

# 专 周 实 训（课程设计） 报 告

姓 名：\_\_\_\_\_

学 号：\_\_\_\_\_

系 别：\_\_\_\_\_信息工程学院\_\_\_\_\_

专 业：\_\_\_\_\_软件技术\_\_\_\_\_

年 级：\_\_\_\_\_

班 级：\_\_\_\_\_24 软件班\_\_\_\_\_

指导教师：\_\_\_\_\_李 唐\_\_\_\_\_

2025 年 12 月 15 日 至 2025 年 12 月 26 日

# 专 周 实 训（课程设计） 计 划 书

所 在 单 位：      24 级 信息工程学院 系 软件技术专业 专业

专周实训（课程设计）题目：学生管理系统

专周实训（课程设计）目的：

- 1、掌握 SSM 项目开发的软件，比如 MyEclipse、IDEA 等，能够配置搭建 SSM 项目的开发环境；
- 2、掌握企业项目实际的开发流程，包括需求分析、概要设计、详细设计、项目迭代开发等；
- 3、掌握数据库的设计与实现；
- 4、掌握企业项目迭代开发的思想，能够通过项目需求分解出独立子功能；
- 5、掌握使用 SSM 技术完成项目每个子功能的开发实现；
- 6、掌握综合案例开发，能够利用综合案例开发掌握 SSM 发项目技术的综合应用；
- 7、按照实训文档的编写要求完成报告；

专周实训（课程设计）内容要求：

- 1、学生独立完成 SSM 小型项目的需求分析、概要设计、详细设计等项目开发流程环节。
- 2、学生独立搭建 SSM 开发项目的开发环境。
- 3、学生根据项目的详细设计独立完成 SSM 技术的中小型的 Web 项目功能的开发实现。
- 4、学生在项目开发过程中独立思考，完成对所学知识整理、概括、消化吸收，以及围绕课堂教学内容，阅读参考书籍和资料，完成扩展知识的学习。
- 5、学生通过过程文档、实验报告等清晰地表达自己解决问题的思路和步骤。
- 6、学生独立思考、深入钻研问题，针对各种常见需求项目完成提出多种解决方案， 并分析设计项目开发模型，以及进行优化设计。

---

**专周实训（课程设计）时间安排：**

2025 年 12 月 16 日：搭建 SSM 开发环境（包括开发环境和数据库数据准备）

2025 年 12 月 22 日：完成用户登录（包括登录功能、登录权限验证）、用户注册、用户退出

2025 年 12 月 23 日：完成列表的展示（包括列表展示、分页功能、条件查询）

2025 年 12 月 24 日：完成添加、修改、删除功能

2025 年 12 月 25 日：完成模板下载功能、导入导出功能

2025 年 12 月 26 日：完成实训报告的撰写和提交

## 专周实训（课程设计）报告：（可自行增加页面）

### 1、学生管理系统功能的简单描述和功能框架图

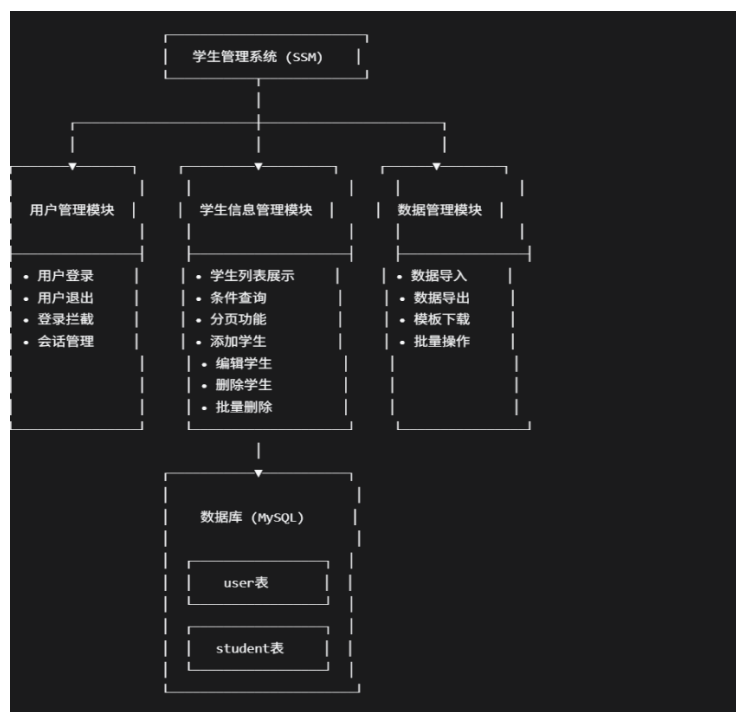
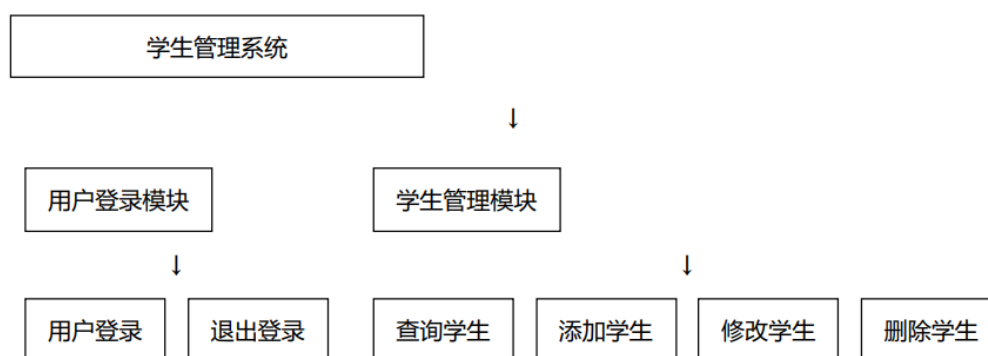
客户管理系统主要帮助公司完成客户信息的基本管理，包括查询数据，增加数据，修改数据，删除数据以及对应的导入导出数据。本系统中主要实现了两大功能模块：用户登录模块和客户管理模块，该系统的功能框架图如下：

#### 学生管理系统数据库设计

##### 1. 系统功能简介

学生管理系统主要实现学生信息的管理和用户权限控制，包含两大核心模块：用户登录模块和学生管理模块。

##### 2. 功能框架图



本系统采用经典的 SSM（Spring + SpringMVC + MyBatis）框架开发，主要分为三大功能模块。用户管理模块负责系统访问控制，包括登录验证、会话管理和权限拦截；学生信息管理模块是核心业务模块，实现了对学生信

息的全方位管理；数据管理模块提供数据的批量处理能力。所有模块通过统一的 Controller 层接收请求，Service 层处理业务逻辑，DAO 层操作数据库，最终将结果返回给前端 JSP 页面展示。

2、学生管理系统的数据库表的设计

1. 用户表 (user)

表名: user

字段名	数据类型	描述	约束
id	INT	用户ID	主键, 自增
username	VARCHAR	用户名	唯一, 登录账号
password	VARCHAR	密码	加密存储
real_name	VARCHAR	真实姓名	-
role	VARCHAR	角色	如: admin (管理员)、teacher (教师)
create_time	DATETIME	创建时间	-
update_time	DATETIME	更新时间	-

**作用：**用于存储系统登录用户的账户信息，实现身份验证与权限控制。

**关键字段说明：**

**id：**主键，自增，唯一标识用户记录。

**username：**唯一索引，作为登录账号，不可重复。

**password：**加密存储用户密码，增强安全性。

**role：**区分用户角色（如 admin、teacher），用于权限控制。

**create\_time 与 update\_time：**记录用户创建与信息更新时间，便于审计与追踪。

## 2. 学生表 (student)

表名: student

字段名	数据类型	描述	约束
id	INT	学生ID	主键, 自增
student_id	VARCHAR	学号	唯一, 学生编号
name	VARCHAR	姓名	-
gender	VARCHAR	性别	如: 男、女
age	INT	年龄	-
birthday	DATE	出生日期	-
major	VARCHAR	专业	-
class_name	VARCHAR	班级	-
phone	VARCHAR	手机号	-
email	VARCHAR	邮箱	-
address	VARCHAR	地址	-
create_time	DATETIME	创建时间	-
update_time	DATETIME	更新时间	-

作用: 存储学生基本信息, 是系统的核心数据实体, 支撑学生管理所有业务功能。

关键字段说明:

id: 主键, 自增, 唯一标识学生记录。

student\_id: 学号, 具有唯一性, 是学生在系统中的业务标识。

name、gender、age、birthday: 记录学生基本个人信息。

major 与 class\_name: 记录学生所属专业与班级, 支持按专业、班级查询。

phone、email、address: 记录学生联系方式与住址信息。

create\_time 与 update\_time: 记录学生信息创建与最后更新时间。

## 3、学生管理系统功能详细设计的描述和对应的流程图

学生管理系统手动启动流程

简化版流程

- 1. 准备环境
  - ↓
- 2. 进入项目文件夹
  - ↓
- 3. 下载项目需要的工具包
  - ↓
- 4. 整理项目文件
  - ↓
- 5. 检查8080端口是否被占用
  - ↓
- 6. 如果端口被占用，关闭占用的程序
  - ↓
- 7. 启动服务器
  - ↓
- 8. 在浏览器中打开 http://localhost:8080
  - ↓
- 9. 使用系统
  - ↓
- 10. 关闭服务器

详细解释

1. 准备环境

- 确保电脑上安装了Java和Maven
- 不需要记命令，能正常使用即可

2. 进入项目文件夹

file:///D:/冯智灵/Documents/BEIFENG/OneDrive/Desktop/Java实训/学生管理系统启动流程.html 1/3

2025/12/23 14:34

学生管理系统手动启动流程

- 打开文件资源管理器，找到项目存放的位置
- 示例：D:\冯智灵\Documents\BEIFENG\OneDrive\Desktop\Java实训

3. 下载项目需要的工具包

- 在项目文件夹中打开终端（命令行）
- 输入：mvn dependency:resolve 并回车
- 等待下载完成（看到“BUILD SUCCESS”就可以了）

4. 整理项目文件

- 输入：mvn compile 并回车
- 等待整理完成（同样看到“BUILD SUCCESS”）

5. 检查8080端口是否被占用

- 输入：netstat -ano | findstr :8080 并回车
- 如果有很多文字出现，说明端口被占用了

6. 如果端口被占用，关闭占用的程序

- 输入：taskkill /F /PID [数字] 并回车
- 把[数字]换成上一步看到的最后一列数字（进程ID）

7. 启动服务器

- 输入：mvn tomcat7:run 并回车
- 等待服务器启动（看到“Starting Tomcat”相关信息）

file:///D:/冯智灵/Documents/BEIFENG/OneDrive/Desktop/Java实训/学生管理系统启动流程.html 2/3

2025/12/23 14:35

学生管理系统手动启动流程

8. 在浏览器中打开

- 打开任何一个浏览器（Chrome、Edge等）
- 在地址栏输入：http://localhost:8080 并回车
- 就能看到登录页面了

9. 使用系统

- 输入用户名和密码登录
- 使用学生管理系统的各种功能

10. 关闭服务器

- 在启动服务器的终端窗口中
- 按下键盘上的 Ctrl + C 组合键
- 输入 Y 并回车确认关闭

注意事项

- 所有命令都在项目文件夹中执行
- 启动服务器后，不要关闭终端窗口
- 如果遇到问题，可以重新按照步骤来一遍
- 如果还是不行，可以查看终端窗口里的提示信息，大部分问题都有说明

4、学生管理系统功能模块实现的文字描述、界面效果截图和对应的源代码截图

## 一、登录页面代码

主要功能：

提供用户名和密码输入框

包含登录按钮

显示登录错误信息

使用 POST 方法提交到 /user/login 接口

```
1 <form action="/user/login" method="post">
2     <c:if test="${not empty error}">
3         <div class="error-message">${error}</div>
4     </c:if>
5     <div class="form-group">
6         <label for="username">用户名: </label>
7         <input type="text" id="username" name="username" required>
8     </div>
9     <div class="form-group">
10        <label for="password">密码: </label>
11        <input type="password" id="password" name="password" required>
12    </div>
13    <input type="submit" value="登录">
14 </form>
```

## 二、登录功能实现

主要功能：

处理登录请求

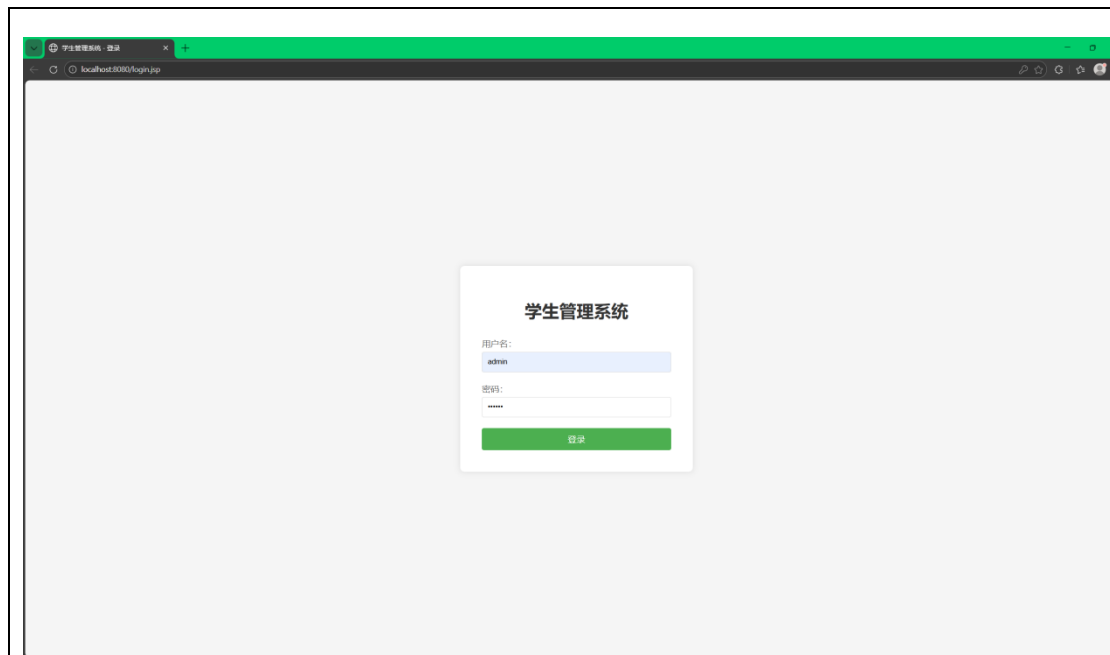
验证用户名和密码

管理用户会话

实现退出功能

```
Java
1 // 登录功能
2 @RequestMapping(value = "/login", method = RequestMethod.POST)
3 public String login(String username, String password, Model model,
4 HttpSession session) {
5     User user = userService.login(username, password);
6     if (user != null) {
7         // 登录成功，将用户信息存入session
8         session.setAttribute("user", user);
9         // 重定向到学生列表页面
10        return "redirect:/student/list";
11    } else {
12        // 登录失败，返回登录页面并显示错误信息
13        model.addAttribute("error", "用户名或密码错误");
14        return "login";
15    }
16 }
17 // 退出功能
18 @RequestMapping(value = "/logout", method = RequestMethod.GET)
19 public String logout(HttpSession session) {
20     // 清除session中的用户信息
21     session.invalidate();
22     // 重定向到登录页面
23     return "redirect:/login.jsp";
24 }
```





### 三、学生管理页面

#### 1. 学生列表页面（主管理页面）

核心功能：

学生信息展示（表格形式）

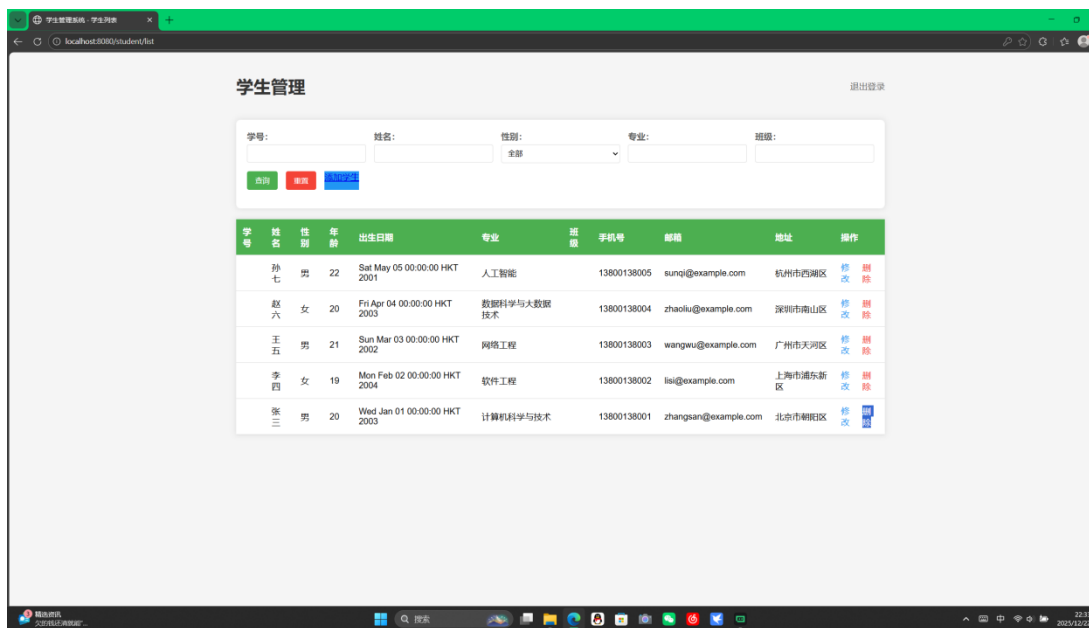
多条件查询功能（学号、姓名、性别、专业、班级）

添加学生入口

修改/删除学生操作

分页功能

```
jsp
4      <!-- 多条件查询字段 -->
5      <div class="search-row">
6          <div class="form-group">
7              <label for="studentId">学号: </label>
8              <input type="text" id="studentId" name="studentId" value="${student.studentId}">
9          </div>
10         <!-- 其他查询字段 -->
11     </div>
12     <div class="search-row">
13         <button type="submit">查询</button>
14         <button type="button" class="reset-btn" onclick="resetForm()">重置</button>
15         <a href="/student/add" class="add-btn">添加学生</a>
16     </div>
17 </form>
18 </div>
19
20 <!-- 学生列表表格 -->
21 <table>
22     <tr>
23         <th>学号</th>
24         <th>姓名</th>
25         <th>性别</th>
26         <!-- 其他表头 -->
27         <th>操作</th>
28     </tr>
29     <c:forEach items="${students}" var="student">
30         <tr>
31             <td>${student.studentId}</td>
32             <td>${student.name}</td>
33             <td>${student.gender}</td>
34             <!-- 其他数据列 -->
35             <td>
36                 <div class="action-links">
37                     <a href="/student/edit?id=${student.id}" class="edit-link">修改</a>
38                     <a href="/student/delete?id=${student.id}" class="delete-link" onclick="return confirm('确定要删除该学生吗?')">删除</a>
39                 </div>
40             </td>
41         </tr>
42     </c:forEach>
43 </table>
```



## 2. 添加学生页面

核心功能：

学生信息录入表单

表单验证（必填字段）

错误信息展示

```

1 <div class="add-form">
2   <c:if test="{not empty error}">
3     <div class="error-message">${error}</div>
4   </c:if>
5   <form action="/student/add" method="post">
6     <div class="form-row">
7       <div class="form-group">
8         <label for="studentId">学号: </label>
9         <input type="text" id="studentId" name="studentId" required>
10      </div>
11      <div class="form-group">
12        <label for="name">姓名: </label>
13        <input type="text" id="name" name="name" required>
14      </div>
15    </div>
16    <!-- 其他表单字段 -->
17    <div class="form-actions">
18      <button type="submit" class="submit-btn">添加</button>
19      <button type="button" class="cancel-btn" onclick="window.location.
20        href='/student/list'">取消</button>
21    </div>
22  </form>
23 </div>

```

### 3. 编辑学生页面

核心功能：

学生信息修改表单

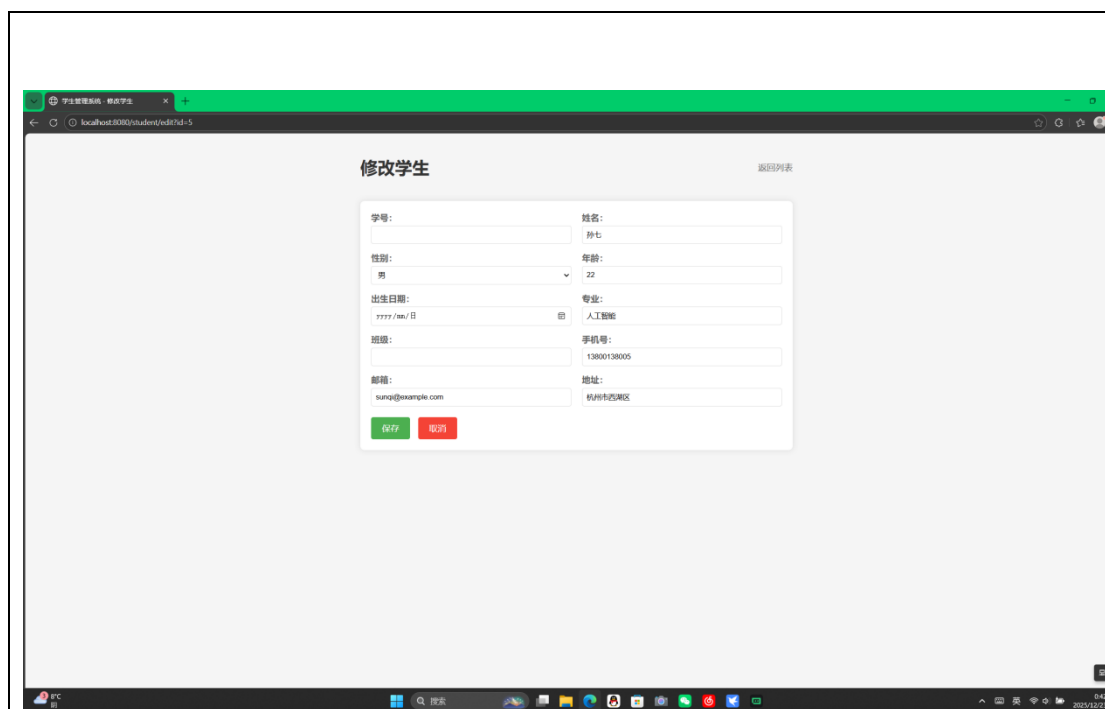
表单数据预填充

错误信息展示

```

1 <div class="edit-form">
2   <c:if test="{not empty error}">
3     <div class="error-message">${error}</div>
4   </c:if>
5   <form action="/student/edit" method="post">
6     <input type="hidden" name="id" value="{student.id}">
7     <div class="form-row">
8       <div class="form-group">
9         <label for="studentId">学号: </label>
10        <input type="text" id="studentId" name="studentId" value="{
11          {student.studentId}" required>
12      </div>
13      <!-- 其他表单字段 -->
14    </div>
15    <div class="form-actions">
16      <button type="submit" class="submit-btn">保存</button>
17      <button type="button" class="cancel-btn" onclick="window.location.
18        href='/student/list'">取消</button>
19    </div>
20  </form>
21 </div>

```



### 学生总结：

通过为期两周的《JavaWeb 开发实战》专周实训，我独立完成了基于 SSM（Spring + SpringMVC + MyBatis）框架的“学生管理系统”开发任务，深刻体会了企业级 Web 项目的完整开发流程。

本次实训中，我首先掌握了 IDEA 开发环境中 SSM 框架的整合配置，独立完成了项目初始化、数据库连接池（Druid）及 MyBatis 的配置。在功能开发层面，我实现了用户登录、会话管理与拦截器验证，确保了系统的安全性。核心的学生信息管理模块包含了列表分页展示、多条件组合查询、以及完整的增删改查（CRUD）操作。此外，我还完成了数据的 Excel 导入与导出功能，利用 Apache POI 库处理文件读写，增强了系统的实用性。

在实践过程中，我遇到了不少挑战。例如，在实现动态 SQL 条件查询时，需仔细设计 MyBatis 的 Mapper 映射文件；在处理文件上传与数据批量导入时，需要兼顾功能实现与异常处理。通过查阅文档、调试代码，我不仅解决了这些问题，更提升了自己独立分析问题和 Debug 的能力。我也认识到自己在代码结构优化与异常处理的严谨性上仍有提升空间。

总而言之，本次实训将课堂所学的分散知识点（如 Spring 的 IOC/AOP、MVC 模式、MyBatis 的 ORM 映射）串联成一个完整的项目，极大地锻炼了我的工程实践能力、文档撰写能力和系统性思维。感谢李唐老师的悉心指导，我将继续夯实基础，努力成为一名合格的软件技术人才。

---

**指导教师评定结论：**

该生在本次 JavaWeb 实训中态度认真，顺利完成了学生管理系统的开发任务。对课堂所学知识能够较好地运用，编程能力有了较大提高。希望后续能继续保持学习态度，并在代码细节上多加打磨。

**专周实训（课程设计）总分：**

注：过程性成绩为百分制，录入教务系统时折算为五级制。

指导教师签字（签章）：