

项目2 基于“学习共同体”理念的教学平台

1 开发任务概述

1.1 开发目标

“学习共同体”是由学生和教师（助教）共同组成的，以完成共同的学习任务为载体，以促进成员全面成长为目的，强调在学习过程中以相互作用式的学习观作指导，通过人际沟通、交流和分享各种学习资源而相互影响、相互促进的基层学习集体。相比于传统教学的“教师-学生”学习模式，“学习共同体”还引入了“学生-学生”学习模式。本项目需要你实现一个基于“学习共同体”理念的教学平台，该平台可以在原有教师和助教出题基础之上实现学生之间相互出题，然后系统自动分配题目，并通过一定的机制保证自己出的题目不会分配给自己；该平台还可以实现学生之间互评，并且为了避免出现评价不公平的情况，系统需要引入一些调停机制，对于存在争论（例如多个互评分数的方差过大）的题目，系统自动发给教师评判等；该平台还可以实现类似于Gitbook的共同编辑文档功能。当老师设置一个章节（作业所属的单元知识）并将对应的作业布置下来后，每位同学可以以树形结构的形式编辑该章节下的文档来提交的作业，使得自己的作业可以共享给其他同学。因此，本项目将提交课程作业，学生相互出题和学生互评等课程任务作为一个完整的主题。

1.2 前提约束与依赖条件

1.2.1 前提约束

- 不得抄袭，被发现后果自负。
- 允许借鉴与学习开源项目与材料。

1.2.2 依赖条件

- 能够熟练编写HTML、CSS、JavaScript，掌握JSON和RESTful的基础应用，对JavaEE基础架构、Spring Boot、MyBatis、数据库设计有一定的认识。并对课程中涉及的其他Web技术有一定的基础和了解。
- 本项目所涉及的技术范围是：
 - Web基础课程的知识（HTML，CSS，JavaScript），AJAX技术；
 - 建议前端采用Angular组件开发；
 - 后端采用Spring Boot框架，MyBatis作数据库持久化层。数据库不限制，可以使用MySQL、MongoDB或者图数据库（如Neo4j）；
 - 采用前后端分离的架构，后端提供RESTful风格的API，给前端调用，前后端之间用JSON或者XML传递数据；
 - 前端与后端的程序均部署在云服务器上，在线上环境直接演示。

2 系统基本功能与流程

2.1 系统角色

角色名称	职责描述
教师 (助教)	开设课程，设立项目，出题，设立评分标准，对学生进行评分等。
学生	加入课程，学生间组队及成员划分（组长对项目进行规划，布置任务，监督完成情况；组员可查看项目任务，标记完成情况）；学生间出题与互评，提交作业等。
后台管理员	对系统的用户、课程、项目进行增删改查，对平台的监管等。

2.2 功能要求

2.2.1 基本功能

- 前台页面
 - 提供用户注册。注意：仅学生需要注册，老师（助教）由管理员添加。
 - 提供用户登录。
- 用户后台页面
 - 记录用户信息、用户角色等；
 - 添加/修改/删除个人信息以及上传头像等
 - 可查看历史作业发布情况，当前用户历史作业提交情况以及其他用户共享的作业。
- 学生出题
 - 提供学生出题的功能。
- 学生答题
 - 提供试题分发的功能。注意：试题分发时需要引入一定的机制保证学生不会分发到自己出的试题。
 - 提供学生答题的功能。
- 学生互评
 - 提供学生间互评的功能。注意：学生间互评需要引入一定的调停机制，有争议的题目自动提交给教师（助教）处理。
- 文档共同编辑
 - 教师（助教）可以设置一个章节，并在章节下布置对应的作业。注意：当教师（助教）布置完成作业后，学生应能收到新作业的提示。
 - 每位同学可以以树形结构的形式编辑该章节下的文档来提交的作业，以及其他同学能文档已更新的提示，并且可以查看文档更新内容。
- 传统教学平台上该具备的功能，包括但不限于
 - 教师（助教）出题、教师（助教）批改等功能。从代码复用性角度，可以将其与学生出题、学生互评等功能结合，但针对不同用户角色显示不同功能界面。

- 教师（助教）的课程管理功能。仅老师可以增删课程，学生可以查看课程；课程发布后，不可修改。
- 教师（助教）的评分功能。教师（助教）评分时，应显示相应的参考数据，比如作业完成情况，讨论板留言内容等。学生可以查看老师对自己的各项评分。
- 管理员的用户管理（增删查改）功能。管理员可以添加任何用户，其中老师用户只能由管理员添加。
- 课程内容分享功能。教师、助教可以上传PPT/PDF等文件，学生可以在线浏览或者下载文件。
- 系统部署在云服务器上，提供可以访问的公网地址。

2.2.2 进阶功能

- 你可以从下面三组进阶功能中任选一组实现：
 - 第一组：系统设计的进阶功能
 - 系统可以提供多种题目类型，如选择题、填空题、解答题等。对于不同类型的题目，鼓励引入不同的调停机制，鼓励设计多种判断是否存在争议的方法。
 - 管理员可以进行细致的、图形化的权限管理。
 - 其他具有实际意义且富有挑战性的功能。
 - 第二组：文档编辑内容的版本回溯
 - 由于文档是开放编辑权限的，因此教师和学生可以自由增删文档内容。版本回溯可以帮助教师和学生定位到之前任意版本的文档以查看内容，并可以实现文档内容的恢复。注意：实现该功能你要考虑到存储效率的问题，最低效的方法是保存编辑后每一份文档的内容，然后按需回溯，然而这会极大增加存储开销。如果采用这种实现方式，小组将会获得偏低的分数。
 - 第三组：文档共同编辑的一致性维护
 - 在文档被多人同时编辑时，实现文档内容一致且完整，注意：该功能难度大，实现该功能需要考虑并发问题，可以参考腾讯文档、飞书等在线文档编辑系统。

2.3 性能要求

除了功能需求外，平台还应在以下几个方面有一定的要求：

- 灵活性：尽量使用灵活的设计，提高复用与适应性，后台项目架构要规范。
- 安全性：充分考虑到安全性，数据传输以及密码的保存，使用相关安全技术。
- 易用性：为便于师生群体使用本系统，其应在界面和功能上做到清晰易懂，降低新用户的学习成本。

3 系统说明

给出部分界面供同学们参考，实现时同学们可自由发挥。

用户登录

基于“学习共同体”理念的教学平台示例

欢迎你，夏家峰

当前角色：助教

修改密码

账户信息

退出登录

导航栏

课程日历 >

收件箱 >

反馈与帮助 >

我的课程

高级Web设计 >



4 评分细则

4.1 分数组成

- 基本功能分（满分 100 分）：要求完成系统基本内容与流程。
- 进阶功能分（最多可加20分）：包括但不限于更具挑战性的功能，更好的开发部署流程、设计模式。
- 个人工作分（小组总共30分）：根据小组分工及个人完成的工作量（贡献比例）为组员分配对应的分数。
- 个人最终得分 = 基本功能分 + 进阶功能分 + 个人工作分，值域为 [0, 150]。

4.2 评分点

功能项	得分项	说明	最高分数
UI和交互（满分10分）	界面美观		5
	交互体验流畅	页面跳转和交互无明显失误，菜单和返回正常使用。	5
基本功能（满分60分）	注册和登录页面		5
	用户后台页面		5
	学生出题		7
	学生答题		10
	学生互评		8
	文档共同编辑		10
	传统教学平台功能		15
工程能力（满分20分）	文档	说明清晰完整，结构图准确易懂。	10
	代码风格、注释	代码整洁、拓展好，设计模式等。	5
	系统架构	重点关注复用性、可拓展性。	5
项目部署（满分10分）	将服务器部署到公有云上		5
	使用Docker部署		5
进阶功能（任选一组实现，最高可加20分）	系统设计的进阶功能	第一点10分，第二点5分，第三点5分，根据实际情况最高可加20分。	20
	文档编辑内容的版本回溯	根据实际情况最高可加20分。	20
	文档共同编辑的一致性维护	根据实际情况最高可加20分。	20

4.3 评分点说明

- 每一项的分数取决于该项功能的完成度，完成度和可用性越好，分数越高。
- 项目完整度和易用性评价标准：
 - A. 功能残缺，不能完整运行，响应慢；
 - B. 完成规定的用户功能和操作，无明显瑕疵；

- C. 界面舒适，操作合理，响应迅速，鲁棒性强；
 - A、B、C 分别对应分数的0-30分，30分-70分，70分-100分。
- 附加功能必须在文档中明确写出，概述该功能并描述实现原理。
- 项目设计文档需要至少包含：
 - 项目组织以及其中每个文件的说明。
 - 关键功能实现的细节：
 - 服务器部署配置的详细介绍。
- 团队分工文档需要至少包含：
 - 团队成员、分工、具体完成工作，列出每个人的贡献比例。
 - 其他你们想说明的问题。

5 提交

- 提交材料包含以下三项：
 - 源代码：推荐使用 Git 进行协作，提交到 GitHub 等 Git 托管平台上。
 - 文档：推荐使用 Markdown 编写项目文档，与源代码一同提交到 Git 托管平台上。
 - 可供访问的公网地址，以及系统的操作说明（玩法）。
- 提交材料需要进行压缩打包，并提交到超星学习通上，文件名中请包含小组所有成员的姓名与学号。
- 友情提示：请尽早开工，临时赶工很有可能完不成。