enter marks for subjects:

PCS:	87
CS:	78
SS:	89
OOP:	90
PBL:	69

Message

```
Total: 413
Percentage: 82.60%
Result: Pass
```

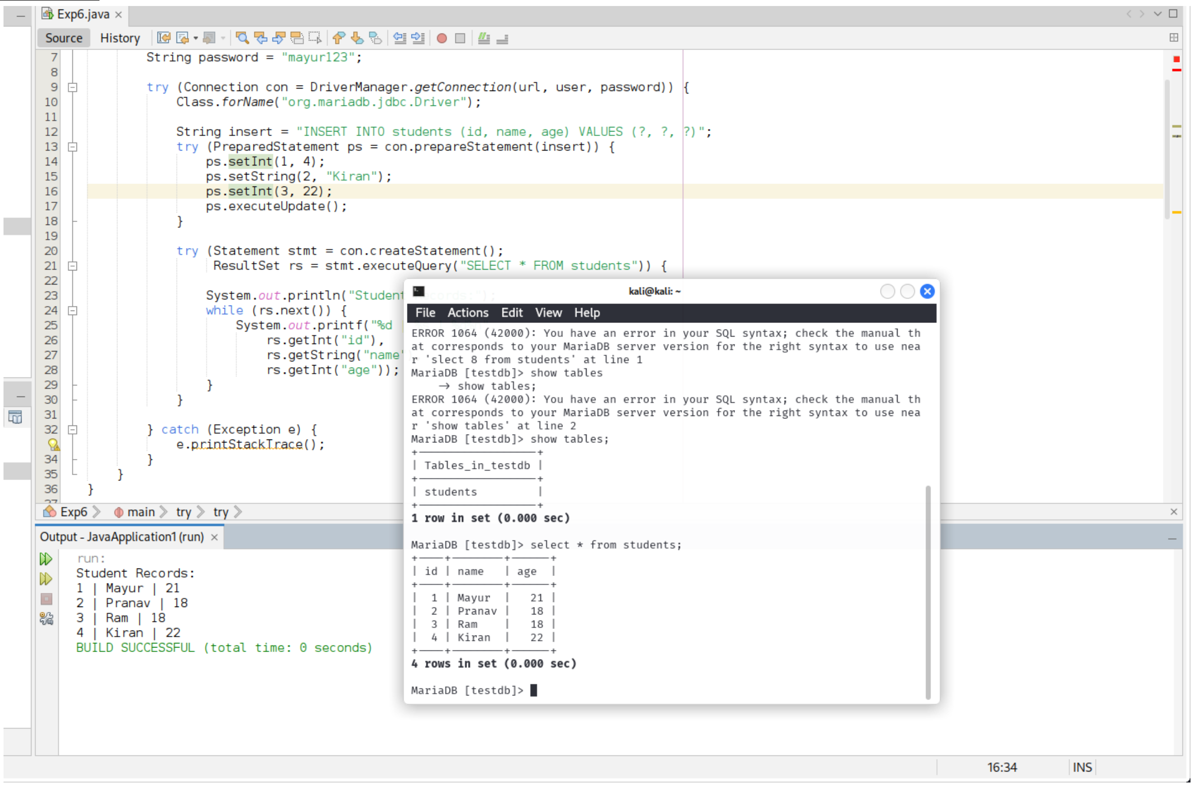
Experiment-4

Code:

Exp4.java

```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:



```java
        String password = "mayur123";

        try (Connection con = DriverManager.getConnection(url, user, password)) {
            Class.forName("org.mariadb.jdbc.Driver");

            String insert = "INSERT INTO students (id, name, age) VALUES (?, ?, ?)";
            try (PreparedStatement ps = con.prepareStatement(insert)) {
                ps.setInt(1, 4);
                ps.setString(2, "Kiran");
                ps.setInt(3, 22);
                ps.executeUpdate();
            }

            try (Statement stmt = con.createStatement();
                 ResultSet rs = stmt.executeQuery("SELECT * FROM students")) {

                System.out.println("Student
                while (rs.next()) {
                    System.out.printf("%d
                        rs.getInt("id"),
                        rs.getString("name"
                        rs.getInt("age"));
                }
            }

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

```
run:
Student Records:
1 | Mayur | 21
2 | Pranav | 18
3 | Ram | 18
4 | Kiran | 22
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual th
at corresponds to your MariaDB server version for the right syntax to use nea
r 'slect 8 from students' at line 1
MariaDB [testdb]> show tables
    → show tables;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual th
at corresponds to your MariaDB server version for the right syntax to use nea
r 'show tables' at line 2
MariaDB [testdb]> show tables;
+-----------------+
| Tables_in_testdb |
+-----------------+
| students         |
+-----------------+
1 row in set (0.000 sec)

MariaDB [testdb]> select * from students;
+----+--------+-----+
| id | name   | age |
+----+--------+-----+
|  1 | Mayur  |  21 |
|  2 | Pranav |  18 |
|  3 | Ram    |  18 |
|  4 | Kiran  |  22 |
+----+--------+-----+
4 rows in set (0.000 sec)

MariaDB [testdb]>
```

Experiment-5

Code:

Addition.java (Remote Interface)

```java
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)

```java
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;}}
```

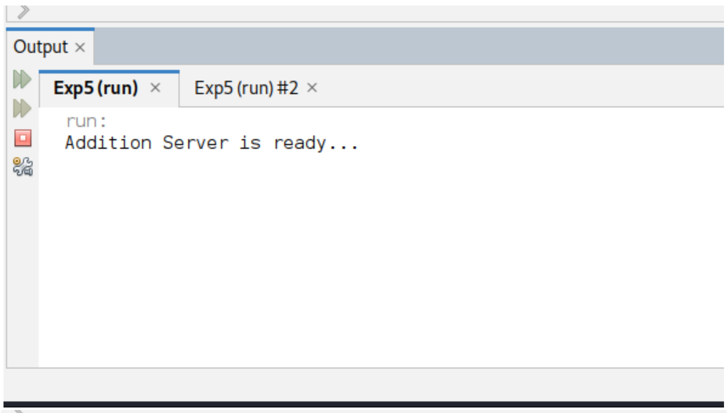
AdditionServer.java (RMI Server)

```java
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();
      }}}
```

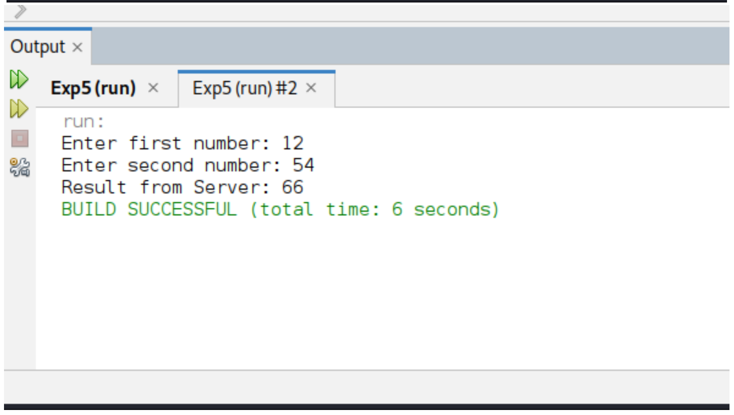
AdditionClient.java (RMI Client)

```java
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:



```
run:
Addition Server is ready...
```



```
run:
Enter first number: 12
Enter second number: 54
Result from Server: 66
BUILD SUCCESSFUL (total time: 6 seconds)
```

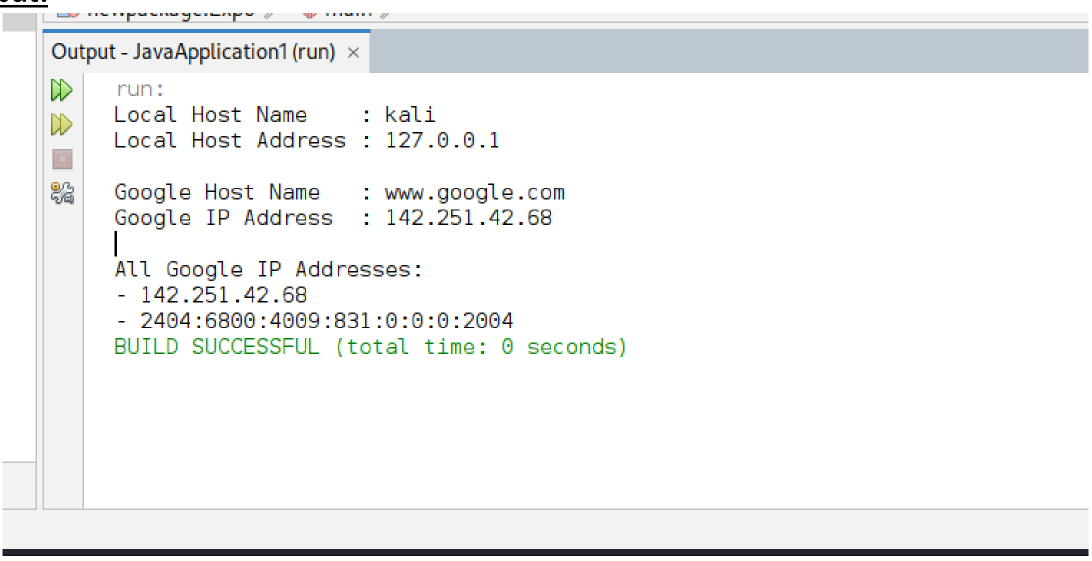
Experiment-6

Code:

Exp6.java

```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:



```
run:
Local Host Name    : kali
Local Host Address : 127.0.0.1

Google Host Name   : www.google.com
Google IP Address  : 142.251.42.68

All Google IP Addresses:
- 142.251.42.68
- 2404:6800:4009:831:0:0:0:2004
BUILD SUCCESSFUL (total time: 0 seconds)
```

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

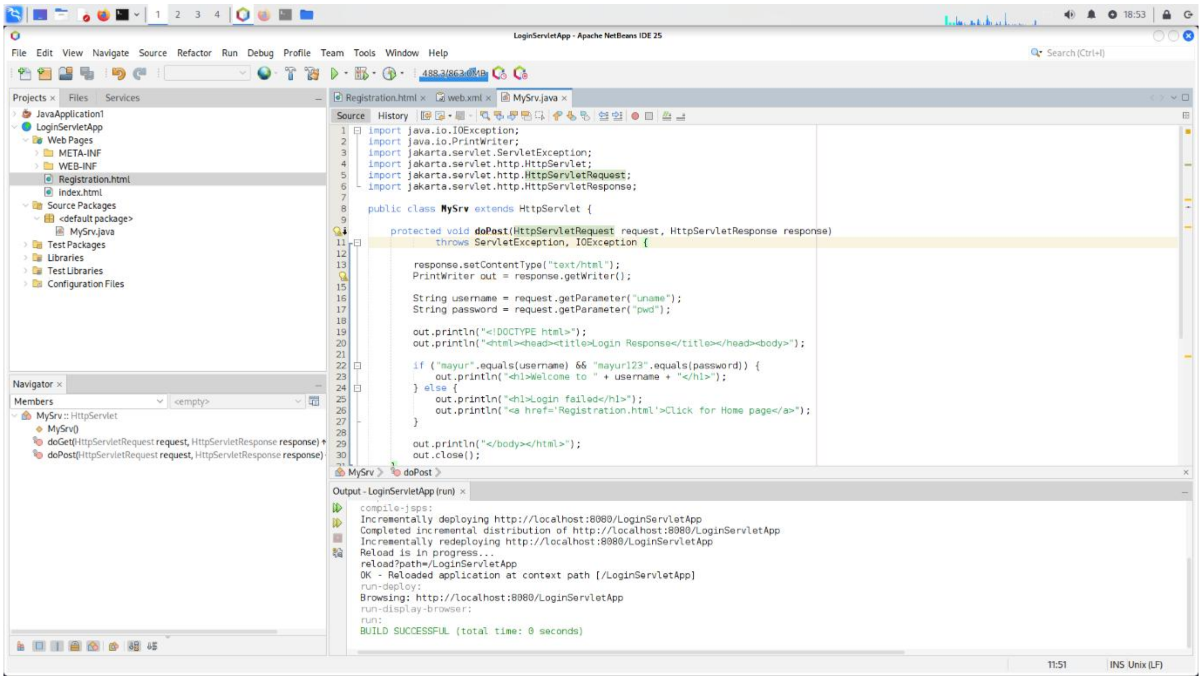
```
<!DOCTYPE html>
<html><head><title>Login Page</title></head>
<body bgcolor='#e600e6'>
  <form action='MySrv' method="post">
    <center>
      <h1><u>Login Page</u></h1><h2>
        Username: <input type="text" name="uname" />
        Password: <input type="password" name="pwd" />
        <input 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>
```

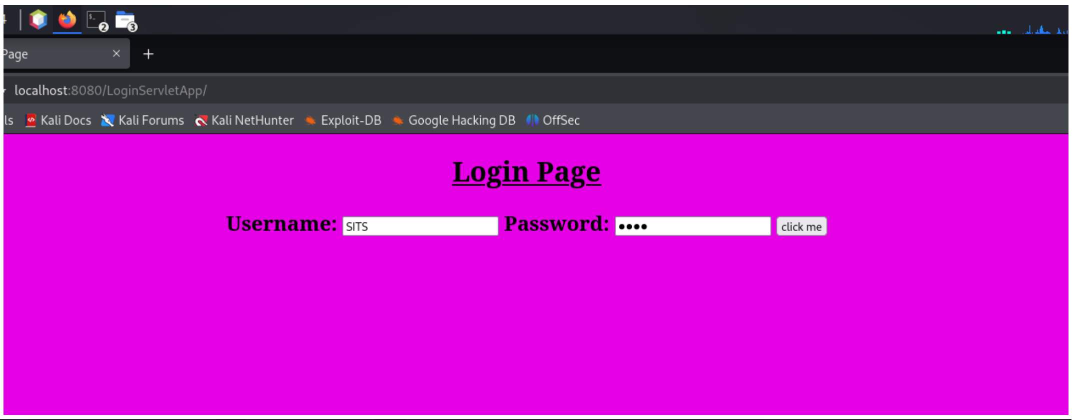
```
</servlet>
  <servlet-mapping>
    <servlet-name>MySrv</servlet-name>
    <url-pattern>/MySrv</url-pattern>
  </servlet-mapping>
  <welcome-file-list>
    <welcome-file>Registration.html</welcome-file>
  </welcome-file-list>
</web-app>
```

Output:



```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 ("mayur".equals(username) && "mayur123".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();
```

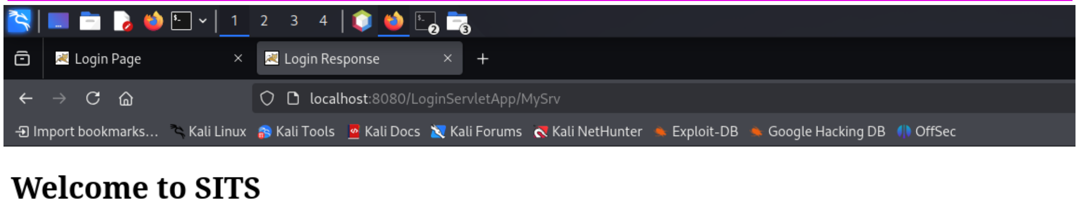
```
compile-jsps:
Incrementally deploying http://localhost:8080/LoginServletApp
Completed incremental distribution of http://localhost:8080/LoginServletApp
Incrementally redeploying http://localhost:8080/LoginServletApp
Reload is in progress...
reload?path=/LoginServletApp
OK - Reloaded application at context path [/LoginServletApp]
run-deploy:
Browsing: http://localhost:8080/LoginServletApp
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 0 seconds)
```



localhost:8080/LoginServletApp/

Login Page

Username: SITS **Password:** •••• click me



localhost:8080/LoginServletApp/MySrv

Welcome to SITS

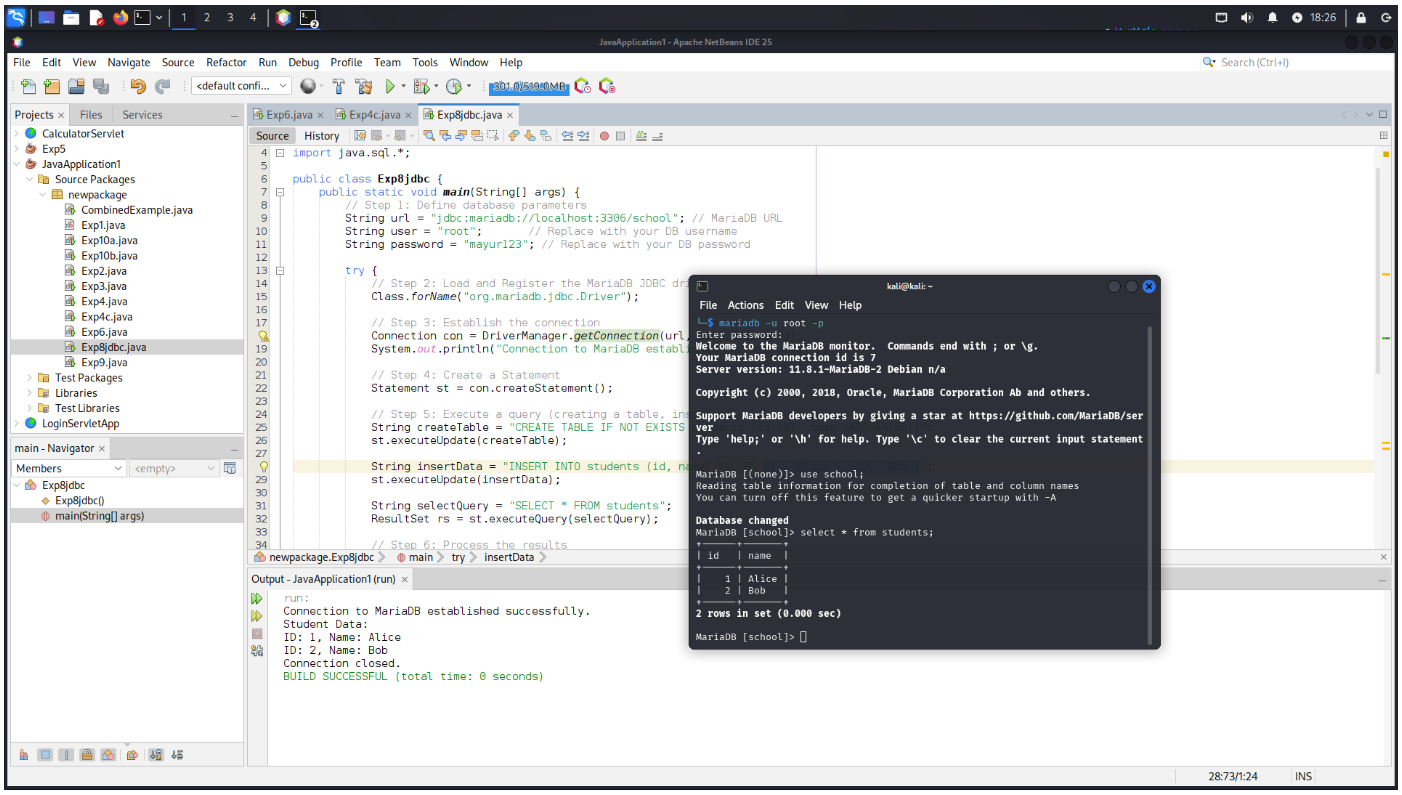
Experiment-8

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:



```
import java.sql.*;

public class Exp8jdbc {
    public static void main(String[] args) {
        // Step 1: Define database parameters
        String url = "jdbc:mariadb://localhost:3306/school"; // MariaDB URL
        String user = "root";        // Replace with your DB username
        String password = "mayur123"; // Replace with your DB password

        try {
            // Step 2: Load and Register the MariaDB JDBC dri
            Class.forName("org.mariadb.jdbc.Driver");

            // Step 3: Establish the connection
            Connection con = DriverManager.getConnection(url,
            System.out.println("Connection to MariaDB establi

            // Step 4: Create a Statement
            Statement st = con.createStatement();

            // Step 5: Execute a query (creating a table, ins
            String createTable = "CREATE TABLE IF NOT EXISTS
            st.executeUpdate(createTable);

            String insertData = "INSERT INTO students (id, na
            st.executeUpdate(insertData);

            String selectQuery = "SELECT * FROM students";
            ResultSet rs = st.executeQuery(selectQuery);

            // Step 6: Process the results
```

```
run:
Connection to MariaDB established successfully.
Student Data:
ID: 1, Name: Alice
ID: 2, Name: Bob
Connection closed.
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
└─$ mariadb -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 7
Server version: 11.8.1-MariaDB-2 Debian n/a

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Support MariaDB developers by giving a star at https://github.com/MariaDB/ser
ver
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement
.

MariaDB [(none)]> use school;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [school]> select * from students;
+------+-------+
| id   | name  |
+------+-------+
|    1 | Alice |
|    2 | Bob   |
+------+-------+
2 rows in set (0.000 sec)

MariaDB [school]>
```

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>");
  }}
```

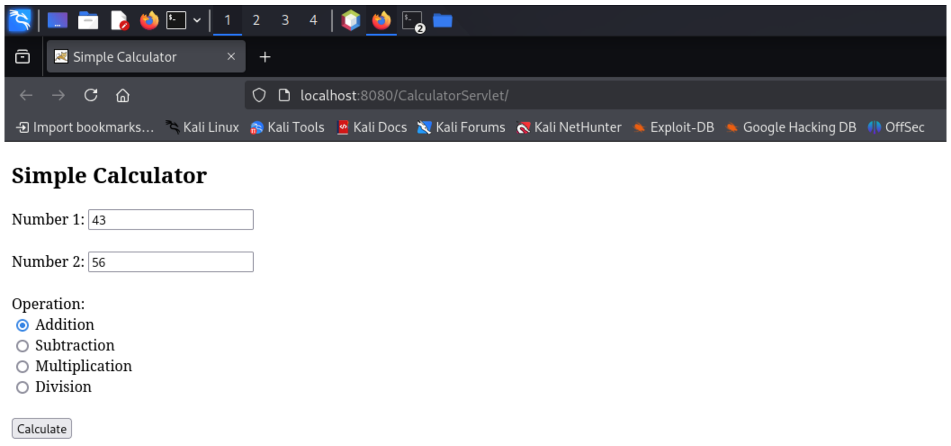
calculator.html

```
<!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:


Simple Calculator
localhost:8080/CalculatorServlet/
Import bookmarks... Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

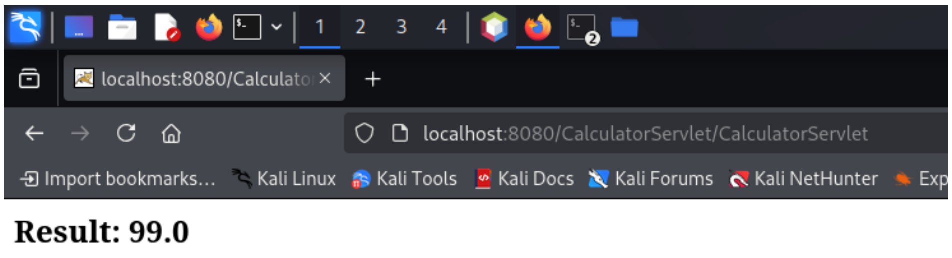
Simple Calculator

Number 1: 43

Number 2: 56

Operation:
[✓ Addition]
Subtraction
Multiplication
Division

Calculate


localhost:8080/Calculato
localhost:8080/CalculatorServlet/CalculatorServlet
Import bookmarks... Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exp

Result: 99.0

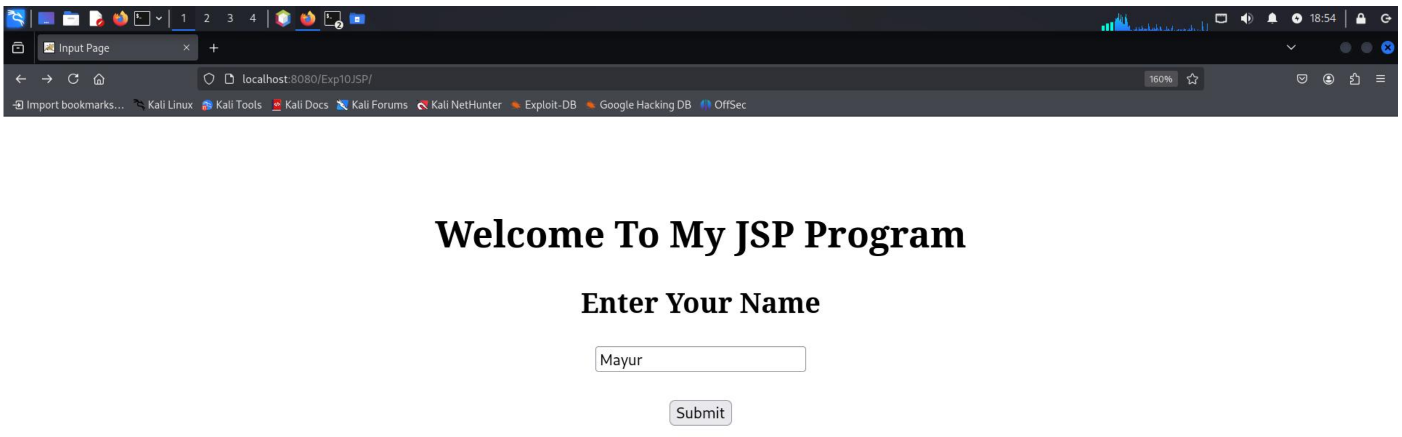
Experiment-10

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:


Input Page
localhost:8080/Exp10JSP/
Import bookmarks... Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
Welcome To My JSP Program
Enter Your Name
Mayur
Submit

Experiment-11

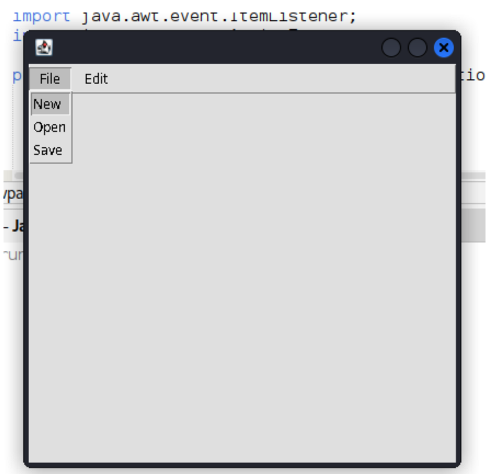
Code:

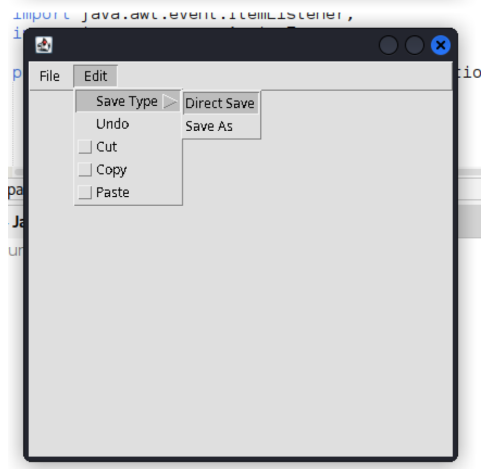
Exp11.java

```
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:





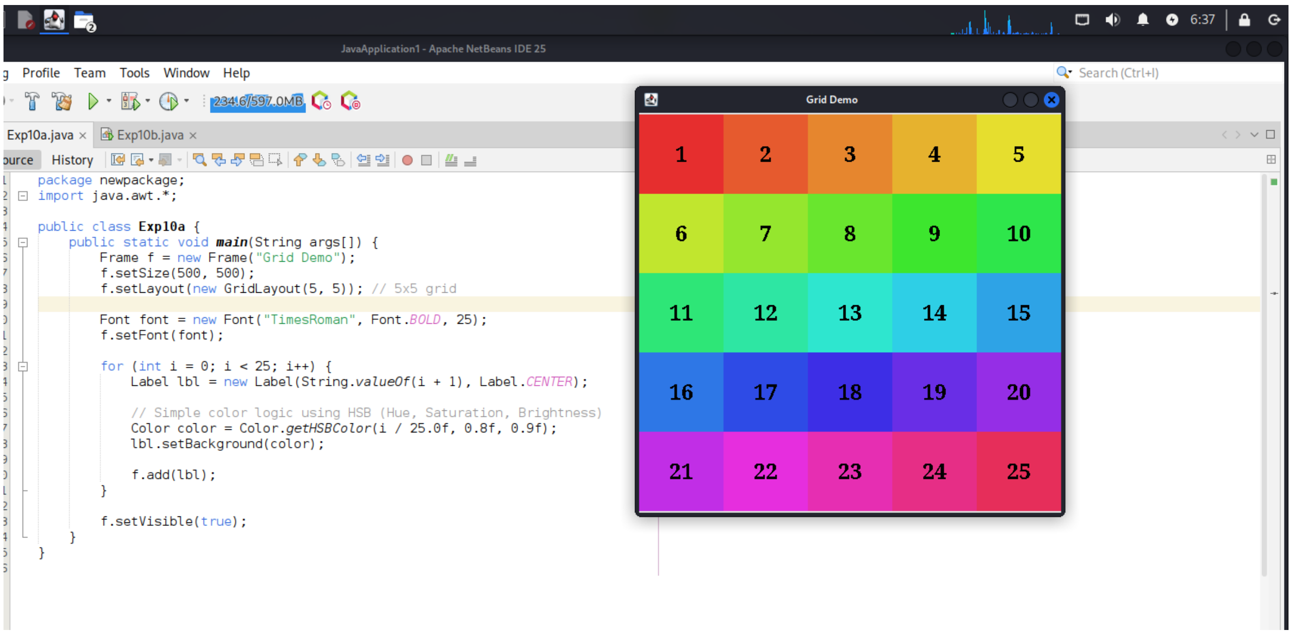
Experiment-12a

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:



```
package newpackage;
import java.awt.*;

public class Exp10a {
    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);
    }
}
```

```
1   2   3   4   5
6   7   8   9   10
11  12  13  14  15
16  17  18  19  20
21  22  23  24  25
```

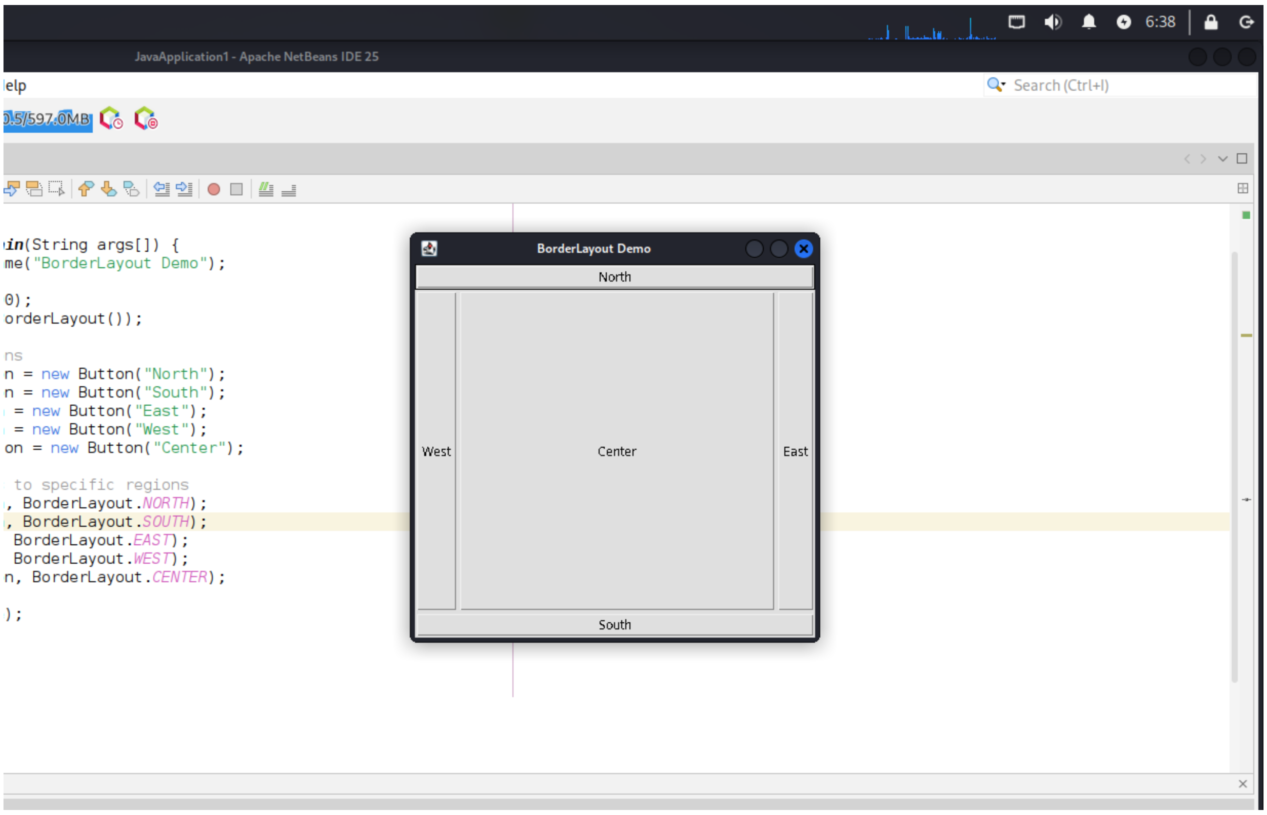
Experiment-12b

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:



```
in(String args[]) {
me("BorderLayout Demo");

0);
orderLayout());

ns
n = new Button("North");
n = new Button("South");
 = new Button("East");
 = new Button("West");
on = new Button("Center");

 to specific regions
, BorderLayout.NORTH);
, BorderLayout.SOUTH);
 BorderLayout.EAST);
 BorderLayout.WEST);
n, BorderLayout.CENTER);

);
```