

软件设计说明书

**项目名称：** 基于区块链的在线学习平台

学生姓名 邓文彬

学 号 202200863

学 院 区块链学院

年 级 2022级

专 业 区块链技术

指导教师 邹林薏

完成日期 2024年11月12日

目 录

[1 引言 1](#_Toc19521)

[1.1 开发背景及目的 1](#_Toc30443)

[1.1.1开发背景 1](#_Toc4893)

[1.1.2开发目的 1](#_Toc9385)

[1.2 命名规范 1](#_Toc19851)

[1.3 术语和缩写词 2](#_Toc19075)

[1.4 参考资料 3](#_Toc31795)

[1.5 版本信息 3](#_Toc32520)

[2 总体设计 3](#_Toc14792)

[2.1 硬件运行环境 3](#_Toc24318)

[2.2 软件运行环境 3](#_Toc20841)

[2.3 子系统清单 4](#_Toc27821)

[2.4 功能模块清单 4](#_Toc14660)

[3 数据库设计 5](#_Toc22154)

[3.1 数据库中表名列表 5](#_Toc6919)

[3.2 数据库中的表关系 6](#_Toc18513)

[3.3 数据库表的详细清单 7](#_Toc8483)

[4 典型的子系统设计 9](#_Toc16841)

[4.1 个人信息管理系统设计 9](#_Toc14070)

[4.2 课程管理系统设计 10](#_Toc2244)

[4.3 学习进度系统设计 10](#_Toc6984)

[4.4 认证与证书管理系统设计 11](#_Toc14371)

[4.5 用户信息维护系统设计 11](#_Toc24522)

[5 界面设计 12](#_Toc15326)

[5.1 网站设计母板 12](#_Toc1398)

[5.2 主页: /index/ 13](#_Toc19040)

[6 接口设计 14](#_Toc9472)

[6.1 用户接口 14](#_Toc11705)

[6.2 外部接口 14](#_Toc19381)

[6.3 内部接口 15](#_Toc886)

[7 角色授权设计 15](#_Toc14467)

[8 系统错误处理 15](#_Toc4125)

[8.1 出错信息 15](#_Toc16888)

[8.2 故障预防与补救 15](#_Toc27551)

[8.3 系统维护设计 16](#_Toc13685)

**智慧校园区块链学习软件设计说明书**

**1 引言**

## 1.1 开发背景及目的

## 1.1.1开发背景

开发的软件名称为“智慧校园区块链学习平台”（以下简称“学习平台”），旨在利用区块链技术为教育机构提供一个安全、透明、高效的在线学习环境。任务提出者为[江西软件职业技术大学]，该机构认识到传统教育模式的局限性，并寻求通过技术创新来提升教育质量和管理效率。开发者[邓文彬]拥有丰富的区块链和教育软件开发经验，负责整个学习平台的设计、开发和测试工作。主要用户群体包括学生、教师、教育管理人员以及对继续教育感兴趣的个人。实现软件的单位为[江西软件职业技术大学]，负责软件的部署、维护和用户支持，确保平台的稳定运行和持续优化。项目批准时间为2024年9月1日，预计完成日期为2024年12月11日。

## 1.1.2开发目的

随着区块链技术的快速发展，教育领域也开始探索其应用潜力。基于区块链的自主在线学习平台旨在利用区块链的去中心化、安全性和透明性，提供一个可信赖的学习环境。该平台的开发目的是为用户提供一个自主学习的空间，确保学习记录的不可篡改性，并通过智能合约实现自动化的学习管理和评估。

为了满足用户多样化的需求，本文提出以下目标：

1.通过区块链技术，确保用户的个人数据和学习记录的安全性和隐私性。

2.设计简洁友好的用户界面，提供流畅的用户体验。

3.根据用户的学习习惯和需求，提供定制化的学习内容和推荐。

4.利用区块链的不可篡改性，确保所有学习记录和评估结果的透明和可信。

5.通过智能合约实现课程管理、学习评估和证书颁发的自动化，减少人为干预，提高效率。

## 1.2 命名规范

数据库表名：所有表名使用英文命名，首字母大写，采用驼峰式命名法。例如：UserAccount, CourseCatalog, LearningRecord, EnrollmentHistory, PaymentTransaction, CourseFeedback, UserProfileSettings。

数据库字段名：字段名以英文命名，首字母大写，采用驼峰式命名法。例如：UserID, CourseName, StartDate, EndDate, PaymentStatus, FeedbackScore, EnrollmentDate, ProfilePictureURL, LastLoginTime。

变量命名规范

私有变量：私有变量以小写英文命名，并以“\_”开头。例如：\_userList, \_courseDetails, \_loginStatus, \_profileSettings, \_transactionHistory。

公共变量：公共变量采用大写字母开头的驼峰式命名法。

例如：CourseList, UserProfile, PaymentHistory, CurrentUser, ActiveSessions, CompletedCourses, RecommendedCourses。

函数命名规范

函数名称：所有函数名称以大写字母开头，采用驼峰式命名。例如：DisplayCourseCatalog(), RegisterUser(), SubmitAssignment(), GetEnrollmentHistory(), ProcessPayment(), UpdateUserProfile(), GenerateReport(), CalculateFeedbackScore()。函数名应以动词开头。

其他命名规则

类名：类名以大写字母开头，采用驼峰式命名。例如：UserManager, CourseService, PaymentProcessor, FeedbackAnalyzer, EnrollmentController, SessionManager, ReportGenerator。

常量命名：所有常量均使用全大写英文命名，单词间用下划线分隔。例如：MAX\_COURSE\_COUNT, DEFAULT\_USER\_ROLE, SESSION\_TIMEOUT, MAX\_UPLOAD\_SIZE, MIN\_PASSWORD\_LENGTH, COURSE\_FEEDBACK\_THRESHOLD, MAX\_LOGIN\_ATTEMPTS。

## 1.3 术语和缩写词

区块链（Blockchain）：一种分布式账本技术，具有去中心化、不可篡改和透明的特点。

智能合约（Smart Contract）：在区块链上运行的自动化协议，用于执行合约条款。

自主学习（Self-directed Learning）：学习者根据自己的需求和兴趣，自主选择学习内容和进度的学习方式。

## 1.4 参考资料

[1]王宇.区块链赋能的可信数据安全共享机制研究[J].现代商贸工业,2024,45(19):46-48.DOI:10.19311/j.cnki.1672-3198.2024.19.015.

[2]张晨.区块链智能合约安全实践教学平台搭建分析[J].安徽电气工程职业技术学院学报,2024,29(03):109-116.

[3]《需求规格说明书》.区块链学院,2024

## 1.5 版本信息

版本更新信息如表1-1所示。

表 1-1 版本更新表

|  |  |  |  |
| --- | --- | --- | --- |
| 版本号 | 创建人 | 创建日期 | 更新纪要 |
| V1.0 | 邓文彬 | 2024-11-12 | 文档创建 |
| V1.1 | 邓文彬 | 2023-11-13 | 界面设计编写 |

**2 总体设计**

## 2.1 硬件运行环境

服务器的配置与型号如下表2-1所示。

表 2-1 配置与型号

|  |  |
| --- | --- |
| 配置 | 型号 |
| CPU | 12th Gen Intel(R) Core(TM) i5-12500H 2.50 GHz 以上 |
| 内存 | 16.0GB |
| 网络配置 | 100M网卡及以上 |

## 2.2 软件运行环境

服务器的操作系统和型号如下表2-2所示。

表 2-2 配置与型号

|  |  |
| --- | --- |
| 配置 | 型号 |
| 操作系统 | Windows 11 版本:23H2 |
| 数据库 | MySQL-8.0.15 |

## 2.3 子系统清单

子系统清单如表2-3所示。

表 2-3 子系统功能表

|  |  |
| --- | --- |
| 子系统名称 | 子系统功能描述 |
| 个人信息管理系统 | 注册登录功能的实现，个人信息维护更新 ，管理  员管理网页用户 |
| 课程管理系统 | 用户可以选择课程，查看课程目录，课程信息维护，支持课程推荐及课程搜索。 |
| 学习进度管理系统 | 记录并管理用户的学习进度和成绩，显示学习计划和完成情况 |
| 认证与证书管理系统 | 使用智能合约进行学习成果的自动评估，颁发区块链认证的学习证书。 |
| 用户信息维护系统 | 管理员负责维护用户数据、权限管理和账户设置。 |

## 2.4 功能模块清单

功能模块清单如表2-4所示。

表 2-4 功能模块表

|  |  |
| --- | --- |
| 名称 | 模块功能描述 |
| 注册账号 | 游客注册为会员，填写个人信息完成账号注册 |
| 登录系统 | 已注册用户通过账号和密码登录系统 |
| 找回密码 | 会员丢失密码后，通过身份验证流程重新设置密码 |
| 浏览课程目录 | 游客和会员查看课程目录，了解课程内容 |
| 选择课程 | 会员选择并加入课程，加入个人学习计划 |
| 课程播放 | 实现课程视频的播放，提供暂停、快进和回放等功能 |
| 作业提交 | 会员完成课程后提交作业，系统记录并传递给评估模块 |
| 课程讨论 | 会员在课程讨论区发表观点、回复他人并进行互动 |

续表 表 2-4 功能模块表

|  |  |
| --- | --- |
| 查看学习进度 | 会员查看学习计划进度、已完成课程和成绩 |
| 个性化推荐 | 根据用户兴趣和学习记录推荐适合的课程 |
| 维护用户信息 | 用户更新个人信息，管理员管理用户的账号和权限 |
| 管理课程内容 | 管理员添加、编辑和更新课程内容 |
| 维护学习记录 | 管理员管理和更新用户的学习记录 |
| 评估学习成果 | 评估用户的学习情况，记录评估分数，反馈学习效果 |
| 颁发认证证书 | 学习完成后颁发区块链认证证书，记录在用户的学习记录中 |
| 生成学习报告 | 分析用户的学习记录，生成可视化学习报告 |

**3 数据库设计**

## 3.1 数据库中表名列表

数据库中表名如表3-1所示。

表 3-1 数据库列表

|  |  |  |
| --- | --- | --- |
| 编号 | 表名 | 用途 |
| 1 | CourseCatalog | 课程目录表 |
| 2 | Enrollment | 用户课程注册表 |
| 3 | Assignment | 作业提交记录表 |
| 4 | Discussion | 课程讨论记录表 |
| 5 | Progress | 用户学习进度表 |
| 6 | Recommendation | 个性化推荐记录表 |
| 7 | Admin | 管理员信息表 |
| 8 | CourseContent | 课程内容管理表 |
| 9 | LearningRecord | 学习记录表 |
| 10 | Certification | 区块链认证证书表 |
| 11 | Feedback | 用户反馈表 |

续表 表 3-1 数据库列表

|  |  |  |
| --- | --- | --- |
| 12 | Report | 学习报告表 |

## 3.2 数据库中的表关系

在本学习平台的数据库中，主要的表关系如下所示：

**3.2.1 用户信息表**

UserInfo（用户信息表）

主键：UserID

关联关系：

与 CourseEnrollment（课程选课表）表存在一对多关系，即一个用户可以选择多个课程。

与 LearningProgress（学习进度表）表存在一对多关系，即一个用户可以有多个学习进度记录。

与 Certification（证书表）表存在一对多关系，即一个用户可以获得多个学习认证。

**3.2.2 课程信息表**

Course（课程信息表）

主键：CourseID

关联关系：

与 CourseEnrollment（课程选课表）表存在一对多关系，即一个课程可以被多个用户选修。

与 LearningProgress（学习进度表）表存在一对多关系，即一个课程可以有多个学习进度记录。

与 Recommendation（课程推荐表）表存在一对多关系，即一个课程可以被推荐给多个用户。

**3.2.3 课程选课表**

CourseEnrollment（课程选课表）

主键：EnrollmentID

外键：UserID（关联 UserInfo 表），CourseID（关联 Course 表）

关联关系：中间表，用于管理用户和课程的多对多关系，即用户可以选修多个课程，课程也可以被多个用户选修。

**3.2.4 学习进度表**

LearningProgress（学习进度表）

主键：ProgressID

外键：UserID（关联 UserInfo 表），CourseID（关联 Course 表）

关联关系：用于记录用户在每门课程中的学习进度，一个用户可以在多个课程中有不同的学习进度。

**3.2.5 证书表**

Certification（证书表）

主键：CertificateID

外键：UserID（关联 UserInfo 表）

关联关系：用于记录用户获得的学习认证信息，一个用户可以获得多个证书。

**3.2.1 课程推荐表**

Recommendation（课程推荐表）

主键：RecommendationID

外键：CourseID（关联 Course 表），UserID（关联 UserInfo 表）

关联关系：用于记录系统对用户的课程推荐，一个用户可以接收到多个课程推荐，且每门课程可以推荐给多个用户。

各表的主外键关系以及连接方式如下：

UserInfo 和 Course 通过 CourseEnrollment 表实现多对多关系。

UserInfo 和 Course 通过 LearningProgress 表实现多对多关系。

UserInfo 和 Certification 表是一对多关系。

UserInfo 和 Course 通过 Recommendation 表实现多对多关系。

## 3.3 数据库表的详细清单

**3.3.1 课程信息表**

课程信息如表3-2所示。

表 3-2 课程信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 英文字段名 | 中文字段名 | 数据类型 | 是否允许为空 | 主键/外键 |
| CourseID | 课程ID | varchar(20) | 否 | 主键 |

续表 表 3-2 课程信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CourseName | 课程名称 | varchar(50) | 否 | / |
| Description | 课程描述 | text | 是 | / |
| Duration | 课程时长（小时） | int | 否 | / |
| InstructorID | 教师ID | varchar(20) | 否 | 外键 |
| StartDate | 开始日期 | date | 否 | / |
| EndDate | 结束日期 | date | 是 | / |
| Category | 课程分类 | varchar(30) | 否 | / |
| Level | 难度级别 | ENUM('初级','中级','高级') | 否 | / |

**3.3.2 用户信息表**

用户信息如表3-3所示。

表 3-3 用户信息表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 英文字段名 | 中文字段名 | 数据类型 | 是否允许为空 | 主键/外键 |
| UserID | 用户ID | varchar(20) | 否 | 主键 |
| UserName | 用户名 | varchar(25) | 否 | / |
| Password | 密码 | varchar(64) | 否 | / |
| Email | 邮箱 | varchar(50) | 否 | / |
| PhoneNumber | 手机号码 | varchar(15) | 是 | / |
| RegistrationDate | 注册日期 | date | 否 | / |
| UserRole | 用户角色 | ENUM('普通用户', 'VIP用户') | 否 | / |
| Status | 账户状态 | ENUM('Active', 'Inactive', 'Banned') | 否 | / |
| LastLoginTime | 上次登录时间 | datetime | 否 | / |

**4 典型的子系统设计**

## 4.1 个人信息管理系统设计

**4.1.1 类设计**

该子系统实现用户注册登录、个人信息维护与更新，管理员管理网页用户的功能。

User 类

表示用户的基本信息和相关操作。

属性：

userID（String）：用户ID

userName（String）：用户名

password（String）：用户密码（加密存储）

email（String）：用户邮箱

phoneNumber（String）：用户联系电话

registrationDate（Date）：注册日期

status（String）：用户状态（如 "Active", "Inactive"）

方法：

register()：用户注册

login(String userName, String password)：用户登录

updateInfo()：更新用户的个人信息

resetPassword()：重置密码

viewProfile()：查看用户个人信息

Admin 类

用于管理员管理用户信息。

属性：

adminID（String）：管理员ID

adminName（String）：管理员姓名

role（String）：管理员角色（如 "SuperAdmin", "UserManager"）

方法：

manageUserAccounts()：管理用户账户

resetUserPassword(String userID)：重置用户密码

changeUserStatus(String userID, String status)：更改用户状态

viewAllUsers()：查看所有用户信息

## 4.2 课程管理系统设计

**4.2.1 类设计**

该子系统支持用户选择课程、查看课程目录、课程信息维护，以及课程推荐和搜索功能。

Course 类

用于存储课程的基本信息。

属性：

courseID（String）：课程ID

courseName（String）：课程名称

description（String）：课程描述

duration（int）：课程时长

instructor（String）：讲师信息

category（String）：课程类别

方法：

getCourseInfo()：获取课程信息

recommendCourses()：推荐课程

searchCourses(String keyword)：按关键词搜索课程

viewCourseDetails(String courseID)：查看课程详情

## 4.3 学习进度系统设计

**4.3.1 类设计**

该子系统记录并管理用户的学习进度和成绩，显示学习计划和完成情况。

Progress 类

记录用户的学习进度信息。

属性：

progressID（String）：进度ID

userID（String）：用户ID

courseID（String）：课程ID

completionPercentage（int）：完成百分比

grade（String）：成绩

lastAccessDate（Date）：最后访问日期

方法：

updateProgress(int percentage)：更新学习进度

viewProgress()：查看学习进度

calculateGrade()：计算用户的成绩

## 4.4 认证与证书管理系统设计

**4.4.1 类设计**

该子系统使用智能合约进行学习成果的自动评估，并颁发区块链认证的学习证书。

Certification 类

表示用户获得的证书信息。

属性：

certificateID（String）：证书ID

userID（String）：用户ID

courseID（String）：课程ID

issueDate（Date）：颁发日期

certificateHash（String）：区块链认证哈希值

方法：

generateCertificate()：生成认证证书

verifyCertificate(String certificateHash)：验证证书的真实性

viewCertificateDetails()：查看证书详细信息

## 4.5 用户信息维护系统设计

**4.5.1 类设计**

该子系统由管理员负责，主要维护用户数据、权限管理和账户设置。

UserMaintenance 类

用于管理员维护用户信息和权限。

属性：

maintenanceID（String）：维护ID

adminID（String）：管理员ID

userID（String）：用户ID

changeDescription（String）：更改描述

方法：

updateUserRole(String userID, String newRole)：更新用户角色

deactivateUser(String userID)：禁用用户账户

viewUserInfo(String userID)：查看用户信息

logMaintenanceActivity()：记录维护操作

**5 界面设计**

本系统由于使用 Django 框架，而在 Django 中，url 是实现页面跳转和其他用户交互功能的核心，所以本节的功能模块设计介绍以 url 为主线。

## 5.1 网站设计母板

本系统在设计前端页面时使用的“母版”的设计思想，即系统绝大多数页面的布局都继承自该母版页面，也就是说，系统的每个页面的顶部和左侧的布局和功能完全相同，只是中间部分会有布局和功能上的变化。如图5-1所示。

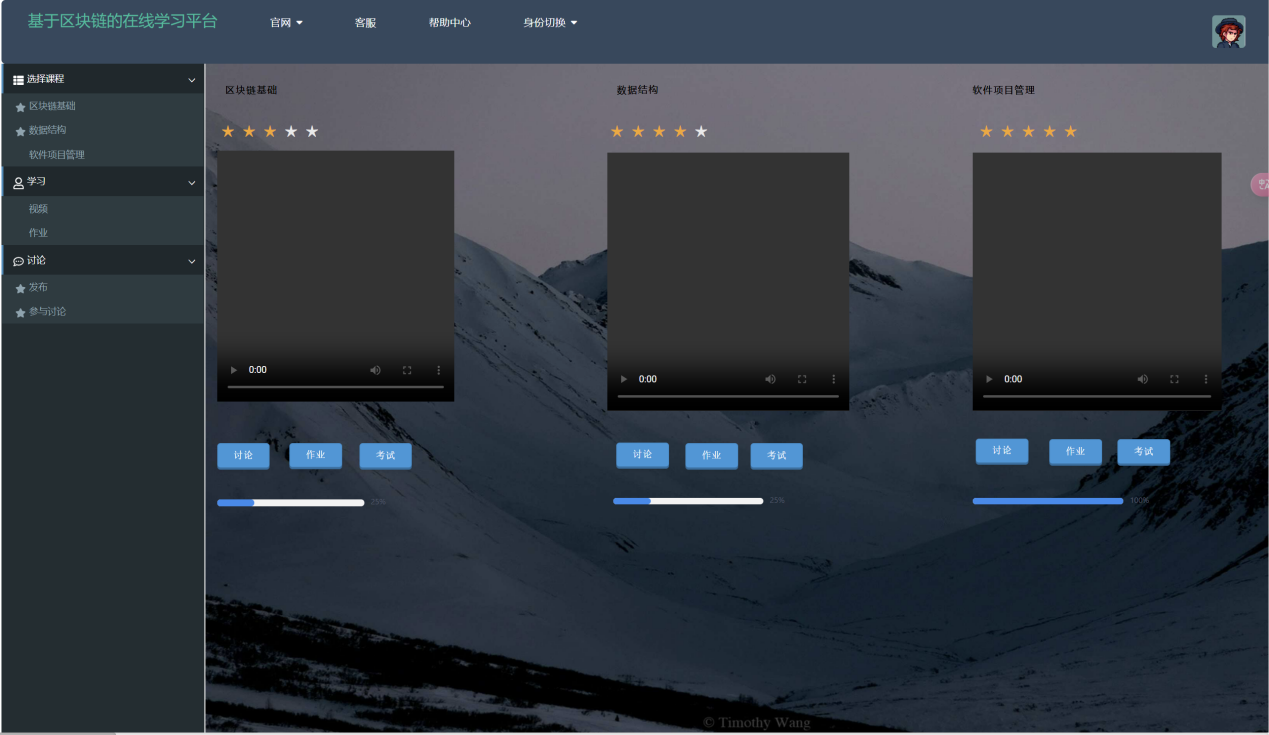


图 5-1 网站设计母版图

因此，本节之后其他功能模块的设计介绍将不包括顶部和左侧的设计功能。

**5.1.1 页面组成**

(1)顶部：页面顶部包含导航栏，显示 “首页”、“课程”、“帮助中心”和“身份切换” 等功能，右上角显示用户头像用于个人中心或账户设置。

(2)左侧：左侧为课程导航栏，分为多个课程模块，如“区块链基础”、“数据结构”和“软件项目管理”等。学习模块下包含“视频”和“作业”选项，讨论模块提供“发布”和“参与讨论”功能。

(3)主内容区域:主内容区域展示了多个课程的概要，每个课程块中有视频播放控件、课程名称、星级评分以及进度条。每个课程块下方提供“讨论”、“作业”和“考试”按钮，供用户参与课程的不同环节。进度条直观地显示了用户在课程中的学习进度。

**5.1.2 调用描述**

游客在使用系统时，点击头部的“登录”或“注册”超链接，将跳转到登录页面 /login/。用户注册成功并登录后，点击头部的“注销”超链接时，系统将会注销已登录的用户信息，并弹出“已注销”通知，页面恢复为游客使用状态。

点击头部的 Logo，将跳转到系统主页 /index/。

点击头部的个人信息查看按钮，可以跳转到 /user\_info/ 页面，在该页面用户可以查看和修改自己的个人信息。

点击导航栏最左侧的“首页”按钮，将跳转到系统主页 /index/。

点击导航栏中的“全部课程”按钮，将跳转到 /course\_catalog/ 页面，该页面展示所有的课程信息，供用户浏览和选择。

点击导航栏中的“更多”按钮，将弹出下拉菜单，包括“个人信息”、“学习计划”、“我的证书”等选项。点击“个人信息”按钮，将跳转到 /user\_info/ 页面，用户可以在该页面查看或修改个人信息；点击“学习计划”按钮，将跳转到 /learning\_plan/ 页面，在该页面用户可以查看当前的学习进度和学习目标；点击“我的证书”按钮，将跳转到 /my\_certificates/，用户可以在该页面查看和管理已获得的区块链认证证书。

在课程页面中，用户可以点击课程下方的“讨论”、“作业”或“考试”按钮，分别进入课程讨论区、作业提交页面和考试页面，进行相应的课程互动。

点击左侧功能模块的文字超链接，将跳转到相应的功能模块页面，此处不再赘述。

## 5.2 主页: /index/

**5.2.1 参数**

无。

**5.2.2 调用背景**

作为本系统的访问入口，用户在浏览器输入网址，按下回车后默认跳转至此页面。

用户登录验证通过以后，会跳转至此页面。

在本系统的任意页面点击网页左上角的网站 logo 或者导航条上的主页超链接会跳转至此页面。

**5.2.3 页面组成**

本页面采用上下分层结构。上方为轮播图区，轮播图包含一般循环播放的三张图片，每张图片播放时间约为 10 秒，用于展示平台特色、新课程推荐、学习活动等信息；轮播图下方是推荐课程展示区，该展示区以表格式的卡片布局展示热门课程或系统推荐课程。页面底部用于展示平台信息、“关于我们”模块及相关资源链接。

**5.2.4 调用描述**

在推荐课程展示区，课程信息以卡片形式呈现。点击卡片中的课程图片或课程名称，将跳转到 /course\_detail/?courseID=?，该页面展示该课程的具体信息；点击卡片中的“加入学习”按钮，将把该课程添加到用户的学习列表。

**6 接口设计**

## 6.1 用户接口

为了提供清晰、美观、易用的用户界面，本项目选用了 Django 框架作为后端支撑，并结合 HTML、CSS、JavaScript 构建了前端页面，以 Django 自带的 URL 路由系统管理页面跳转和用户交互。前端界面设计简洁明了，重点在于课程展示、学习进度管理和证书查询等功能模块，为用户提供流畅的学习体验。

## 6.2 外部接口

在数据处理方面，本项目涉及较多的数据存储及交互。后端使用 Django 搭建，数据库使用 MySQL Server 8.0.23 存储用户、课程、学习进度等相关数据，通过 Django ORM 进行数据库交互。系统的认证和通知功能通过 SMTP 协议发送邮件，利用 163 邮箱服务器进行邮件发送。

## 

## 6.3 内部接口

本系统以数据为中心，网站各模块均通过 Django ORM 模块与数据库进行交互。在页面跳转时，通过 session 管理用户状态，并通过 GET 请求、表单提交等方式传递参数，确保用户在浏览课程、管理学习进度和查看证书时，信息能够正确地加载和展示。

**7 角色授权设计**

本项目的使用角色有三类：游客、普通用户、管理员，角色授权设计如表7-1所示。

表 7-1 角色授权设计表

|  |  |  |  |
| --- | --- | --- | --- |
| 子系统名称 | 管理员 | 会员 | 游客 |
| 个人信息管理系统 | ● | ● | ○ |
| 课程管理系统 |  | ● | ● |
| 学习进度管理系统 | ● | ● | ○ |
| 认证与证书管理系统 | ● | ● |  |
| 用户信息维护系统 | ● |  |  |
| ●表示有全部权限，○表示享有权限但并非全部权限。 | | | |

**8 系统错误处理**

## 8.1 出错信息

(1)对游客、普通用户输入的各项内容进行有效性和安全性检查，避免错误输入，确保数据安全。

(2)捕获程序运行中的异常，按统一格式提供出错提示信息，便于游客和普通用户理解与操作。

(3)当游客或普通用户尝试访问权限以外的内容时，系统将自动导航至统一的错误提示页面，并引导用户返回正确的操作路径

## 8.2 故障预防与补救

(1)使用统一的权限控制机制，确保不同角色的访问权限得到合理限制，防止未经授权的访问。

(2)对程序中涉及的敏感数据进行加密处理，增强系统的安全性，减少因黑客攻击而造成的潜在危害。

(3)定期进行数据库的全量和增量备份，确保平台数据在发生故障时可以迅速恢复，保障学习数据和课程信息的完整性。

## 8.3 系统维护设计

(1)模块化与分层设计：编码时遵循模块化和分层设计原则，增强系统逻辑的清晰性和独立性，减少模块间的耦合度，提升系统的可维护性。

(2)面向对象的设计：采用面向数据和面向对象结合的设计方法，确保模块划分符合面向对象思想，便于扩展和维护。

(3)代码规范与注释：编码过程中遵循一致的命名规范，确保标识符具有明确意义，同时添加适量注释，便于后续开发人员理解代码逻辑，提高可读性和维护性。