在线课程教学平台

可行性分析报告

课程：软件工程

成员：郑皓天

提交时间：2024 年 12 月 27 日

目录

（一）建设目标

- **核心目标：**搭建简洁高效的在线课程教学平台

  平台能够实现“游客”“学生”“教师”“管理员”四类角色的基本功能需求，注重功能实用性和开发效率。

- **具体功能目标**
  - 游客角色：能够浏览课程和注册成为学生用户，权限受限以引导注册。
  - 学生角色：查看和搜索课程，注册课程并在线学习课程内容，提交作业、参与课程评测，查看成绩与反馈。
  - 教师角色：创建课程并设置课程信息（如标题、简介、分类），编排课程内容，上传教学资料，设计作业与考试，查看学生学习进度和成绩统计。
  - 管理员角色：管理用户（创建、删除、分配角色），管理课程内容（审核课程、删除违规课程），监控平台运行状态（查看错误日志、维护数据库）。
- **设计目标**
  - 简洁的系统架构：采用轻量级框架和工具，保证系统快速开发和上线。
  - 友好的用户界面：以响应式设计为主，确保 PC 和移动端均能流畅访问。
  - 稳定性与安全性：提供基本的权限控制和数据安全措施（如密码加密和用户分级管理）。
- **性能目标**

  支持 100 名左右用户的并发访问，保证页面加载流畅，操作无卡顿。
- **开发目标**
  - 开发周期：一周内完成从需求分析到上线的所有开发任务。
  - 独立开发：依托个人开发能力，从设计到实现全流程完成项目。
- **扩展性目标**
  - 保留后续功能扩展的可能性
  - 模块化设计，为系统升级或迁移提供便利。

（二）可能的解决方案

- **系统架构设计**

预计采用前后端分离架构，前端负责页面渲染和用户交互，后端提供数据服务和逻辑处理。

- **前端技术栈：**
  - Vue 3.0：构建用户界面，支持响应式设计和组件化开发。
  - Vite：提供高效的开发环境和快速热更新，提升开发效率。
  - Element-Plus：使用现成的组件库快速搭建界面，减少开发工作量。
  - Pinia：轻量级状态管理工具，管理用户信息、课程数据等全局状态。
  - Vue-Router：实现多页面路由切换，支持权限控制和动态加载页面。
  - Axios：用于与后端进行 HTTP 通信，实现接口调用和数据交互。
  - ESLint：规范代码风格，保持代码一致性和可维护性。
- **后端技术栈：**
  - Java：实现业务逻辑，保证系统稳定性和可扩展性。
  - Tomcat：提供高性能的 Web 容器，部署后端服务。
  - MySQL：关系型数据库，用于存储用户信息、课程数据、作业记录等。
- **功能模块设计**
  - 用户模块：实现游客、学生、教师、管理员四种角色的用户注册、登录及权限分配。
  - 课程模块：支持课程创建、浏览、搜索、注册和内容学习。
  - 作业模块：学生提交作业，教师查看和评分，支持作业评测功能。
  - 后台管理模块：管理员管理用户和课程，监控平台运行状态。

（三）技术可行性分析

- **现有技术栈支持快速开发**
- **前端技术：**
  - Vue 3 是目前成熟的前端框架，提供响应式编程和组件化开发，配合 Vite 提高开发效率。
  - Element-Plus 提供丰富的 UI 组件，减少自定义组件开发的工作量。
  - Pinia 和 Vue-Router 可以简化状态管理和路由配置，适合快速构建权限控制和页面跳转功能。
  - Axios 是主流的 HTTP 请求库，支持统一封装接口请求和处理。
- **后端技术：**
  - Java 具备稳定性和可靠性，适合实现复杂的业务逻辑。

- MySQL 是常用的关系型数据库，支持结构化数据存储，满足课程、用户、作业等数据管理需求。
- Tomcat 轻量级且性能良好，足以支撑中小型系统的部署与运行。

- **开发人员能力**

本人对 Vue 和 Java 技术栈有一定熟悉程度，能够独立完成前后端的开发与联调。

前后端分离架构清晰，便于单人逐步实现模块并进行调试和优化。

- **工具与框架支持**

Vite 提供了开发环境的快速搭建能力，减少基础配置的时间。

使用现成的开发工具（如 WebStorm、IntelliJ IDEA 和 Postman）可以快速完成代码编辑、调试和接口测试。

- **部署环境可行性**

前端构建后可托管在 Nginx 或静态文件服务器上，后端应用可部署在本地或云端 Tomcat 上，部署过程简单。

MySQL 数据库易于安装和维护，可通过本地或云端部署满足开发需求。

- **技术扩展性**

系统采用模块化设计，后续可以在现有架构基础上新增功能模块。

前后端分离的架构允许未来通过微服务化实现更大规模的系统扩展。

- **开发周期保障**

Vue 和 Element-Plus 能快速实现界面功能，降低 UI 开发的复杂度。

- **潜在风险与解决方案**
  - 时间限制：一周开发时间较短，需要优先实现核心功能（用户管理、课程浏览、作业功能）。
  - 并发支持：初期通过简单限流或缓存方案（如 Redis）解决基本并发需求，满足小规模用户访问。
  - 安全性：初步实现加盐哈希密码加密和角色权限控制，保障系统安全。

（四）经济可行性分析

- **主要开销**

阿里云服务器租用费用：每天 2.23 元，项目开发时间为一周，共需支付：$2.23 \times 7 = 15.61$ 元，该费用低廉，完全在个人可接受范围内。

- **其他潜在开销**

- ❖ 域名注册（如需上线测试）：一般为每年 50-100 元，不是项目必须，可暂时忽略。
- ❖ 数据库费用：项目初期 MySQL 可在云服务器本地部署，无需额外费用。
- ❖ 开发工具：采用免费工具（如 VS Code、IntelliJ IDEA 社区版），不存在开发软件费用。

➢ **资源配置与性价比**

服务器配置：阿里云服务器基础配置（如 1 核 CPU、1GB 内存）足以满足本项目需求，既保证性能，又控制成本。

平台使用人数：预计用户量较少，无需高配置服务器，进一步降低运行成本。

➢ **开发成本**

由于项目由本人独立开发，无需额外支付开发费用或外包费用，显著降低成本。

➢ **长期运行费用**

如果项目需要长期运行，按照每天 2.23 元计算，一年运行成本为：$2.23 \times 365 \approx 814$ 元，即使长期运行，成本仍处于低水平，适合小型项目或个人开发维护。

➢ **成本收益比**

开发成本主要集中在服务器租赁，合计费用不足 20 元，开发周期短且目标明确。

对于个人而言，该项目不仅提供实际的开发经验，还可以作为作品展示，提升个人竞争力，潜在收益较高。

（五）操作可行性分析

➢ **开发过程可行性**

- ❖ 技术能力：本人熟悉前后端开发所需的技术栈（Vue 3、Vite、Element-Plus、Java、Tomcat 和 MySQL），具备独立完成项目开发的能力。
- ❖ 开发工具：使用成熟的开发工具（VS Code、IntelliJ IDEA、Postman 等）和框架（Vue、Spring Boot），开发效率高，无额外学习成本。
- ❖ 开发周期：一周时间内，按模块化设计优先完成核心功能，完全可行。

➢ **部署过程可行性**

- ❖ 后端部署：阿里云服务器提供可靠的环境支持，通过 Tomcat 部署 Java 应用，流程简单，本人已具备操作能力。
- ❖ 前端部署：利用 Nginx 部署前端静态资源，操作简单且效率高。
- ❖ 数据库配置：MySQL 可在云服务器本地部署，连接后端服务实现高效的数据管理，已无技术障碍。

- **用户操作可行性**
  - 用户界面设计：使用 Element-Plus 快速搭建简洁、易用的界面，用户可以轻松浏览课程、注册账号、提交作业等。
  - 多角色操作：通过角色权限分离，确保游客、学生、教师和管理员的操作界面简洁且功能明确，避免混淆。
- **维护和扩展可行性**
  - 模块化设计：系统采用前后端分离架构，各功能模块独立，便于后期维护和扩展。
  - 文档支持：Vue 和 Element-Plus 等框架文档丰富，问题解决方便快捷。
- **潜在风险及解决方案**
  - 服务器配置不足：若并发量过高，可临时升级阿里云服务器配置。
  - 功能未完全实现：若时间不足，优先开发用户注册、课程管理等核心功能，后续逐步优化。

（六）社会可行性分析

- **项目背景及意义**

本在线课程教学平台是软件工程课程的大作业，旨在通过实际项目开发，全面提升个人在软件设计、前后端开发、系统部署等方面的能力。

平台发布在 GitHub，不仅方便课程提交和评审，还能作为公开项目供其他学生和开发者参考，具有一定的教育推广意义。

- **教育与实践价值**
  - 大学生实践平台：该项目面向学生，设计简单易用，可作为同类课程教学系统的参考实现，为教育领域提供一种低成本、高效的课程管理解决方案。
  - 开源共享：项目代码开源后，其他开发者可以通过 GitHub 学习其架构设计和功能实现，促进技术交流和共享，进一步提升开发水平。
  - 职业发展：作为一个完整的实践项目，该平台能够丰富个人项目经验，有助于未来求职时展示技术实力。
- **社会效益**
  - 提高学习效率：学生用户可通过平台实现高效的课程学习和作业管理，教师用户可通过数据统计了解学习效果，提升教学效率。
  - 降低开发门槛：开源后，教育机构或个人开发者可以参考代码，快速搭建类似平台，推动教育信息化的发展。
  - 技术推广：平台的前后端技术栈（Vue、Java、MySQL 等）成熟稳定，

适合初学者学习和实践，进一步普及相关技术。

- **潜在风险及应对**
  - 平台受众有限：因目标群体主要为学生和开发者，实际使用人群较少，但这并不影响其作为学习和展示项目的意义。
  - 代码安全性：在 GitHub 上开源时，需要注意敏感信息（如数据库配置、密钥等）的保护，避免造成安全隐患。
- **项目对社会的积极影响**

项目不仅能帮助开发者自身成长，还能促进更多人了解软件开发流程，提高自主学习能力，对个人职业发展具有积极意义。

（七）开发计划

- **项目周期：**一周
- **任务划分及时间安排：**

◆ **需求分析与系统设计（1 天）**

- 确定项目功能需求，包括用户角色及对应功能（游客、学生、教师、管理员）。
- 绘制功能模块图和数据库 ER 图，规划前后端接口设计。
- 明确系统架构：前后端分离，罗列需要的业务接口。

◆ **数据库设计与后端搭建（2 天）**

- Day 1：

创建 MySQL 数据库及表（用户、课程、作业、评测等）。

编写基础的 CRUD 操作代码。

- Day 2：

使用 Java 搭建后端服务，完成用户认证、权限管理功能。

开发课程管理、作业管理、数据统计等核心功能的接口。

◆ **前端开发（3 天）**

- Day 1：

配置 Vue 3 + Vite + Element-Plus 开发环境，搭建基本页面框架。

实现登录注册、课程列表等基础页面。

- Day 2：

开发学生用户功能页面，包括课程详情、在线学习、作业提交等。

开发教师用户功能页面，包括创建课程、编辑内容、查看统计等。

- Day 3：

  完成管理员功能页面，包括用户管理、系统配置等功能。

  调试页面样式和功能，确保前后端联调无误。

- **测试与部署（1 天）**
  - 编写测试用例，对前后端主要功能进行功能测试和接口测试。
  - 修复已知问题，优化页面交互和后端逻辑。
  - 部署项目：将前端打包后通过 Nginx 部署到阿里云服务器，将后端服务通过 Tomcat 部署，确保可正常访问。

- **任务优先级：**

优先实现登录注册、课程浏览与管理等核心功能；其他功能根据时间逐步完善。

（八）需求分析

- **系统用户角色及功能需求**

- **游客**

功能：仅能浏览课程列表，无法注册课程或提交作业，功能受限。

- **学生**

功能：注册登录平台，查看课程详情，注册课程，在线学习课程内容，提交作业，查看作业评测结果。

- **教师**

功能：创建课程，编排课程内容，设计习题并发布作业，查看课程学习情况及数据统计。

- **管理员**

功能：管理用户账号（添加、删除、修改用户信息），管理课程数据（创建、删除、编辑课程信息），系统配置维护。

- **功能模块划分**

- **用户管理模块**
  - 用户注册、登录、角色分配及权限管理。

- **课程管理模块**
  - 游客：查看课程列表。
  - 学生：注册课程、学习课程内容。
  - 教师：创建和编辑课程，上传教学资源。

- **作业管理模块**

- 学生：在线提交作业，查看作业评测结果。
- 教师：发布作业，查看学生提交情况，评分评测。

◆ **数据统计模块**

- 教师：查看课程学习情况（完成率、学生作业得分分布）。
- 管理员：查看系统使用数据（用户数、课程数）。

➢ **非功能需求**

◆ **性能需求**

支持少量并发请求（预计使用人数较少）。

界面响应时间小于 1 秒，核心功能操作完成时间小于 3 秒。

◆ **可靠性**

系统需保证 99% 的可用时间，避免严重错误导致系统崩溃。

◆ **安全性**

实现用户数据加密存储，敏感信息如密码需加盐哈希处理。

限制不同角色的访问权限，防止非法操作。

◆ **扩展性**

系统模块化设计，方便后续增加新功能或优化性能。

➢ **数据需求**

- 用户数据表：存储用户基本信息（用户名、角色、密码、注册时间等）。
- 课程数据表：存储课程标题、简介、教师信息、资源链接等。
- 作业数据表：存储作业内容、提交时间、评测分数等。

➢ **前后端交互需求**

前端通过 Axios 调用后端接口，完成用户认证、课程操作、数据交互等。

接口设计包括用户认证、课程查询与管理、作业提交与评测等功能。