# 山西工程技术学院

## 《JavaEE 程序设计》实验报告

（ 2024 - 2025 学年第 学期）

课程名称： Java EE 程序设计

专业班级：22 计算机科学与技术一班

学　　号：2210708130

学生姓名：郝泓毅

任课教师：王晓霞

2025 年 5 月

# 实验报告

| 实验名称 | Java EE 开发环境配置 | | | 指导教师 | 王晓霞 |
|---|---|---|---|---|---|
| 实验类型 | 验证型 | 实验学时 | 2 | 实验时间 | 2.19 |

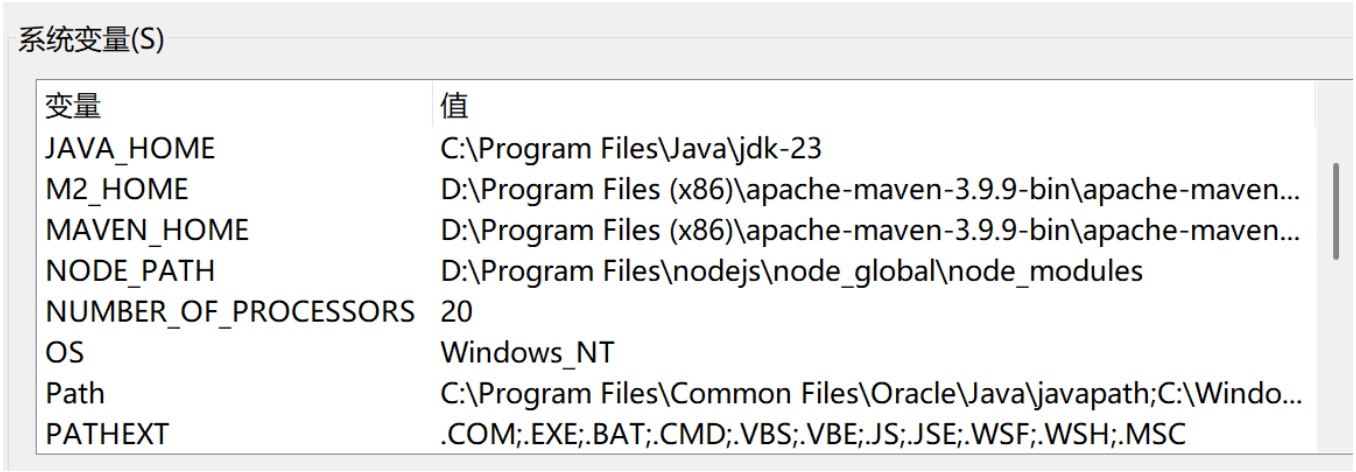
**一、实验目的与要求**

了解 JAVA 虚拟机及 JAVA 运行环境；安装 JDK；安装服务器 TOMCAT；安装 IDE MyEclipse。

**二、实验环境**

多媒体计算机 60 台。MySQL8.0

**三、实验内容和步骤**

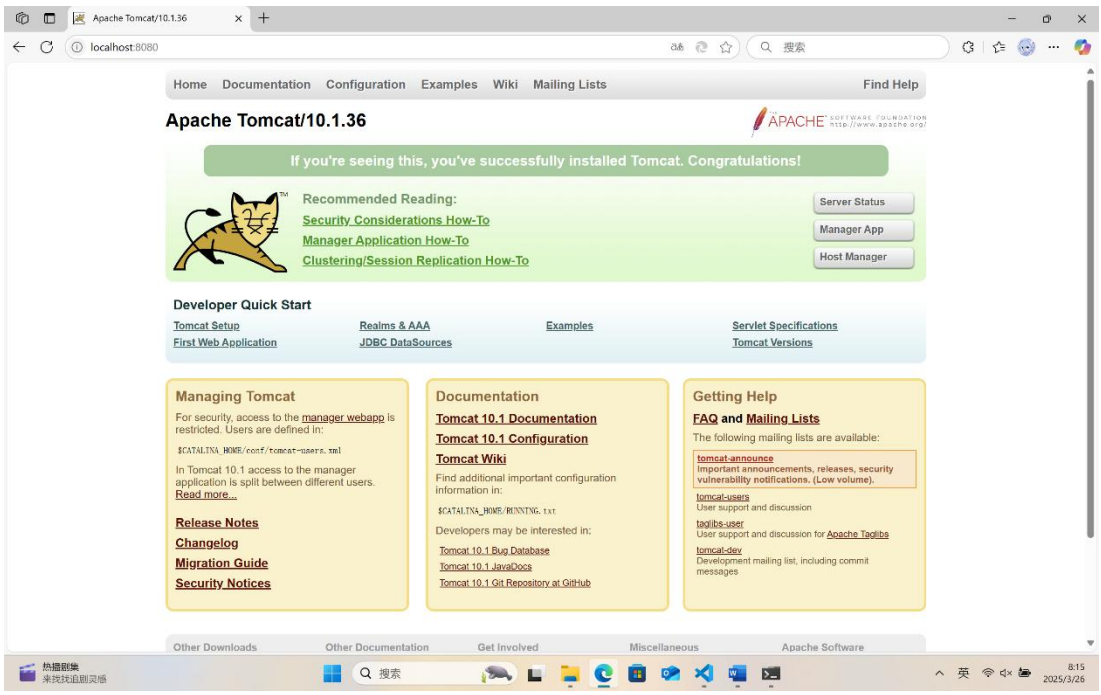



```
C:\Users\Lanyi>java -version
java version "23.0.2" 2025-01-21
Java(TM) SE Runtime Environment (build 23.0.2+7-58)
Java HotSpot(TM) 64-Bit Server VM (build 23.0.2+7-58, mixed mode, sharing)

C:\Users\Lanyi>javac -version
javac 23.0.2

C:\Users\Lanyi>
```

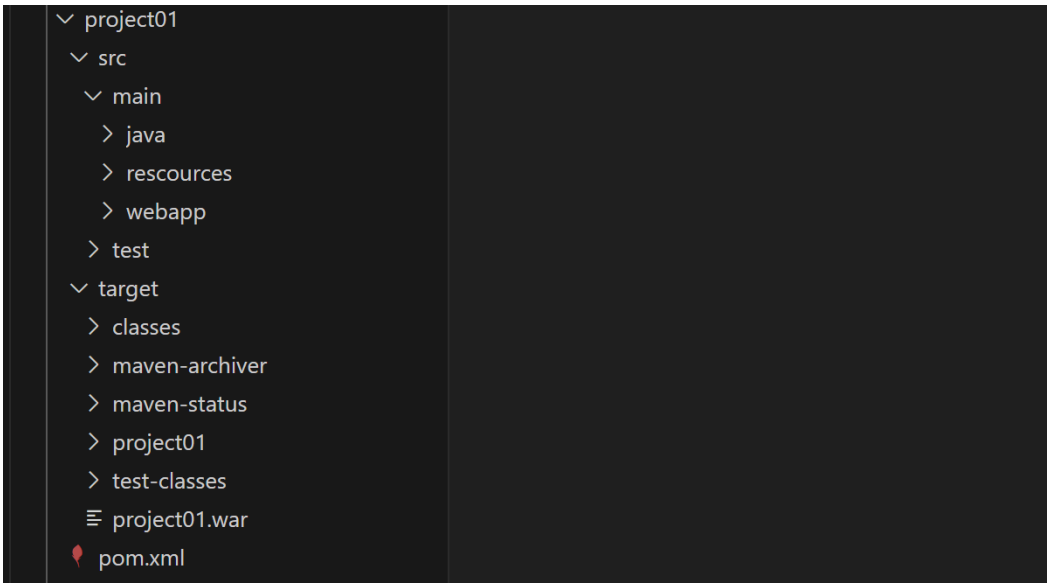
1. 安装 JDK，配置三个环境变量，测试 JDK 安装是否正确；

2. 安装服务器 TOMCAT，测试其安装是否正确；

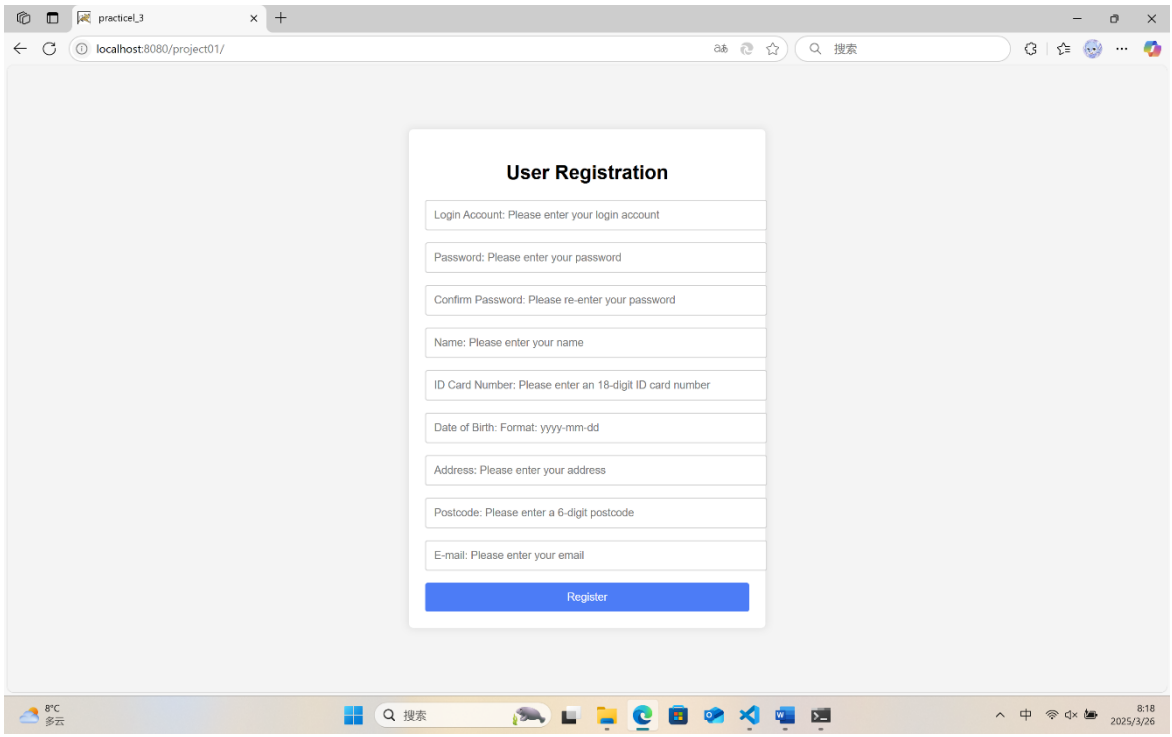
（3）安装 MyEclipse，在 MyEclipse 中绑定 JDK，配置 TOMCAT 服务器；




（4）练习新建 Java 动态项目；




（5）练习 JSP 文件的运行。




四、实验小结和思考

本次实验通过搭建 JavaEE 基础开发环境，使我进一步理解了 JDK、Tomcat 和开发工具之间的关系。配置环境变量时需要格外小心路径正确性；通过 MyEclipse 创建和运行 JSP 文件，初步体验了 JavaEE 开发的流程，也为后续 Web 开发实验打下基础。

| 实验成绩 | | 批阅日期 | | 批阅人 | |
|---|---|---|---|---|---|

# 实验报告

| 实验名称 | JDBC 编程训练 | | | 指导教师 | 王晓霞 |
|---|---|---|---|---|---|
| 实验类型 | 设计型 | 实验学时 | 4 | 实验时间 | 2.26 |

**二、实验目的与要求**

通过桥链接数据库，实现对数据的增删改查。

熟练掌握数据库的各种连接方法；

熟练掌握对数据库数据的增、删、改、查。

**二、实验环境**

多媒体计算机 60 台。MySQL8.0 javaweb

**三、实验内容和步骤**

创建数据库 StuManagement,新建表 student(sno,sname,ssex,sage),插入几条记录。创建数据源 mytudent，用 JDBC-ODBC 桥链接，或纯 JAVA 数据库链接，实现对数据的增、删、改、查。

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ page import="java.sql.*" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Student Information Management System</title>
    <link rel="stylesheet" href="css/stylesheet.css">
    <style>

    </style>
</head>
<body>
    <h1>Student Information Management System</h1>
    <%
        Connection con = null;
        PreparedStatement pstmt = null;
        ResultSet rs = null;
        String errorMessage = null;
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");

            // First connect to MySQL server to create database if not exists
            con =
DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/?useUnicode=true&characterEncoding=UT
```

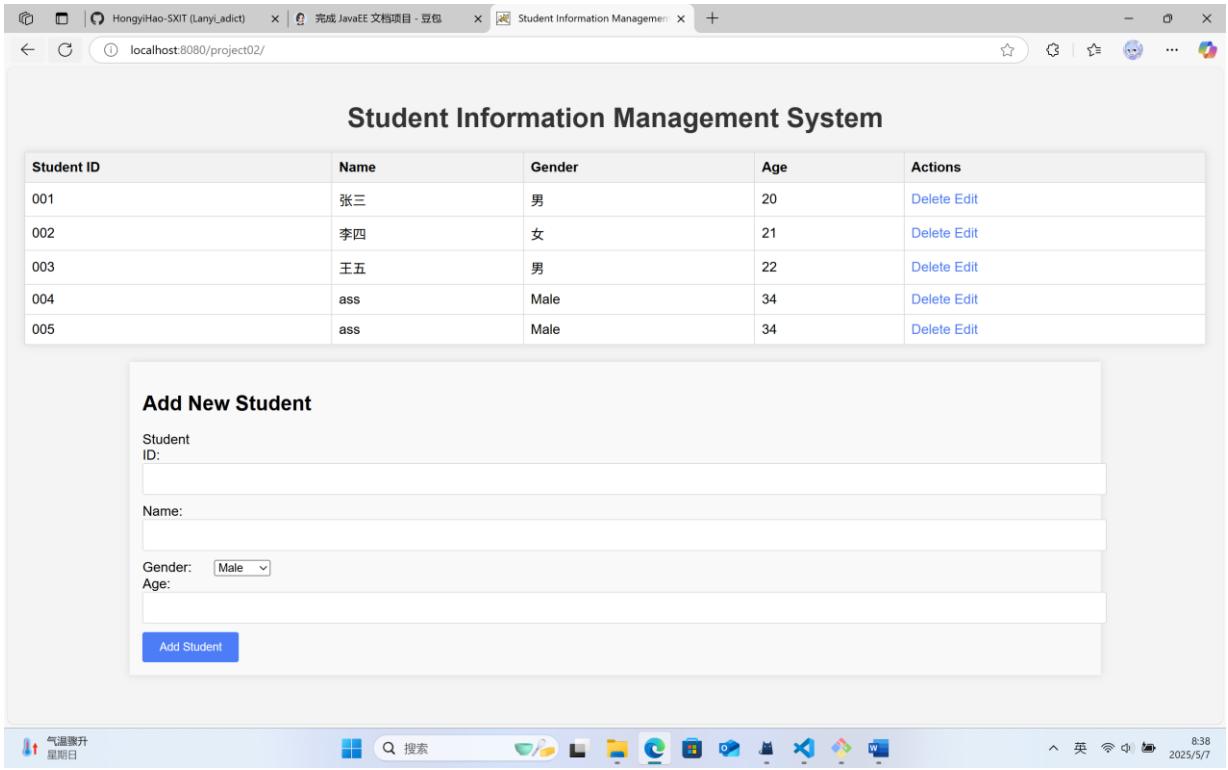
```
F-8&serverTimezone=UTC", "root", "123456");
            pstmt = con.prepareStatement("CREATE DATABASE IF NOT EXISTS StuManagement");
            pstmt.executeUpdate();

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

            con =
DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/StuManagement?useUnicode=true&chara
cterEncoding=UTF-8&serverTimezone=UTC", "root", "123456");

            // Create student table if not exists
            String createTableSQL = "CREATE TABLE IF NOT EXISTS student (" +
                    "sno VARCHAR(10) PRIMARY KEY," +
                    "sname VARCHAR(50) NOT NULL," +
                    "ssex VARCHAR(10) NOT NULL," +
                    "sage INT NOT NULL" +
                    ")";
            pstmt = con.prepareStatement(createTableSQL);
            pstmt.executeUpdate();
            pstmt.close();

            String checkSQL = "SELECT COUNT(*) FROM student";
            pstmt = con.prepareStatement(checkSQL);
            rs = pstmt.executeQuery();
            rs.next();
            int count = rs.getInt(1);
            rs.close();
            pstmt.close();

            if (count == 0) {
                String insertSQL = "INSERT INTO student (sno, sname, ssex, sage) VALUES (?,?,?,?)";
                pstmt = con.prepareStatement(insertSQL);

                pstmt.setString(1, "001");
                pstmt.setString(2, "Zhang San");
                pstmt.setString(3, "Male");
                pstmt.setInt(4, 20);
                pstmt.addBatch();

                pstmt.setString(1, "002");
                pstmt.setString(2, "Li Si");
                pstmt.setString(3, "Female");
                pstmt.setInt(4, 21);
```

```
            pstmt.addBatch();

            pstmt.setString(1, "003");
            pstmt.setString(2, "Wang Wu");
            pstmt.setString(3, "Male");
            pstmt.setInt(4, 22);
            pstmt.addBatch();

            pstmt.executeBatch();
            pstmt.close();
        }

        String action = request.getParameter("action");
        if (action != null) {
            if ("add".equals(action)) {
                // Add operation
                String sno = request.getParameter("sno");
                String sname = request.getParameter("sname");
                String ssex = request.getParameter("ssex");
                int sage = Integer.parseInt(request.getParameter("sage"));

                String addSQL = "INSERT INTO student (sno, sname, ssex, sage) VALUES (?,?,?,?)";
                pstmt = con.prepareStatement(addSQL);
                pstmt.setString(1, sno);
                pstmt.setString(2, sname);
                pstmt.setString(3, ssex);
                pstmt.setInt(4, sage);
                pstmt.executeUpdate();
                pstmt.close();

            } else if ("delete".equals(action)) {
                // Delete operation
                String sno = request.getParameter("sno");
                String deleteSQL = "DELETE FROM student WHERE sno = ?";
                pstmt = con.prepareStatement(deleteSQL);
                pstmt.setString(1, sno);
                pstmt.executeUpdate();
                pstmt.close();

            } else if ("update".equals(action)) {
                // Update operation
                String sno = request.getParameter("sno");
                String sname = request.getParameter("sname");
                String ssex = request.getParameter("ssex");
```

```
                    int sage = Integer.parseInt(request.getParameter("sage"));

                    String updateSQL = "UPDATE student SET sname = ?, ssex = ?, sage = ? WHERE sno
= ?";
                    pstmt = con.prepareStatement(updateSQL);
                    pstmt.setString(1, sname);
                    pstmt.setString(2, ssex);
                    pstmt.setInt(3, sage);
                    pstmt.setString(4, sno);
                    pstmt.executeUpdate();
                    pstmt.close();
                }
            }

            // Query operation
            String selectSQL = "SELECT * FROM student ORDER BY sno";
            pstmt = con.prepareStatement(selectSQL);
            rs = pstmt.executeQuery();
    %>
    <table>
        <tr>
            <th>Student ID</th>
            <th>Name</th>
            <th>Gender</th>
            <th>Age</th>
            <th>Actions</th>
        </tr>
        <% while (rs.next()) { %>
        <tr>
            <td><%= rs.getString("sno") %></td>
            <td><%= rs.getString("sname") %></td>
            <td><%= rs.getString("ssex") %></td>
            <td><%= rs.getInt("sage") %></td>
            <td>
                <a href="student_management.jsp?action=delete&sno=<%= rs.getString("sno") %>"
onclick="return confirm('Are you sure to delete this student?')">Delete</a>
                <a href="javascript:void(0);" onclick="editStudent('<%= rs.getString("sno") %>', '<%=
rs.getString("sname") %>', '<%= rs.getString("ssex") %>', <%= rs.getInt("sage") %>)">Edit</a>
            </td>
        </tr>
        <% } %>
    </table>

    <form action="student_management.jsp" method="post">
```

```
        <input type="hidden" name="action" value="add">
        <h2>Add New Student</h2>
        <label for="sno">Student ID:</label>
        <input type="text" id="sno" name="sno" required><br>
        <label for="sname">Name:</label>
        <input type="text" id="sname" name="sname" required><br>
        <label for="ssex">Gender:</label>
        <select id="ssex" name="ssex" required>
            <option value="Male">Male</option>
            <option value="Female">Female</option>
            <option value="Other">Other</option>
        </select><br>
        <label for="sage">Age:</label>
        <input type="number" id="sage" name="sage" min="10" max="50" required><br>
        <input type="submit" value="Add Student">
    </form>

    <script>
        function editStudent(sno, sname, ssex, sage) {
            var newSname = prompt("Enter new name", sname);
            if (newSname === null) return;

            var newSsex = prompt("Enter new gender (Male/Female/Other)", ssex);
            if (newSsex === null) return;

            var newSage = prompt("Enter new age (10-50)", sage);
            if (newSage === null) return;

            if (newSname && newSsex && newSage) {
                window.location.href = "student_management.jsp?action=update&sno=" + sno +
                                       "&sname=" + encodeURIComponent(newSname) +
                                       "&ssex=" + encodeURIComponent(newSsex) +
                                       "&sage=" + encodeURIComponent(newSage);
            }
        }
    </script>
    <%
        } catch (ClassNotFoundException e) {
            errorMessage = "Database driver error: " + e.getMessage();
        } catch (SQLException e) {
            errorMessage = "Database operation error: " + e.getMessage();
        } catch (NumberFormatException e) {
            errorMessage = "Age must be a number: " + e.getMessage();
        } finally {
```

```
            try {
                if (rs != null) rs.close();
                if (pstmt != null) pstmt.close();
                if (con != null) con.close();
            } catch (SQLException e) {
                errorMessage = "Error closing database connection: " + e.getMessage();
            }
        }
        if (errorMessage != null) {
            out.println("<p class='error'>Error: " + errorMessage + "</p>");
        }
    %>
</body>
</html>
```




五、实验小结和思考

本实验使我掌握了 JDBC 的基本用法，了解了 Web 项目与数据库交互的流程。调试过程中注意 SQL 注入问题、编码格式及异常处理的重要性。通过编写学生信息管理系统，我对 MVC 结构有了初步认知，也更加熟练掌握了 JSP 中使用 JDBC 进行数据库操作的技巧。

| 实验成绩 | | 批阅日期 | | 批阅人 | |
|---|---|---|---|---|---|

# 实验报告

| 实验名称 | JSP 编程练习 | | | 指导教师 | 王晓霞 |
|---|---|---|---|---|---|
| 实验类型 | 设计型 | 实验学时 | 4 | 实验时间 | 3.12 |

**三、实验目的与要求**

会正确编写 JSP 文件并运行出结果。

编写 JSP 页面、使用注释、编写表达式、程序段、声明的方法以及常见的指令；

练习表单开发；

练习 JSP 内置对象的使用。

**二、实验环境**

多媒体计算机 60 台。MySQL8.0 javaweb

**三、实验内容和步骤**

用 application 编写投票计数程序；

**Vote.jsp:**

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Voting Page</title>
</head>
<body>
    <h1>Please select the option you support to vote.</h1>
    <form action="countVote.jsp" method="post">
        <input type="radio" name="option" value="OptionA"> Option A<br>
        <input type="radio" name="option" value="OptionB"> Option B<br>
        <input type="radio" name="option" value="OptionC"> Option C<br>
        <input type="submit" value="Vote">
    </form>
</body>
</html>
```

**Result.jsp:**

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

```
<head>
    <meta charset="UTF-8">
    <title>Voting Results</title>
</head>
<body>
    <h1>Voting Results</h1>
    <%
        Integer optionAVotes = (Integer) application.getAttribute("OptionA");
        Integer optionBVotes = (Integer) application.getAttribute("OptionB");
        Integer optionCVotes = (Integer) application.getAttribute("OptionC");

        if (optionAVotes == null) {
            optionAVotes = 0;
        }
        if (optionBVotes == null) {
            optionBVotes = 0;
        }
        if (optionCVotes == null) {
            optionCVotes = 0;
        }
    %>
    <p>Votes for Option A: <%= optionAVotes %></p>
    <p>Votes for Option B: <%= optionBVotes %></p>
    <p>Votes for Option C: <%= optionCVotes %></p>
    <a href="vote.jsp">Back to the voting page</a>
</body>
</html>
```

**countVote.jsp:**

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Vote Processing</title>
</head>
<body>
    <%
        String selectedOption = request.getParameter("option");
        if (selectedOption != null) {
            Integer voteCount = (Integer) application.getAttribute(selectedOption);
            if (voteCount == null) {
                voteCount = 0;
```

```
            }
            voteCount++;
            application.setAttribute(selectedOption, voteCount);
            out.println("You have successfully voted for " + selectedOption + "!");
        } else {
            out.println("Please select an option to vote!");
        }
    %>
    <br>
    <a href="vote.jsp">Back to the voting page</a>
    <a href="result.jsp">View the voting results</a>
</body>
</html>
```

用 session 开发购物车。

**Count.jsp**

```
<%@ page language="java" import="java.util.*" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>count.jsp</title>
</head>
<body>
    This is the checkout counter. Please confirm the following information.
    <p>
        <%
            String no = (String) session.getAttribute("no");
            out.println("Your membership card number is: " + no);

            Enumeration<String> enumGoods = session.getAttributeNames();
            out.println("<br>List of goods in the shopping cart: <br>");
            while (enumGoods.hasMoreElements()) {
                String key = enumGoods.nextElement();
                String good = (String) session.getAttribute(key);

                if (!(good.equals(no))) {
                    out.println("  " + good + "<br>");
                    session.removeAttribute(key);
                }
            }
        %>
    </p>
</body>
```

```
</html>
```

**Food.jsp**

```
<%@ page language="java" import="java.util.*" pageEncoding="UTF-8" session="true"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>food.jsp</title>
</head>
<body>
    <form action="" method="post" name="form">
        This is the seasoning counter. Please select the goods you want to purchase:
        <br/>
        <input type="checkbox" name="food" value="Harmonized Oil">
        Harmonized Oil
        <input type="checkbox" name="food" value="Fortified Salt">
        Fortified Salt
        <input type="checkbox" name="food" value="Multivitamin Vinegar">
        Multivitamin Vinegar
        <input type="checkbox" name="food" value="Green Seasoning">
        Green Seasoning
        <br/>
        <input type="submit" name="submit" value="Shop">
        <p>
            <a href="LoginID.jsp">Welcome to modify the membership card number</a>
            <a href="count.jsp">Welcome to check the shopping cart!</a>
        </p>
        <%
            request.setCharacterEncoding("utf-8");
            String goods[] = request.getParameterValues("food");
            if (goods != null && goods.length != 0) {
                for (int i = 0; i < goods.length; i++) {
                    session.setAttribute(goods[i], goods[i]);
                }
            }
        %>
    </form>
</body>
</html>
```

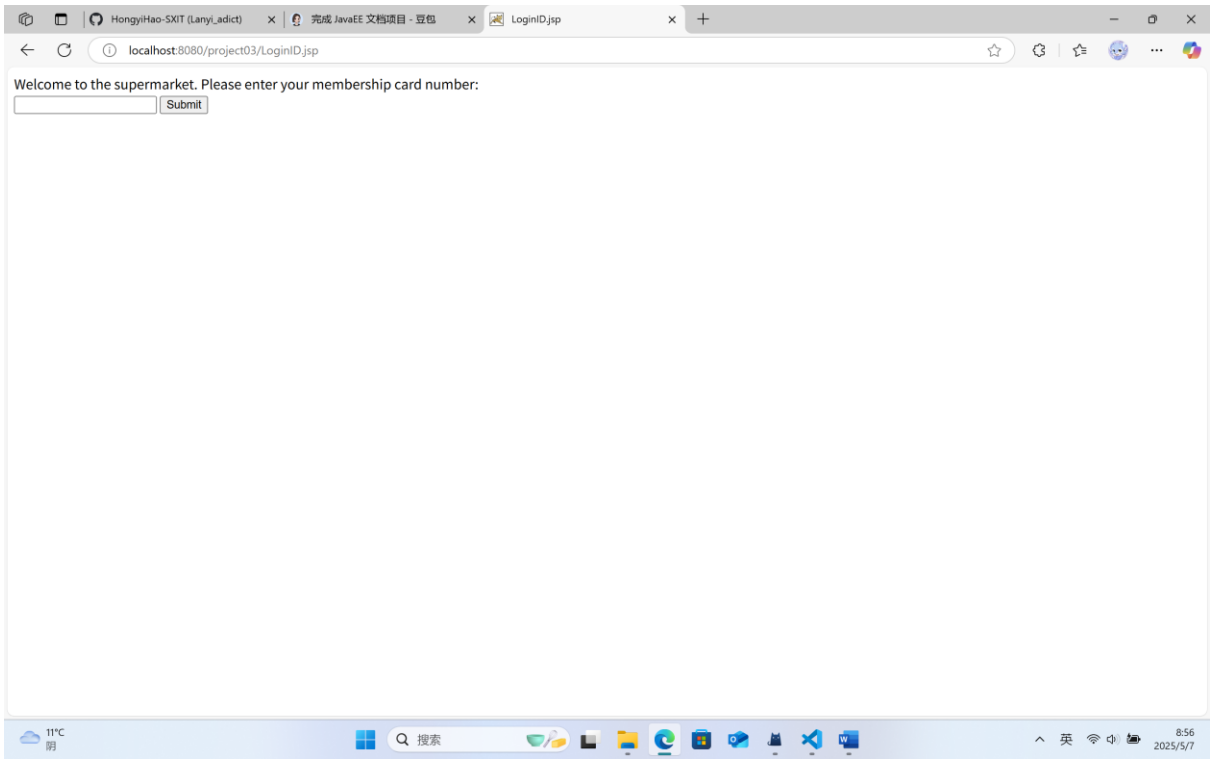
**LoginID.jsp:**

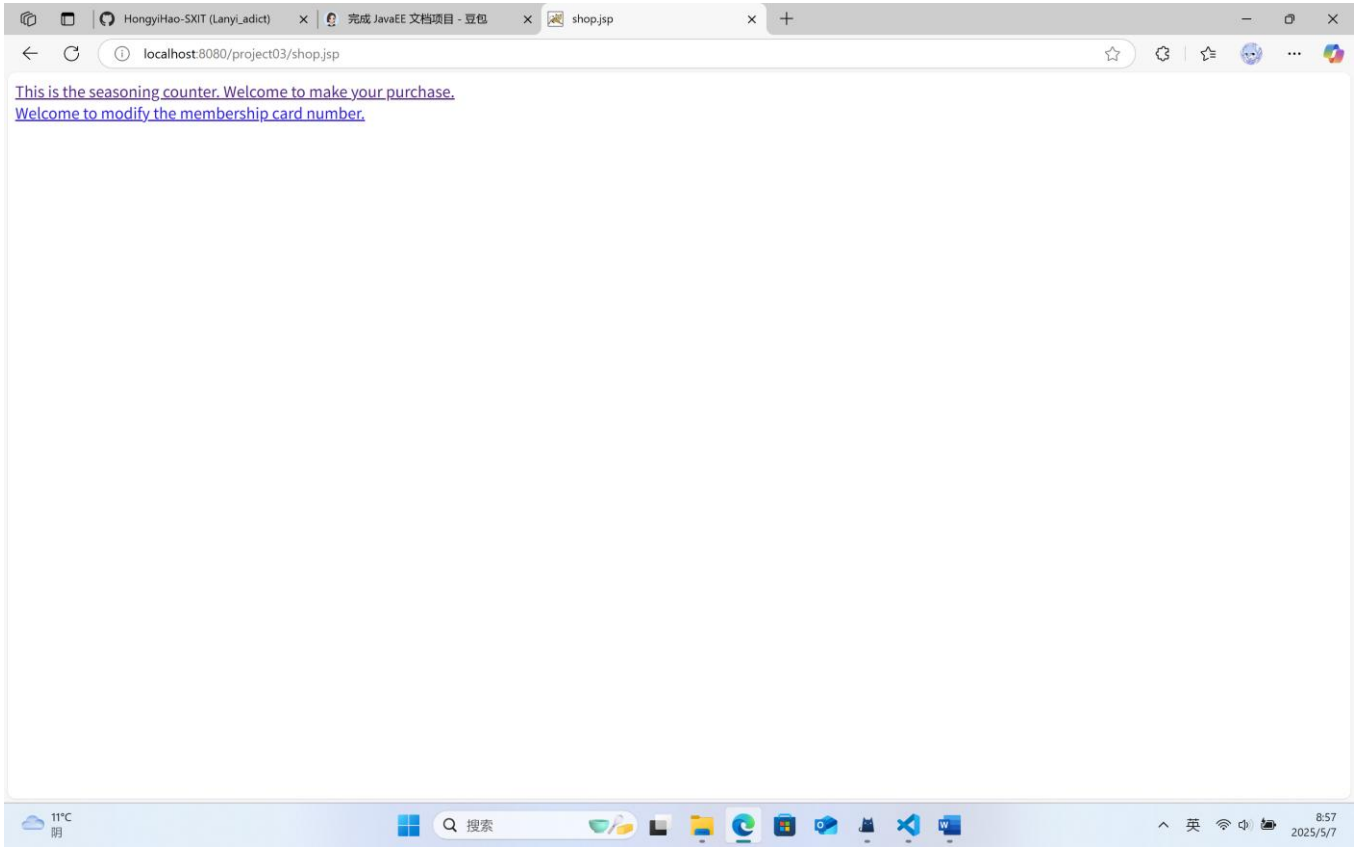
```
<%@ page language="java" import="java.util.*" pageEncoding="gb2312"%>
<!DOCTYPE html>
```

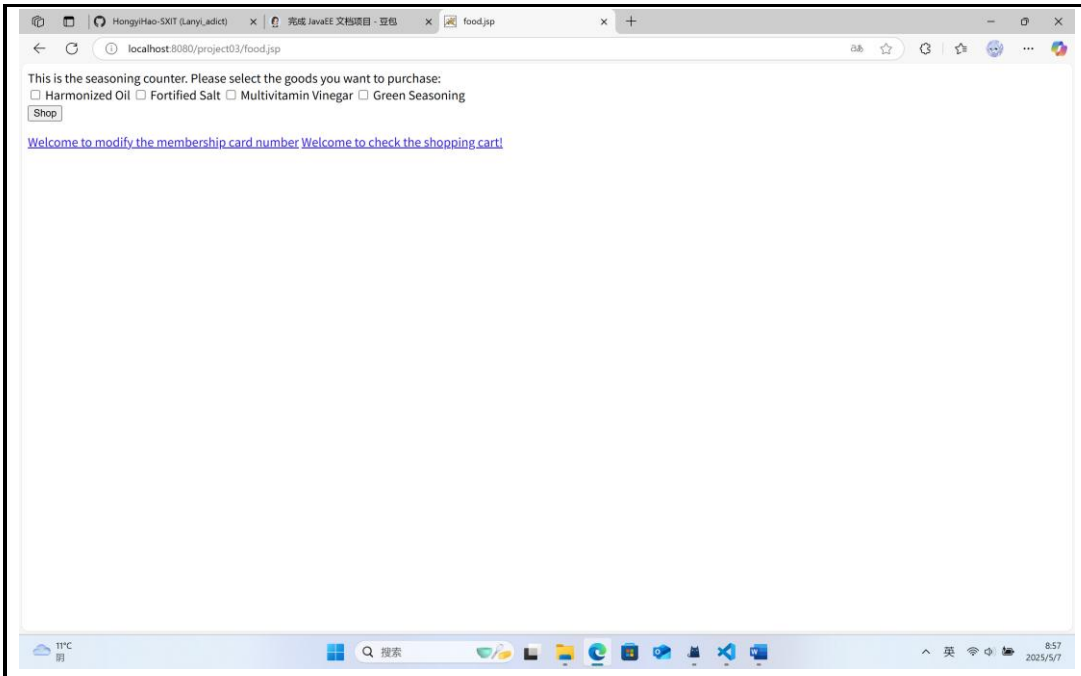
```
<html>
    <head>
        <meta charset="UTF-8">
        <title>LoginID.jsp</title>
    </head>
    <body>
        <form action="shop.jsp" method="post">
            Welcome to the supermarket. Please enter your membership card number:<br>
            <input type="text" name = "no">
            <input type="submit" value="Submit" name="submit">
        </form>
    </body>
</html>
```

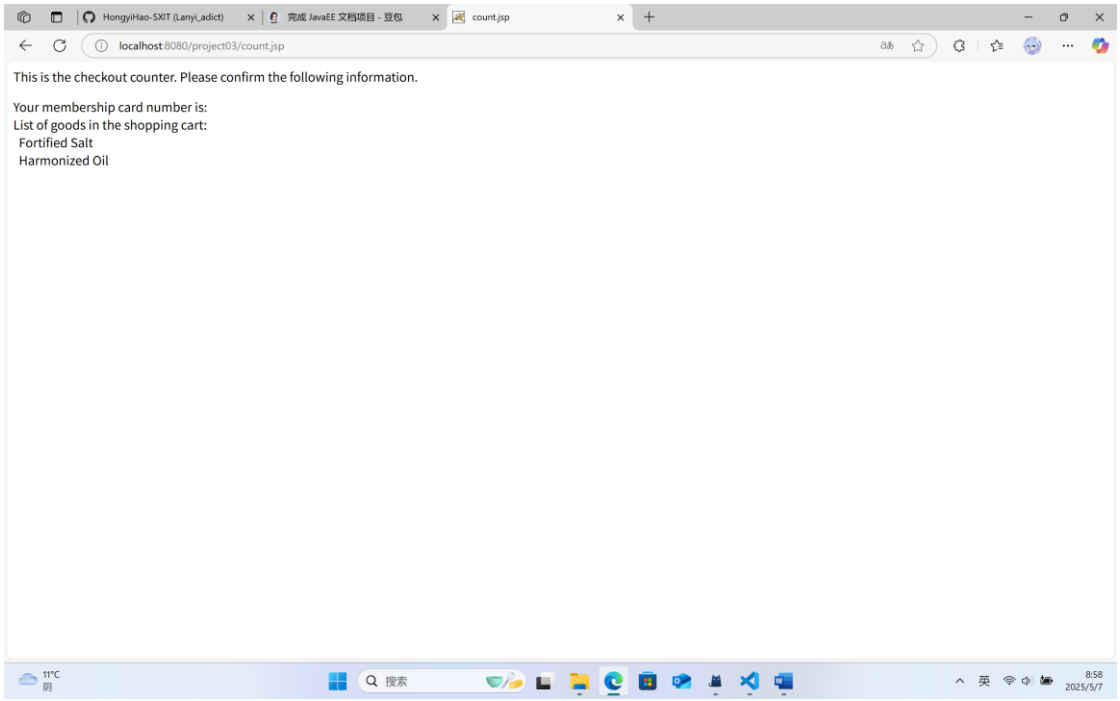
**Shop.jsp:**

```
<%@ page language="java" import="java.util.*" pageEncoding="UTF-8"%>
<html>
<head>
    <meta charset="UTF-8">
    <title>shop.jsp</title>
</head>
<body>
    <%
    String no = request.getParameter("no");
    if(no == null){
        no = "";
    } else{
        session.setAttribute("no", no);
    }%>
    <a href="food.jsp">This is the seasoning counter. Welcome to make your purchase.</a><br>
    <a href="LoginID.jsp">Welcome to modify the membership card number.</a>
</body>
</html>
```


HongyiHao-SXIT (Lanyi_adict)
完成 JavaEE 文档项目 - 豆包
LoginID.jsp
localhost:8080/project03/LoginID.jsp
Welcome to the supermarket. Please enter your membership card number:
Submit
11°C
阴
搜索
英
8:56
2025/5/7


HongyiHao-SXIT (Lanyi_adict)
完成 JavaEE 文档项目 - 豆包
shop.jsp
localhost:8080/project03/shop.jsp
This is the seasoning counter. Welcome to make your purchase.
Welcome to modify the membership card number.
11°C
阴
搜索
英
8:57
2025/5/7




**Countvote.jsp**

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Vote Processing</title>
</head>
<body>
    <%
```

```
        String selectedOption = request.getParameter("option");
        if (selectedOption != null) {
            Integer voteCount = (Integer) application.getAttribute(selectedOption);
            if (voteCount == null) {
                voteCount = 0;
            }
            voteCount++;
            application.setAttribute(selectedOption, voteCount);
            out.println("You have successfully voted for " + selectedOption + "!");
        } else {
            out.println("Please select an option to vote!");
        }
    %>
    <br>
    <a href="vote.jsp">Back to the voting page</a>
    <a href="result.jsp">View the voting results</a>
</body>
</html>
```

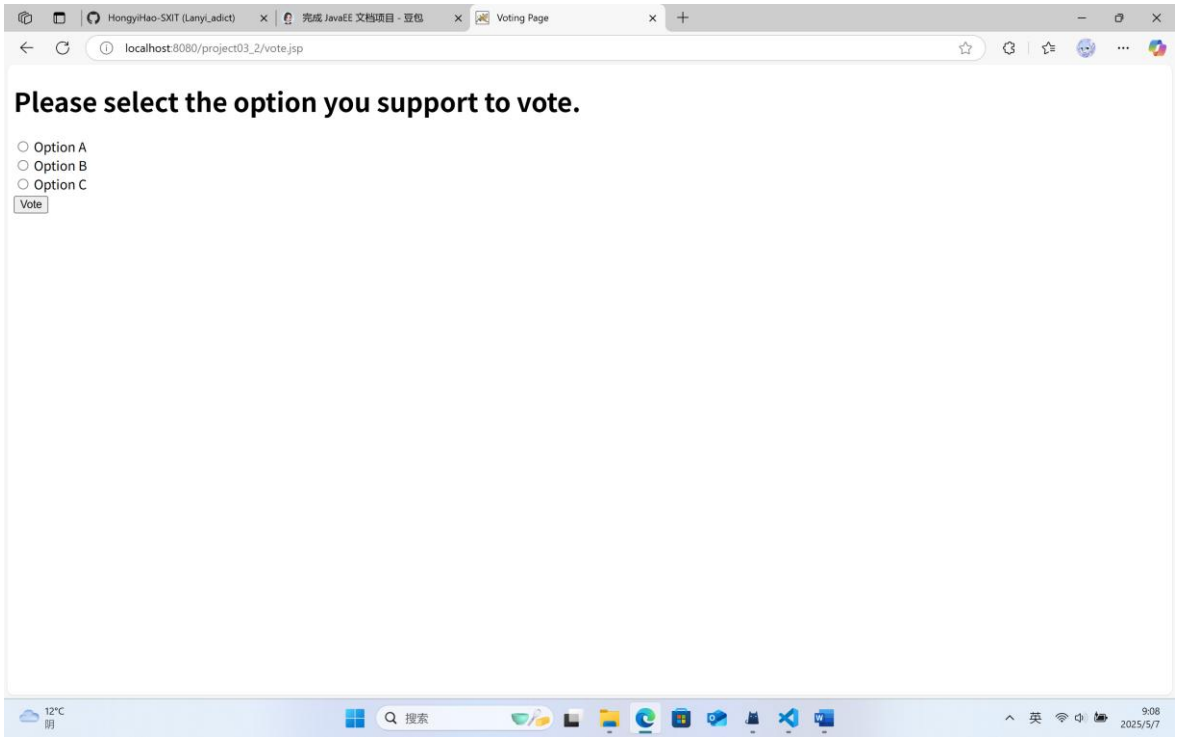
Resule.jsp

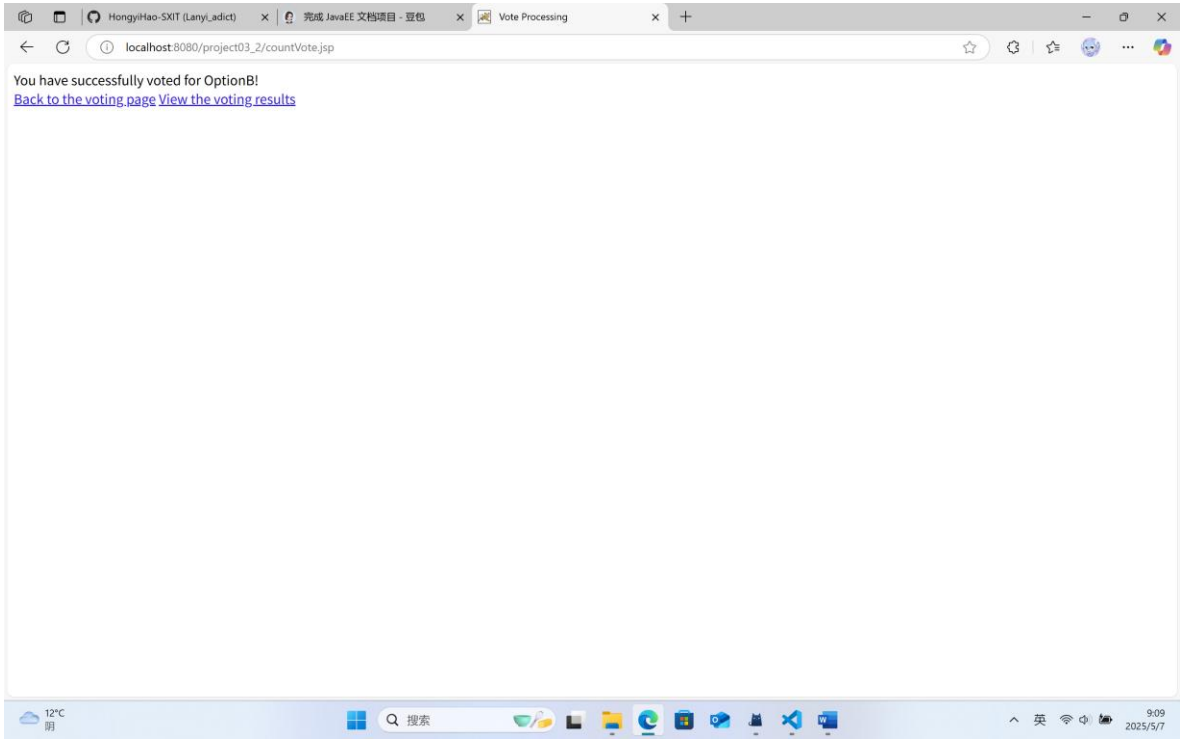
```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Voting Results</title>
</head>
<body>
    <h1>Voting Results</h1>
    <%
        Integer optionAVotes = (Integer) application.getAttribute("OptionA");
        Integer optionBVotes = (Integer) application.getAttribute("OptionB");
        Integer optionCVotes = (Integer) application.getAttribute("OptionC");

        if (optionAVotes == null) {
            optionAVotes = 0;
        }
        if (optionBVotes == null) {
            optionBVotes = 0;
        }
        if (optionCVotes == null) {
            optionCVotes = 0;
        }
```

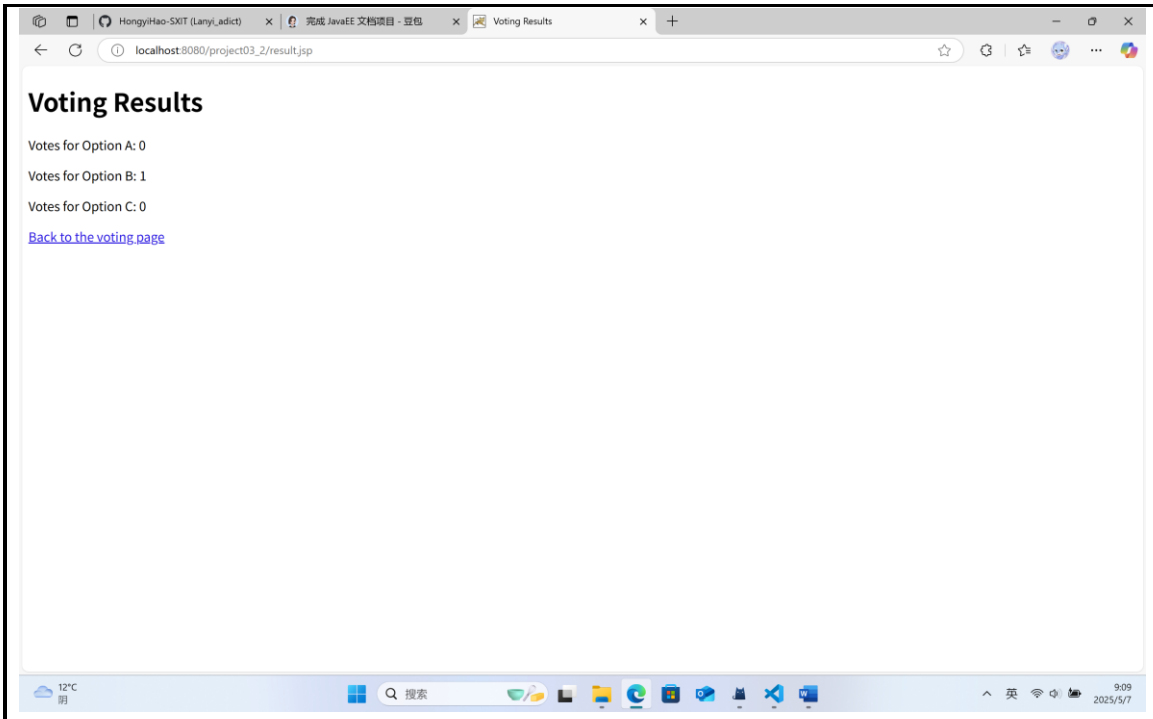
```
    %>
    <p>Votes for Option A: <%= optionAVotes %></p>
    <p>Votes for Option B: <%= optionBVotes %></p>
    <p>Votes for Option C: <%= optionCVotes %></p>
    <a href="vote.jsp">Back to the voting page</a>
</body>
</html>
```

Vote.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Voting Page</title>
</head>
<body>
    <h1>Please select the option you support to vote.</h1>
    <form action="countVote.jsp" method="post">
        <input type="radio" name="option" value="OptionA"> Option A<br>
        <input type="radio" name="option" value="OptionB"> Option B<br>
        <input type="radio" name="option" value="OptionC"> Option C<br>
        <input type="submit" value="Vote">
    </form>
</body>
</html>
```


HongyiHao-SXIT (Lanyi_adict)
完成 JavaEE 文档项目 - 豆包
Voting Page
localhost:8080/project03_2/vote.jsp
Please select the option you support to vote.
Option A
Option B
Option C
Vote
12°C
阴
搜索
英
9:08
2025/5/7


HongyiHao-SXIT (Lanyi_adict)
完成 JavaEE 文档项目 - 豆包
Vote Processing
localhost:8080/project03_2/countVote.jsp
You have successfully voted for OptionB!
Back to the voting page View the voting results
12°C
阴
搜索
英
9:09
2025/5/7




六、实验小结和思考

本实验让我熟悉了 JSP 的核心结构和生命周期，理解了 application 和 session 的作用。通过投票程序和购物车系统的实践，我掌握了用户请求处理及服务器端状态管理方法，同时对 Web 表单处理和内置对象有了更深入的认识。

| 实验成绩 | | 批阅日期 | | 批阅人 | |
|---|---|---|---|---|---|

# 实验报告

| 实验名称 | 练习使用 JavaBean | | | 指导教师 | 王晓霞 |
|---|---|---|---|---|---|
| 实验类型 | 设计型 | 实验学时 | 4 | 实验时间 | 3.26 |

**四、实验目的与要求**

练习在 JSP 中使用 JavaBean。

**二、实验环境**

多媒体计算机 60 台。MySQL8.0 javaweb

**三、实验内容和步骤**

编程实现学生 JavaBean。

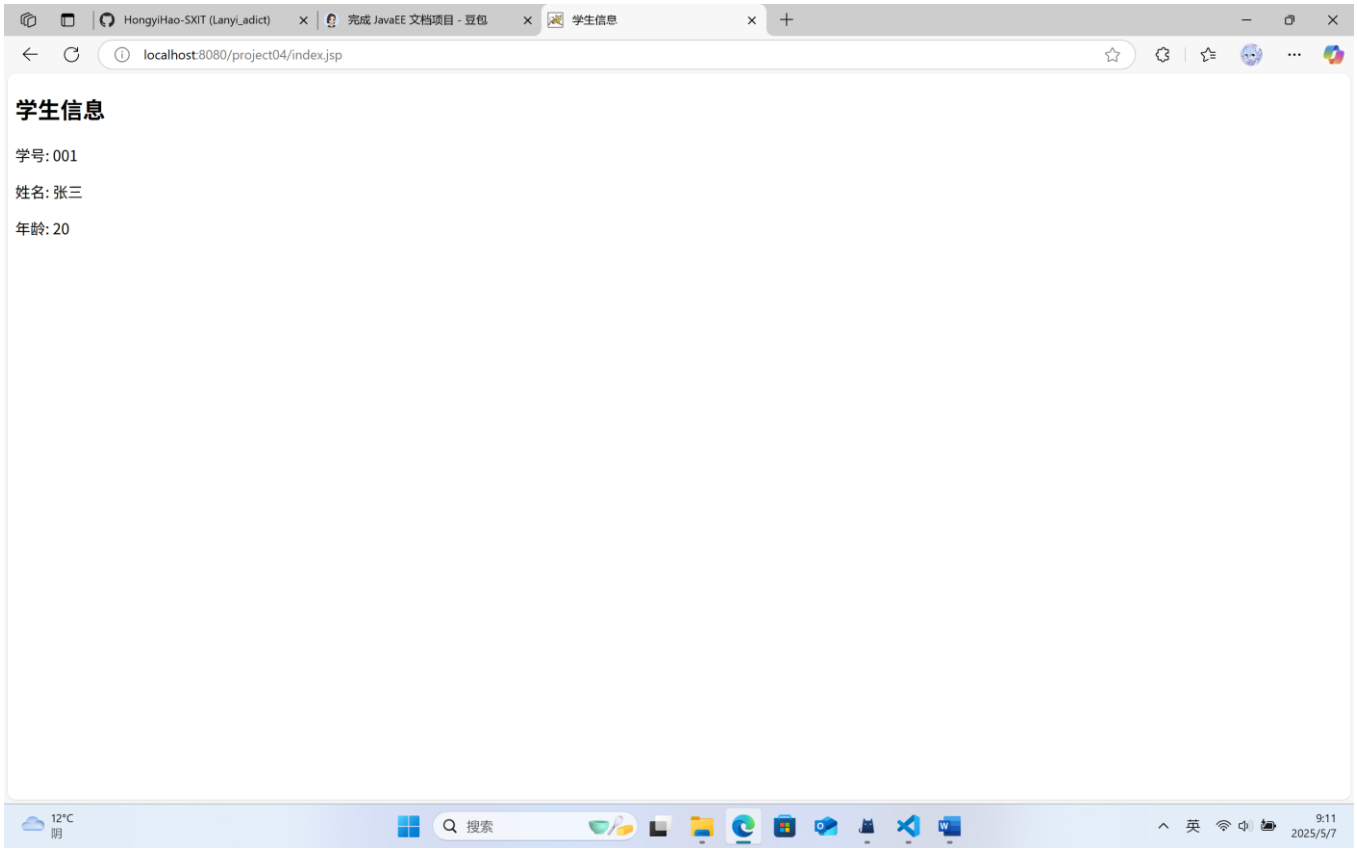
Index.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ page import="com.example.Student" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>学生信息</title>
</head>
<body>
    <jsp:useBean id="student" class="com.example.Student" scope="request">
        <jsp:setProperty name="student" property="id" value="001" />
        <jsp:setProperty name="student" property="name" value="张三" />
        <jsp:setProperty name="student" property="age" value="20" />
    </jsp:useBean>

    <h2>学生信息</h2>
    <p>学号: <jsp:getProperty name="student" property="id" /></p>
    <p>姓名: <jsp:getProperty name="student" property="name" /></p>
    <p>年龄: <jsp:getProperty name="student" property="age" /></p>
</body>
</html>
```

Student.java

```
package com.example;
```

```
public class Student {
    private String id;
    private String name;
    private int age;

    public Student() {
    }

    public Student(String id, String name, int age) {
        this.id = id;
        this.name = name;
        this.age = age;
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }
}
```




七、实验小结和思考

本次实验通过创建 Student JavaBean 并在 JSP 页面中通过 <jsp:useBean> 标签引用，实现了数据的封装与显示，提升了代码的可维护性与重用性。使用 getProperty 和 setProperty 方法后，JSP 页面变得更简洁，Java 逻辑也更清晰。通过本实验我进一步理解了 JavaBean 在 MVC 模式中作为模型的作用，并掌握了 JSP 与 JavaBean 的基本结合方法。

| 实验成绩 | | 批阅日期 | | 批阅人 | |
|---|---|---|---|---|---|

# 实验报告

| 实验名称 | Servlet 的使用 | | | 指导教师 | 王晓霞 |
|---|---|---|---|---|---|
| 实验类型 | 设计型 | 实验学时 | 4 | 实验时间 | 4.9 |

**五、实验目的与要求**

学会 Servlet 的开发方法、Servlet 的生命周期以及在 Servlet 中如何使用 JSP 页面中常用的内置对象等内容。

**二、实验环境**

多媒体计算机 60 台。MySQL8.0 javaweb

三、实验内容和步骤

练习 Servlet 的开发方法。

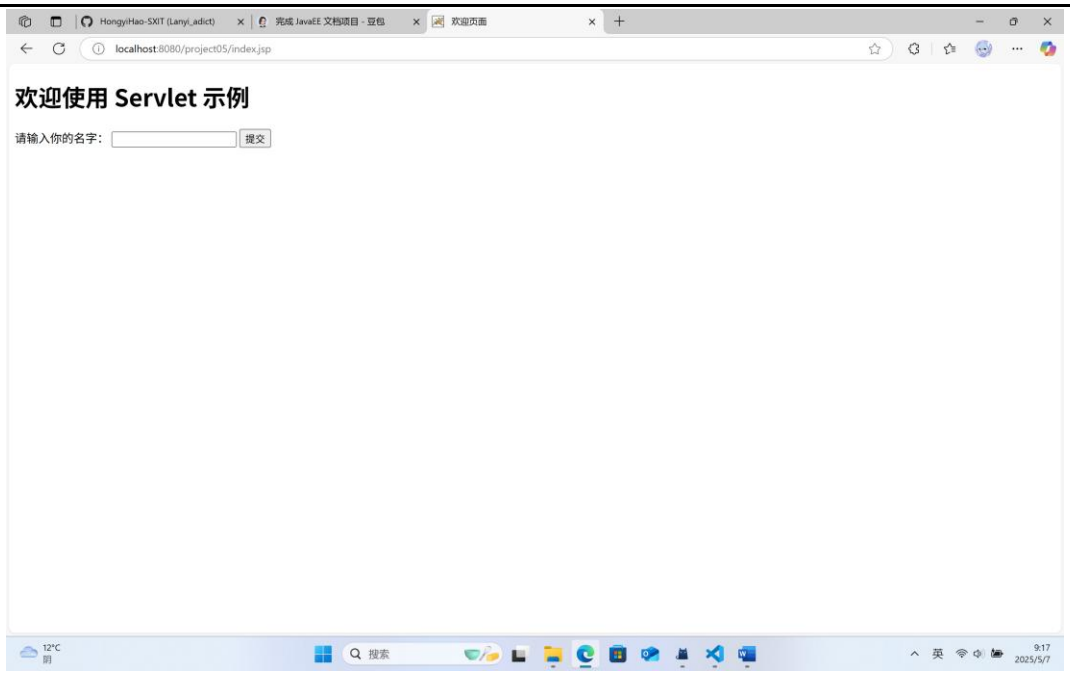
Index.jsp

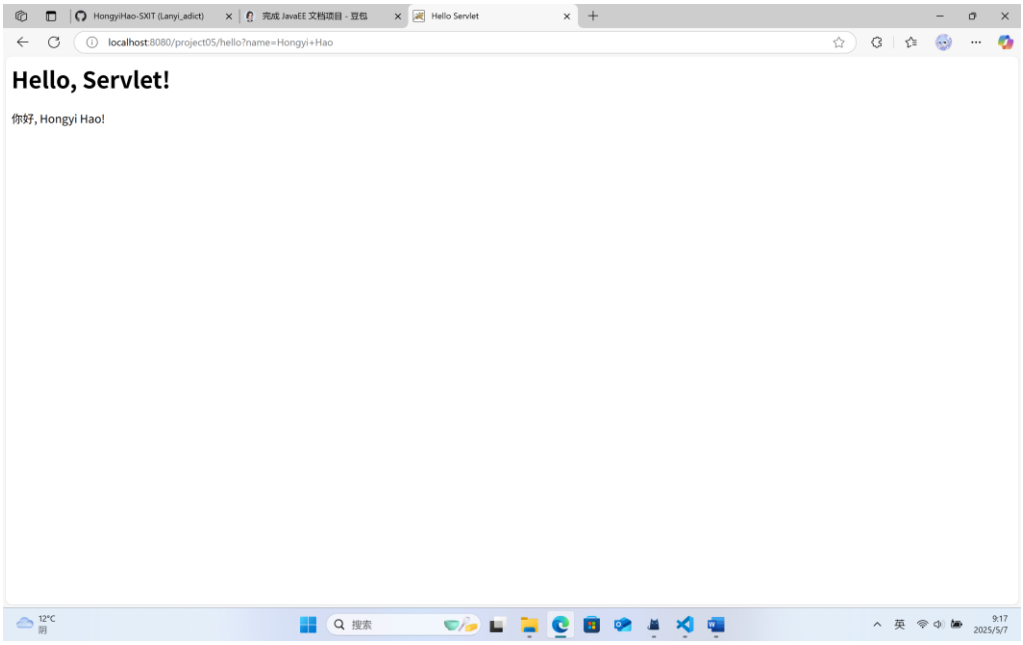
```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>欢迎页面</title>
</head>
<body>
    <h1>欢迎使用 Servlet 示例</h1>
    <form action="hello" method="get">
        <label for="name">请输入你的名字：</label>
        <input type="text" id="name" name="name">
        <input type="submit" value="提交">
    </form>
</body>
</html>
```

HelloServlet.java

```
package com.example;

import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

import java.io.IOException;
```

```
import java.io.PrintWriter;

@WebServlet("/hello")
public class HelloServlet extends HttpServlet {

    @Override
    public void init() throws ServletException {
        super.init();
        System.out.println("HelloServlet 初始化");
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try {
            out.println("<html>");
            out.println("<head><title>Hello Servlet</title></head>");
            out.println("<body>");
            out.println("<h1>Hello, Servlet!</h1>");

            String name = request.getParameter("name");
            if (name != null && !name.isEmpty()) {
                out.println("<p>你好, " + name + "!</p>");
            }
            out.println("</body>");
            out.println("</html>");
        } finally {
            out.close();
             }
         }
         @Override
         protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

             doGet(request, response);
         }
         @Override
         public void destroy() {
             super.destroy();
             System.out.println("HelloServlet 销毁");
         }
     }
```




八、实验小结和思考

通过本次 Servlet 实验，我掌握了 Servlet 的基本开发流程、生命周期方法（init()、doGet()、destroy()）以及在 Servlet 中获取请求参数、输出响应的基本技巧。配合 JSP 页面进行表单提交，让我更深入理解了前后端交互过程。通过亲自配置 Web 应用和输出 HTML，我对 Web 服务处理流程有了更直观的理解，为后续 MVC 架构开发打下基础。

| 实验成绩 | | 批阅日期 | | 批阅人 | |
|---|---|---|---|---|---|

# 实验报告

| 实验名称 | SSH 项目训练 | | | 指导教师 | 王晓霞 |
|---|---|---|---|---|---|
| 实验类型 | 设计型 | 实验学时 | 4 | 实验时间 | 4.23 |

**六、实验目的与要求**

学会用 SSH 进行项目开发。

**二、实验环境**

多媒体计算机 60 台。MySQL8.0 javaweb

**三、实验内容和步骤**

使用 Struts、Spring、Hibernate 进行综合项目开发。

**User.java**

```
public class User {
    private int id;
    private String userName;
    private String blogUrl;
    private int sex;
    private String provinceName;
    private String hobby;
    private String remark;

    public User() {}

    public int getId() { return id; }
    public void setId(int id) { this.id = id; }

    public String getUserName() { return userName; }
    public void setUserName(String userName) { this.userName = userName; }

    public String getBlogUrl() { return blogUrl; }
    public void setBlogUrl(String blogUrl) { this.blogUrl = blogUrl; }

    public int getSex() { return sex; }
    public void setSex(int sex) { this.sex = sex; }

    public String getProvinceName() { return provinceName; }
    public void setProvinceName(String provinceName) { this.provinceName = provinceName; }
```

```
    public String getHobby() { return hobby; }
    public void setHobby(String hobby) { this.hobby = hobby; }

    public String getRemark() { return remark; }
    public void setRemark(String remark) { this.remark = remark; }
}
```

**Database.sql**

```
DROP TABLE IF EXISTS tb_user;

CREATE TABLE tb_user (
    id INT AUTO_INCREMENT PRIMARY KEY COMMENT '用户编号',
    user_name VARCHAR(50) NOT NULL COMMENT '用户姓名',
    blog_url VARCHAR(50) NOT NULL COMMENT '博客地址',
    sex CHAR(2) DEFAULT '2' COMMENT '性别（1：男；2：女；）',
    province_name VARCHAR(20) COMMENT '省份',
    hobby VARCHAR(50) COMMENT '兴趣爱好',
    remark VARCHAR(50) COMMENT '备注'
) COMMENT='用户信息表';
```

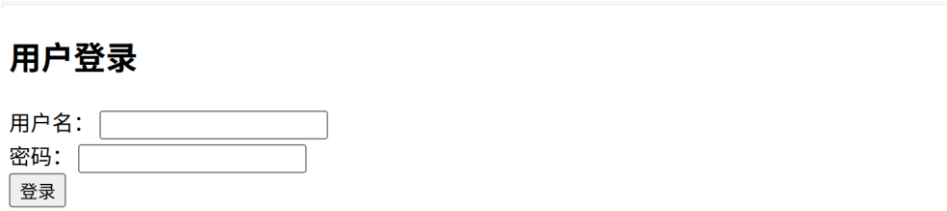
**Login.jsp**

```
<%@ page contentType="text/html;charset=UTF-8" %>
<html>
<head><title>用户登录</title></head>
<body>
<h2>登录</h2>
<form action="login.action" method="post">
    用户名：<input type="text" name="userName"/><br/>
    博客地址：<input type="text" name="blogUrl"/><br/>
    <input type="submit" value="登录"/>
</form>
</body>
</html>
```

**LoginAction.java:**

```
import com.opensymphony.xwork2.ActionSupport;
import lombok.Data;

@Data
public class LoginAction extends ActionSupport {
    private String userName;
```

```
    private String blogUrl;

    public String execute() {
        if ("admin".equals(userName) && "http://blog.example.com".equals(blogUrl)) {
            return SUCCESS;
        } else {
            addActionError("用户名或博客地址不正确！");
            return INPUT;
        }
    }
}
```




九、实验小结和思考

本次实验让我初步体验了 **Struts**、**Spring** 和 **Hibernate** 三大主流框架的整合开发。通过搭建用户登录功能，理解了 **Struts** 的控制流程、**Spring** 的依赖注入机制和 **Hibernate** 的数据持久化功能。通过编写实体类 **User**、配置数据库、前端页面与 **Action** 类的交互，我认识到企业级开发的复杂性与框架优势。本实验提升了我对 **Java EE** 分层开发思想的理解，也让我具备了进行更复杂 **Web** 应用开发的能力。

| 实验成绩 | | 批阅日期 | | 批阅人 | |
|---|---|---|---|---|---|