

Web应用开发课程大作业要求

1. 项目概述

本项目旨在综合应用Web应用开发课程所学知识，培养学生的实际开发能力和团队协作精神。学生将以小组为单位，设计并实现一个基于JSP的网站系统。

1.1 课程目标

1. 掌握Web开发和运行平台的搭建，熟悉HTML、JavaScript和CSS等静态网页开发技术和以JSP2.0为主的动态网页的开发技术。
2. 培养动态网页设计的基本技能，熟练利用JSP技术进行中等难度的动态网页编程，并能自主探索新的网站开发技术以提升网站稳定性。
3. 掌握JSP技术、Servlet技术、JavaBean技术、JDBC技术和EL、JSTL，并基于MVC模式进行Web应用的开发。
4. 了解并掌握开发网站过程中涉及的法律法规和职业行为准则。

1.2 项目重点

- 应用B/S架构设计Web系统
- 综合运用JSP、Servlet、JavaBean等技术
- 实现数据库的连接与使用
- 应用MVC设计模式
- 考虑系统的可维护性、可靠性、可理解性和效率

2. 项目要求

2.1 基本要求

- 使用JSP技术开发动态网页
- 采用B/S架构
- 使用MySQL数据库进行数据存储和管理
- 应用MVC(模型-视图-控制)架构模式
- 系统应具有良好的可维护性、可靠性、可理解性和效率
- 用户界面应直观易用

2.2 技术要求

- 前端：
 - HTML5, CSS3, JavaScript
 - 允许使用前端框架如Bootstrap或Foundation
- 后端：
 - JSP, Servlet
 - 鼓励使用其他web开发后端框架（Spring, MyBatis）
 - JavaBean技术
- 数据库：MySQL，使用JDBC技术
- 服务器：Tomcat
- 开发工具：Eclipse, IntelliJ IDEA等
- 版本控制：建议使用Git

2.3 功能要求

1. 用户认证和授权
 - 实现用户注册、登录、注销功能
 - 实现基于角色的访问控制（RBAC）
2. 数据的增删改查操作
 - 至少包含3个主要数据实体的完整CRUD操作
3. 会话管理
 - 正确使用HttpSession进行会话跟踪
4. 数据验证
 - 实现客户端（JavaScript）和服务器端的数据验证
5. 错误处理和日志记录
 - 实现全面的错误处理机制
 - 使用日志框架（如log4j）记录关键操作和错误
6. 文件上传/下载功能
7. 使用JSTL和EL表达式优化JSP页面

2.4 文档要求

1. 需求分析文档
2. 系统设计文档
 - 包括数据库设计（ER图和表结构）
 - 系统架构图
 - 主要功能流程图

- 3. 测试计划和测试报告
- 4. 小组工作总结报告
 - 包括每个成员的工作内容和贡献
 - 项目过程中遇到的问题和解决方案
 - 对课程的建议和反馈

3. 选题建议

学生可以从以下主题中选择，也可以自行提出其他主题：

- 1. 网上书店系统
- 2. 人事管理系统
- 3. 在线聊天室
- 4. 仓库管理系统
- 5. 学生信息管理系统
- 6. 物流管理系统

4. 评分标准

- 系统功能完整性和正确性：30%
- 代码质量和架构设计：25%
- 用户界面和用户体验：15%
- 文档质量：15%
- 创新性和难度：10%
- 小组协作和表现：5%

评分标准	权重	优秀 (90-100%)	良好 (75-89%)	合格 (60-74%)	需要改进 (0-59%)
系统功能性和正确性	30%	<ul style="list-style-type: none">- 所有要求的功能都正确实现- 系统在各种条件下都能完美运行- 出色的错误处理和边缘情况管理- 创新性地实现了超出要求的附加功能	<ul style="list-style-type: none">- 大多数功能正确实现- 系统运行良好，只有少量问题- 良好的错误处理- 实现了一些附加功能	<ul style="list-style-type: none">- 基本功能已实现- 系统运行有一些bug- 基本的错误处理- 满足最低要求	<ul style="list-style-type: none">- 缺少关键功能- 存在严重bug或系统崩溃- 错误处理差或没有- 未能满足最低要求
代码质量和架构设计	25%	<ul style="list-style-type: none">- 代码清晰、组织良好且有详细文档- 出色地使用MVC架构- 正确使用设计模式- 代码高度可维护和可扩展- 高效的算法和数据结构	<ul style="list-style-type: none">- 代码组织良好且有文档- 良好地使用MVC架构- 使用了一些设计模式- 代码可维护- 代码总体效率不错	<ul style="list-style-type: none">- 代码可读，有基本文档- 尝试使用MVC架构- 基本可维护- 代码功能正常但未优化	<ul style="list-style-type: none">- 代码组织混乱或文档差- 没有清晰的架构- 难以维护- 代码效率低或有问题

评分标准	权重	优秀 (90-100%)	良好 (75-89%)	合格 (60-74%)	需要改进 (0-59%)
用户界面和用户体验	15%	<ul style="list-style-type: none"> - 直观且美观的UI - 完全响应式设计 - 出色的无障碍功能 - 所有页面风格一致且专业 - 创新的UI/UX元素 	<ul style="list-style-type: none"> - 清晰且功能性的UI - 大部分响应式设计 - 一些无障碍功能 - 大多数页面风格一致 - 良好的整体用户体验 	<ul style="list-style-type: none"> - 基本功能性UI - 有限的响应式设计 - 最小程度考虑无障碍 - 设计有一些不一致 - 足够的用户体验 	<ul style="list-style-type: none"> - UI混乱或不吸引人 - 非响应式设计 - 未考虑无障碍 - 设计不一致 - 用户体验差
文档质量	15%	<ul style="list-style-type: none"> - 全面、清晰且结构良好的文档 - 详细的系统设计和架构说明 - 全面的用户手册 - 代码有清晰的注释 - 有见地的项目回顾 	<ul style="list-style-type: none"> - 涵盖主要方面的良好文档 - 清晰的系统设计说明 - 足够的用户手册 - 代码有有用的注释 - 有思考的项目回顾 	<ul style="list-style-type: none"> - 涵盖关键点的基本文档 - 简单的系统设计说明 - 基本的用户说明 - 一些代码注释 - 简短的项目回顾 	<ul style="list-style-type: none"> - 文档不完整或不清晰 - 系统设计说明模糊或缺失 - 用户说明不足 - 极少或没有代码注释 - 项目回顾缺失或肤浅
创新性和难度	10%	<ul style="list-style-type: none"> - 对问题的高度创新解决方案 - 实现了超出课程材料的高级功能 - 集成了前沿技术 - 解决了独特的挑战性问题 	<ul style="list-style-type: none"> - 解决方案中有创新元素 - 实现了一些高级功能 - 使用了超出基本课程要求的技术 - 解决了中等难度的问题 	<ul style="list-style-type: none"> - 有一些创新尝试 - 主要使用课程中涵盖的技术 - 基本使用了要求的技术 - 解决了标准难度的问题 	<ul style="list-style-type: none"> - 缺乏创新性 - 只使用课程中涵盖的基本技术 - 技术挑战性最小 - 解决了简单问题
团队协作和表现	5%	<ul style="list-style-type: none"> - 出色的任务分工 - 明确证据显示持续沟通 - 所有团队成员都有重要贡献 - 有效使用版本控制 - 有见地的同伴评价 	<ul style="list-style-type: none"> - 良好的任务分工 - 有证据显示定期沟通 - 所有团队成员都有贡献 - 使用了版本控制 - 诚实的同伴评价 	<ul style="list-style-type: none"> - 基本的任务分工 - 有一些沟通证据 - 贡献不均但可接受 - 基本使用版本控制 - 完成了同伴评价 	<ul style="list-style-type: none"> - 任务分工差 - 最小程度的沟通 - 贡献严重不均 - 很少或没有使用版本控制 - 同伴评价不完整或缺失

5. 提交要求

5.1 提交内容

1. 完整的系统源代码（包含数据库脚本）
2. 系统演示视频（MP4格式，时长4-5分钟）
3. 项目文档（包括上述所有要求的文档）
4. 小组报告，包含以下内容：
 - 问题描述
 - 需求分析
 - 系统设计
 - 系统实现
 - 系统测试

- 总结反思（每个组员都需要提供个人反思）
- 组员工作分配（详细列出每个组员的工作内容与完成情况）

5.2 提交方式

- 提交时间：2026年1月10日前
- 提交方式：由各班学委收集并统一提交
- 代码应上传至Git仓库，并提供仓库访问链接

6. 注意事项

- 鼓励小组成员之间的有效沟通和任务分工
- 遵守学术诚信，禁止抄袭或剽窃他人作品
- 在开发过程中注意定期备份代码和文档
- 关注Web应用的安全性，如防止SQL注入、XSS攻击等
- 考虑系统的可扩展性和未来维护
- 如遇到技术难题，可在规定时间内向授课教师寻求帮助
- 鼓励学生探索和使用新技术，但需在项目文档中说明原因和收益