UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

NAAC GRADE A+ ACCREDITED UNIVERSITY

Worksheet 4

Student Name: Rahul Saxena **UID: 24MCI10204**
Branch: MCA(AI-ML) **Section/Group: 3B**
Semester: 2nd **Date of Performance: 28/02/25**
Subject Name: Advanced Internet Programming Lab **Subject Code: 24CAP-652**

1. Aim/Overview of the practical:

This assignment demonstrates the implementation of a simple addition operation using JavaBeans and Servlets in a Java EE application. The objective is to understand how JavaBeans facilitate data encapsulation and how Servlets process HTTP requests in a web-based environment.

2. Hardware and Software Requirements:

- Java EE (Jakarta EE)
- JavaBeans
- Servlets
- HTML
- NetBeans IDE
- Tomcat Server

3. Commands/Steps:

a) Set Up Your Development Environment.

- Install NetBeans IDE 24.
- Install JDK 21 and configure it in NetBeans.
- Install Tomcat 10.1 and configure it as a server in NetBeans.
- Install SQL Workbench for database connectivity.

b) Create a New Web Application Project in NetBeans.

- Open NetBeans IDE.

- Click on File > New Project.
- Select Java Web > Web Application and click Next.
- Enter your project name and set the location.
- Choose Apache Tomcat 10.1 as your server and JDK 21 as your platform.
- Click Finish to create the project.

c) AddBeans.java package eejb; import

```
java.io.Serializable; public class AddBeans
implements Serializable {
   private int a;     private
int b;     public void
setA(int a) {
      this.a = a;
   }
   public void setB(int b) {
      this.b = b;
   }
   public int getSum() {
      return a + b;
   }
}
```

d) NewClass.java

```
package servlet; import eejb.AddBeans; //
Import JavaBean import java.io.IOException;
import java.io.PrintWriter; import
jakarta.servlet.ServletException; import
```

```
jakarta.servlet.annotation.WebServlet; import
jakarta.servlet.http.HttpServlet; import
jakarta.servlet.http.HttpServletRequest; import
jakarta.servlet.http.HttpServletResponse;
@WebServlet("/AddServlet") // URL Mapping
public class NewClass extends HttpServlet {

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

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        try {
            // Retrieve input values from request            int a =
Integer.parseInt(request.getParameter("a"));            int b =
Integer.parseInt(request.getParameter("b"));

            // Use JavaBean to store values
            AddBeans addBean = new AddBeans();
            addBean.setA(a);            addBean.setB(b);
int sum = addBean.getSum();  // Compute sum
```

```
            // Display result in response            out.println("<h2> Result: " + sum + "</h2>");
        } catch (NumberFormatException e) {            out.println("<h2>Error: Invalid number
        format. Please enter valid integers.</h2>");
              } finally {
        out.close();         }

           }
        }
```

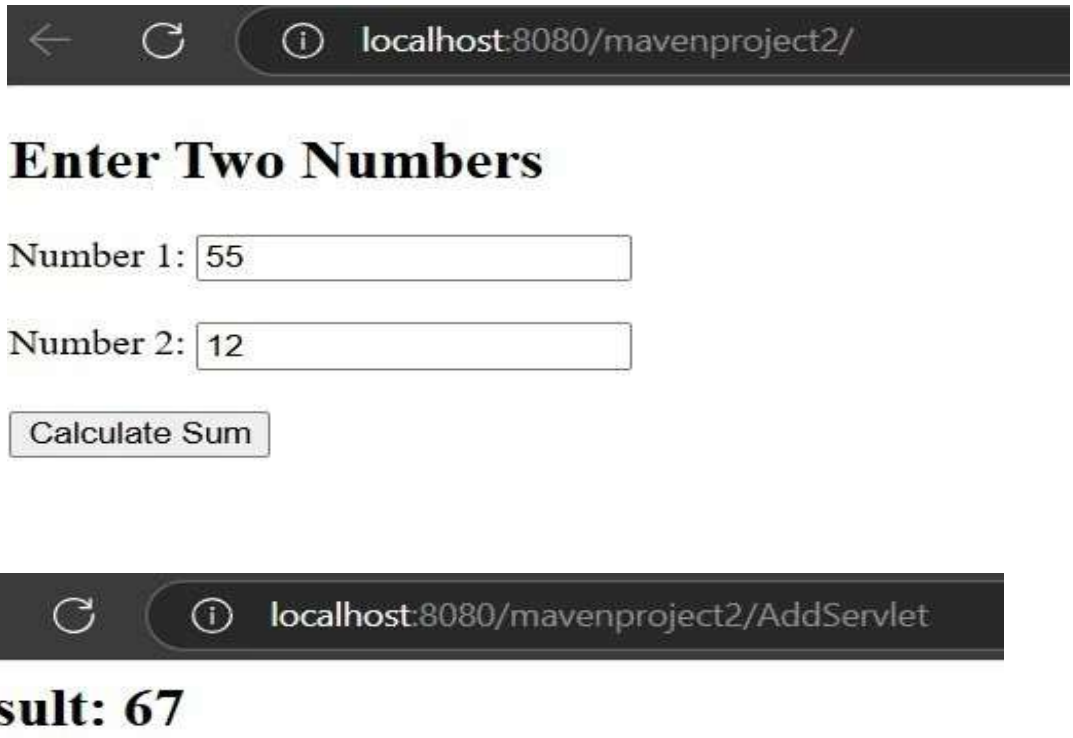
e) index.html

```
<!DOCTYPE html>
<html>
<head>
   <title>Addition Form</title>
</head>
<body>
   <h2>Enter Two Numbers</h2>
   <form action="AddServlet" method="post">
      <label>Number 1:</label>
      <input type="text" name="a" required><br><br>

      <label>Number 2:</label>
      <input type="text" name="b" required><br><br>
```

```
        <button type = "submit">Calculate Sum</button>
    </form>
</body>
</html>
```

4. Result:



Conclusion

This assignment demonstrates how JavaBeans and Servlets work together in a Java EE application. JavaBeans provide a reusable way to handle data, while Servlets manage user interactions efficiently. By implementing this addition operation, we have successfully processed user inputs and displayed dynamic results using Java technologies.

Learning Outcomes

- Understood the concept of JavaBeans and their role in Java EE applications.
- Learned how to retrieve and process form data using Servlets.
- Gained hands-on experience in deploying a Java EE web application on Tomcat. • Explored the integration of JavaBeans with Servlets for better code modularity.