系统设计报告

web教务系统

第4组 李泽祥

2024

目录

[项目需求分析图 2](#_Toc155645640)

[User 2](#_Toc155645641)

[course 2](#_Toc155645642)

[SC 3](#_Toc155645643)

[UI设计 3](#_Toc155645644)

[首页（登录页面） 3](#_Toc155645645)

[学生界面 5](#_Toc155645646)

[教师界面 6](#_Toc155645647)

[管理员界面 7](#_Toc155645648)

[实现方案 8](#_Toc155645649)

[前端 8](#_Toc155645650)

[首页（登录页面） 8](#_Toc155645651)

[学生界面 8](#_Toc155645652)

[教师界面 9](#_Toc155645653)

[管理员界面 10](#_Toc155645654)

[后端 10](#_Toc155645655)

[UserController 10](#_Toc155645656)

[CourseController 10](#_Toc155645657)

[ScController 11](#_Toc155645658)

[UserService、CourseService、ScService 11](#_Toc155645659)

[User、Course、Sc 11](#_Toc155645660)

[Dao层 11](#_Toc155645661)

[数据库 11](#_Toc155645662)

[user表 11](#_Toc155645663)

[course表 12](#_Toc155645664)

[sc表 12](#_Toc155645665)

[特色功能 13](#_Toc155645666)

[设计模式 15](#_Toc155645667)

[遇到的问题及解决方案 16](#_Toc155645668)

[同步异步问题 16](#_Toc155645669)

[解决方法 16](#_Toc155645670)

[小组分工 17](#_Toc155645671)

[李泽祥（组长） 17](#_Toc155645672)

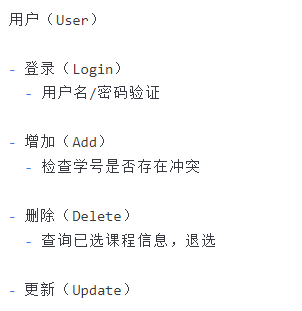
[李迅（组员） 18](#_Toc155645673)

[李申帅（组员） 18](#_Toc155645674)

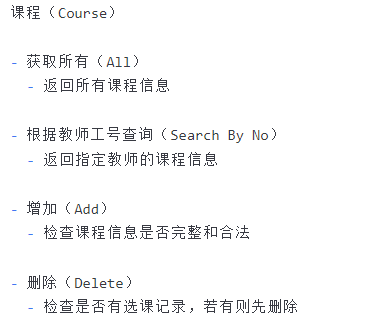
[刘冠良（组员） 18](#_Toc155645675)

# 项目需求分析图

## User



## course



## SC



# UI设计

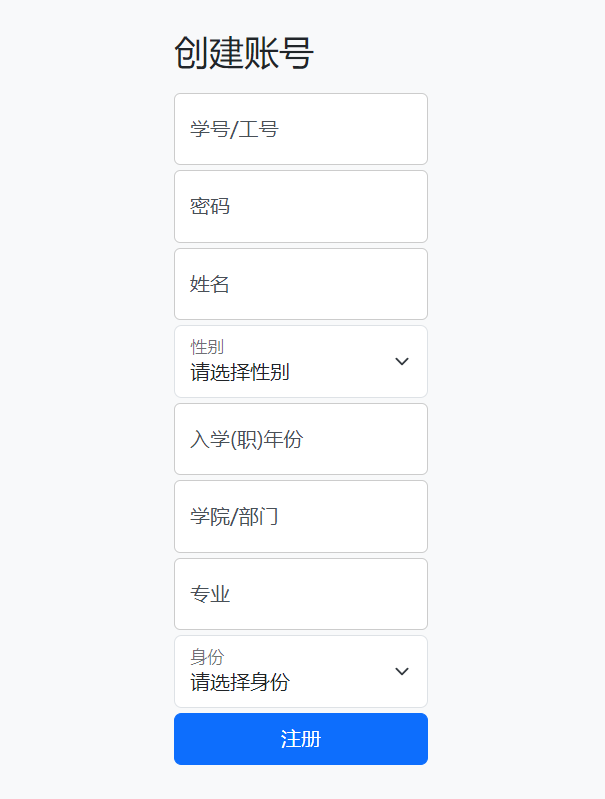
使用bootstrap+前端3件套

## 首页（登录页面）

* 登录页面包含一个登录表单，用户可以输入用户名和密码进行登录。
* 页面上方或下方可以显示欢迎信息，并有一个“注册”的链接。
* 使用背景图片、适当的颜色和字体，以保持页面的美观和一致性。

****

**登录**

****

**注册**

## 学生界面

* **信息查询**：显示学生个人信息，包括姓名、性别、入学年份、专业、学院等。提供一个“修改密码”的链接或按钮，供学生修改密码。
* **查询已选课程信息**：显示学生已选的课程列表，并提供“退课”的选项。
* **查询所有开设课程的信息**：显示所有课程的信息，包括课程名称、学分、学时等。
* **查询已选课程的成绩**：显示学生已选课程的成绩列表。



成绩查询



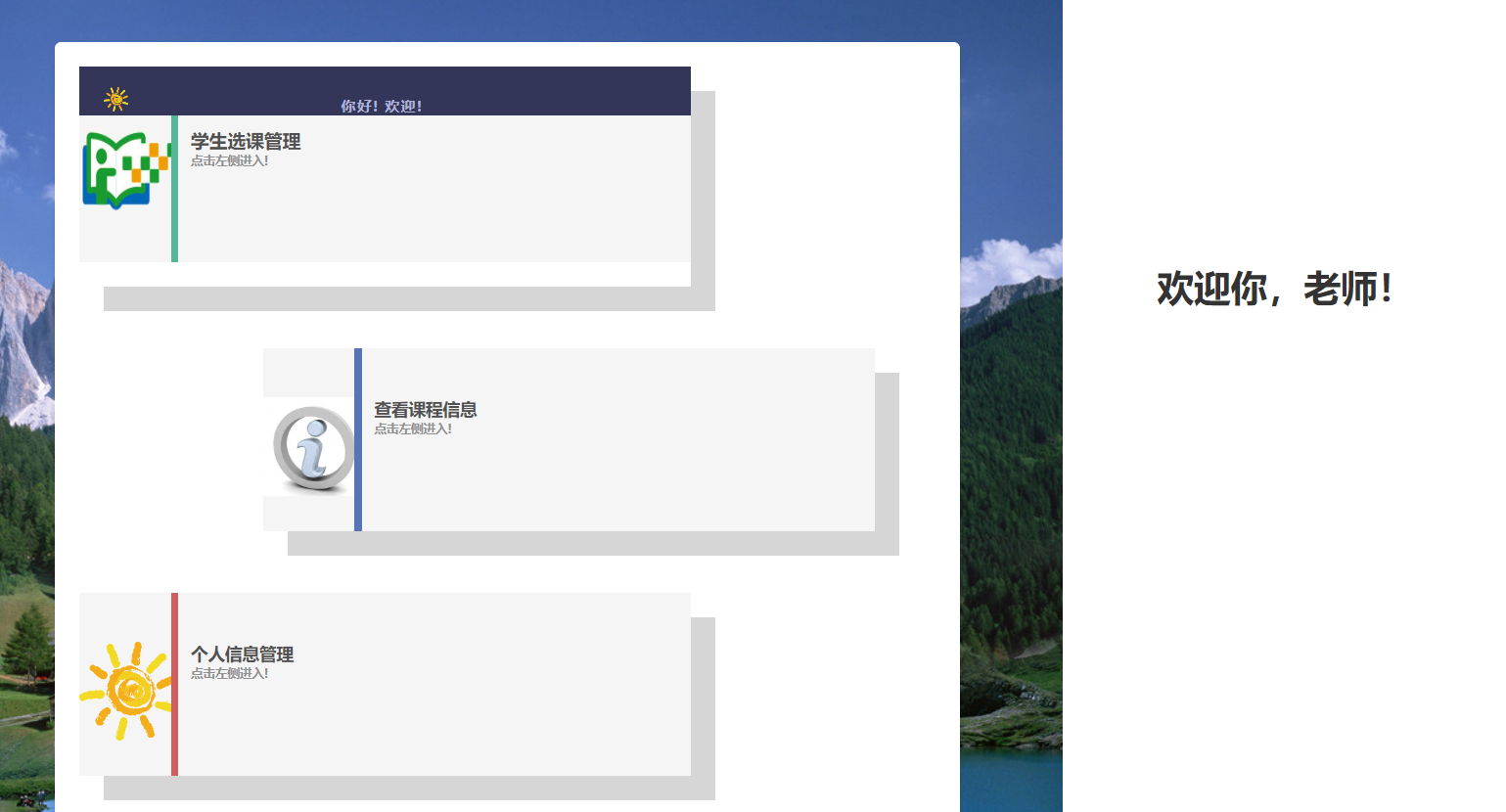
选课



退课

## 教师界面

* **信息查询**：显示教师个人信息，包括姓名、职称等。提供一个“修改密码”的链接或按钮，供教师修改密码。
* **查看学生选此教师开设课程的详情**：显示选此教师课程的学生列表，包括学生个人信息和已选学生人数等。
* **录入成绩**：为已选此教师课程的学生录入成绩。

****

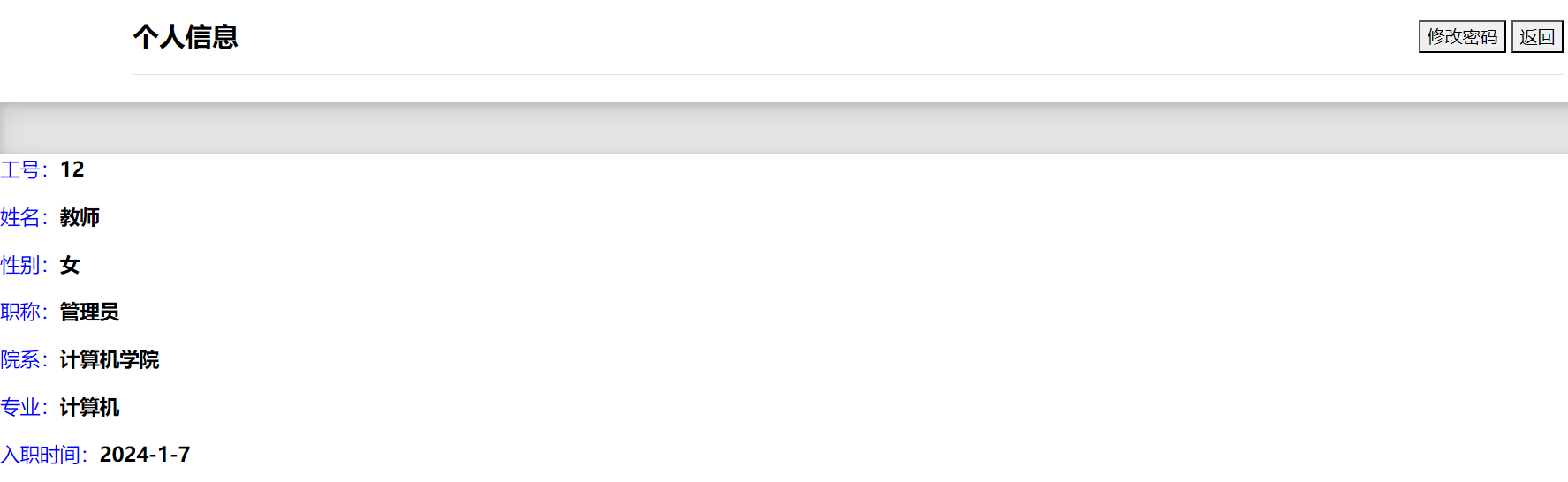
**教师首页**

****

**选课+修改成绩**

****

**查看课程信息**

****

**查看教师个人信息**

## 管理员界面

* **学生管理**：包括“录入学生”、“删除学生”等操作，设计简洁明了的操作按钮和确认提示。
* **教师管理**：包括“录入教师”等操作，同样需要简洁明了的操作按钮和确认提示。
* **课程管理**：包括“创建课程”、“更新课程”、“删除课程”等操作，提供相应的表单和操作按钮。
* **选课管理**：包括“删除选课记录”等操作，设计相应的操作按钮和确认提示。



管理员首页



创建课程



查看用户



查看课程、删除课程

# 实现方案

该项目实现需要前后端分离，前端使用Bootstrap框架+前端3件套进行开发，后端采用RESTful API架构，使用python语言编写，并使用flask框架进行开发。数据库采用MySQL。

具体实现方案如下：

## 前端

前端页面使用HTML、CSS、JavaScript和Bootstrap框架编写，与后端通过RESTful API进行交互。主要页面包括：

### 首页（登录页面）

提供登录表单，用户输入用户名和密码进行登录。

### 学生界面

#### 信息查询

提供查询个人信息、专业信息、学院信息的功能，并且允许修改密码。

#### 查询已选课程信息，退课

提供查询已选课程信息的功能，并且允许学生进行退课操作。

#### 查询所有开设课程的信息

提供查询所有开设课程信息的功能。

#### 查询已选课程的成绩

提供查询已选课程成绩的功能。

### 教师界面

#### 信息查询

提供查询个人信息的功能，并且允许修改密码。

#### 查看学生选此教师开设课程的详情

提供查看学生选此教师开设课程的详情的功能，包括各个学生的个人信息，以及已选学生人数等。

#### 为此教师开设课程中的学生，录入此门课程的成绩

提供为此教师开设课程中的学生录入成绩的功能。

### 管理员界面

#### 学生管理

提供录入学生、删除学生的功能。

#### 教师管理

提供录入教师的功能。

#### 课程管理

提供创建课程、更新课程、删除课程的功能。

#### 选课管理

提供删除选课记录的功能。

## 后端

后端使用Java语言编写，采用Spring Boot框架搭建RESTful API服务。主要实现类包括：

### UserController

实现用户登录、添加用户、删除用户、更新用户信息等功能。

### CourseController

实现获取所有课程信息、根据教师工号查看课程信息、添加课程、删除课程、更新课程等功能。

### ScController

实现删除选课信息、更新选课信息、根据学号得到选课信息、根据课程得到选课信息等功能。

### UserService、CourseService、ScService

分别实现对用户、课程、选课信息的操作方法，并与数据库进行交互。

### User、Course、Sc

分别对应用户、课程、选课信息的数据结构，包含各自的属性和方法。

# Dao层

各自对应着用户、课程、选课信息的数据表，实现对数据表的增、删、改、查等基本操作。

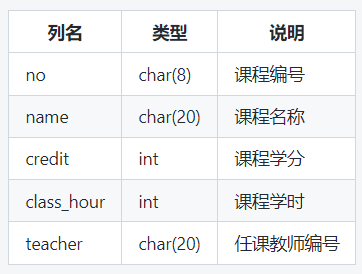
# 数据库

数据库采用MySQL，包含以下数据表：

## user表



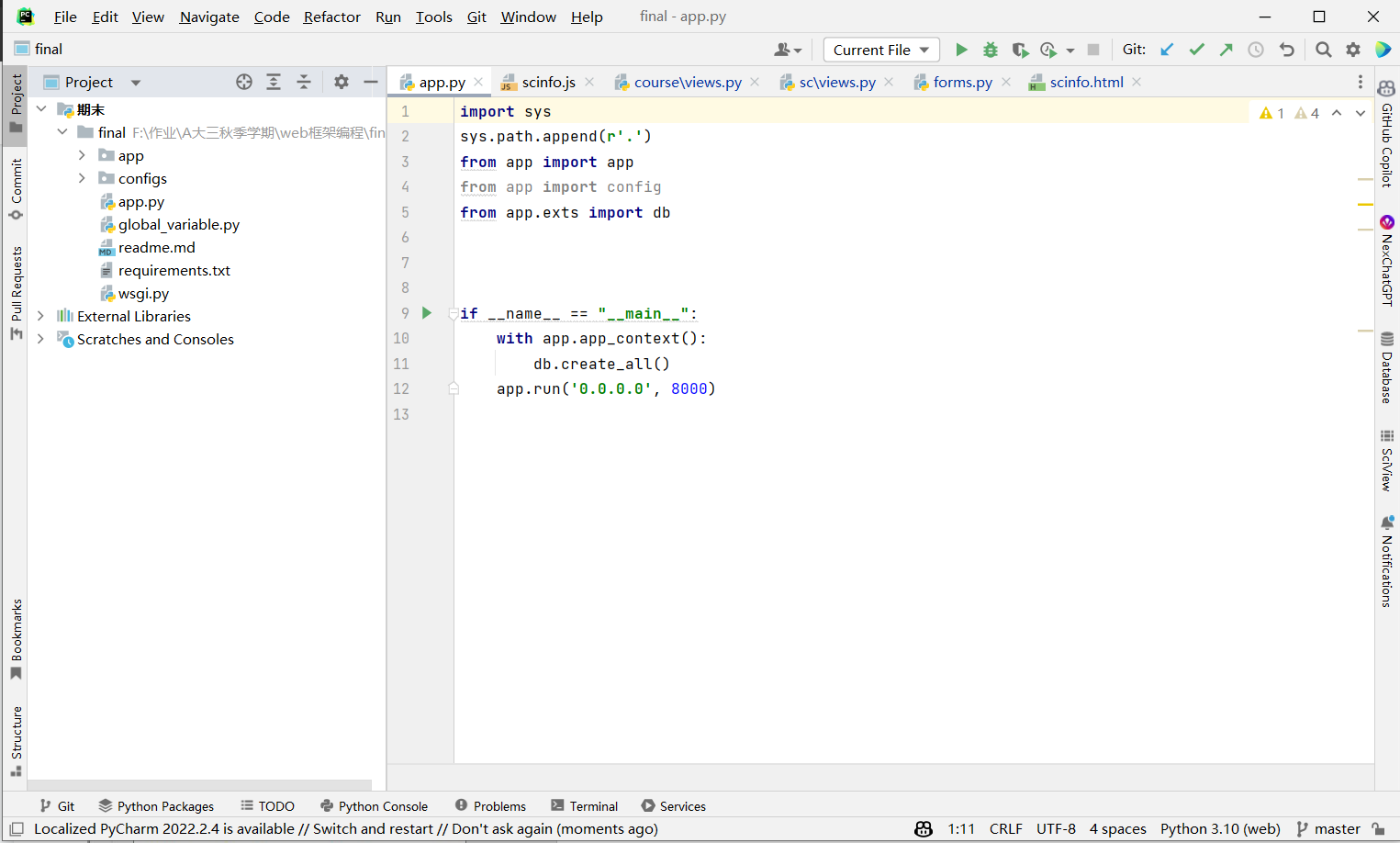
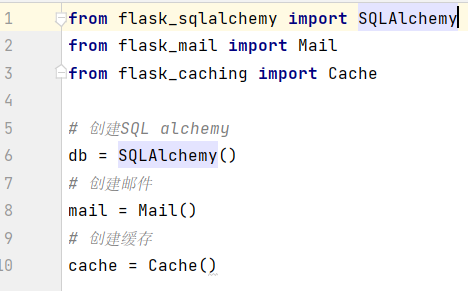
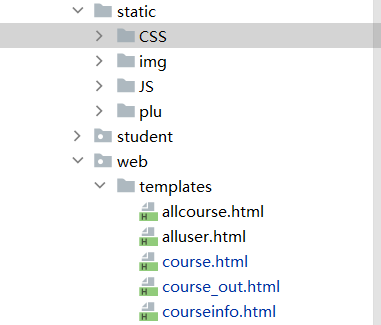
## course表



## sc表



# 特色功能

1. 轻量级的Flask框架：Flask是一个精简而强大的Web框架，它只包含必要的功能，使得项目开发变得简单高效。尤其适用于小型项目和快速原型开发。  
   
2. RESTful架构支持：Flask天生支持RESTful架构，这意味着API接口的设计更加简单明了。通过定义资源和使用HTTP方法，可以轻松实现对数据的增删改查操作。  
   
3. 灵活的数据库连接：Flask使用Python中的SQLAlchemy库，通过简单的配置即可实现与多种数据库的连接，无需添加繁琐的依赖。这使得在项目中选择和切换数据库变得非常方便。  
   
4. Bootstrap框架美化页面：前端使用了流行的Bootstrap框架，它提供了丰富的页面组件和样式，使得页面设计和开发变得更加容易。通过使用Bootstrap，可以轻松实现各种页面元素的美化效果，提升用户体验。  
   
5. 前端3件套配合使用：项目使用前端三件套（HTML、JS、CSS）相互配合，共同构建出丰富、功能完整的网页。HTML负责页面结构，JS实现交互逻辑，CSS负责页面样式。这种配合使用能够提高开发效率，并且使得前端代码更加易于维护和扩展。  
   

# 设计模式

本项目使用了MVC设计模式来组织和管理代码。

1. init.py：init.py文件是Flask应用的入口文件，也是一个包的标识文件。在这个文件中，你可以创建Flask应用实例，配置应用的参数，以及注册蓝图Blueprints。在MVC设计模式中，init.py扮演着控制器Controller的角色，负责接收和处理用户的请求，并将请求转发给相应的视图函数。
2. forms.py：forms.py文件通常用于定义Web表单Form，其中包含了用户输入的字段和验证规则。在MVC设计模式中，forms.py扮演着模型Model的角色，表示数据的结构和验证规则。通过定义表单类，你可以方便地使用Flask-WTF库来处理表单的验证和数据处理。
3. models.py：models.py文件用于定义数据模型Model，包括数据库表的结构和之间的关系。在MVC设计模式中，models.py扮演着模型Model的角色，负责处理和操作数据库的数据。你可以使用SQLAlchemy等数据库工具来定义模型类，通过模型类可以方便地进行数据库操作，如增删改查等。
4. views.py：views.py文件包含了视图函数View，用于处理用户请求并返回相应的响应。在MVC设计模式中，views.py扮演着视图View的角色，负责接收控制器传递过来的请求，并根据请求的参数和数据进行相应的处理和渲染。视图函数通常会调用模型层的方法来获取数据，并将数据传递给模板进行渲染，最后返回给用户。

通过这种MVC模式实现了代码的分层和解耦，提高了代码的可维护性和可扩展性。更好地组织Flask项目，使其结构清晰、易于维护，并且方便地进行功能的扩展和修改。

# 遇到的问题及解决方案

## 同步异步问题

JavaScript是一门单线程语言。所以当fun1()中存在一些访问接口并操作获取的返回数据这一类花费一定时间的代码时，fun2()可以不等fun1()完全执行完就开始执行。即挂起fun1()先运行fun2()，等到fun1()得到结果后，再执行。

同步任务是指在主线程上排队执行的任务，只有前一个任务执行完毕，才能继续执行下一个任务，当我们打开网站时，网站的渲染过程，比如元素的渲染，其实就是一个同步任务。

异步任务是指不进入主线程，而进入任务队列的任务，只有任务队列通知主线程，某个异步任务可以执行了，该任务才会进入主线程，当我们打开网站时，像图片的加载，音乐的加载，其实就是一个异步任务。

异步机制通过事件循环实现的。 JavaScript是单线程运行的语言，也就是说，它只有一个主线程来执行所有的代码。当需要执行异步操作时，如进行网络请求、文件读写等，主线程无法立即获得结果，因此会将这些操作放在任务队列中，等待异步操作完成后再进行处理。

事件循环就是负责处理任务队列中的任务，并将任务分配给主线程执行的机制。事件循环从任务队列中取出一个任务，将其分配给主线程执行，直到执行完毕或者遇到I/O等异步操作，然后再去取下一个任务执行。在执行异步操作时，主线程会将回调函数放入队列中，等待事件循环机制执行完主线程当前的任务后，再从队列中取出回调函数执行。

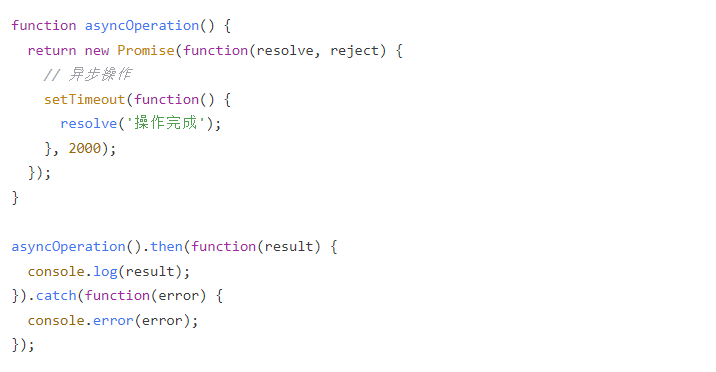
而在我的开发过程中就经常遇到fun1();fun2();而fun2()要用到fun1()网络请求得到的数据。

### 解决方法

1. 回调函数：回调函数是异步编程最基本的形式。通过将一个函数作为参数传递给另一个函数，在异步操作完成后调用该函数来处理结果。例如，可以使用回调函数在异步请求完成后执行某些操作。



1. Promise对象：Promise对象是对回调函数的封装和管理，用于更好地处理异步操作。Promise有三个状态：pending（进行中）、fulfilled（已成功）和rejected（已失败）。通过使用then()和catch()方法，可以在异步操作完成后执行相应的操作。



1. async/await：async/await是ES2017引入的一种更直观、更简洁的异步编程方式。async函数返回一个Promise对象，其中可以使用await关键字来等待异步操作完成，并以同步的方式编写代码。



# 小组分工

## 李泽祥（组长）

1. 完成需求分析
2. 安排各个小组成员工作量
3. 开发登录、注册页面前端
4. 开发后端接口
5. 撰写系统设计、测试报告

## 李迅（组员）

1. 完成教师相关业务逻辑
2. 对教师相关的后端接口进行debug
3. 撰写系统设计报告

## 李申帅（组员）

1. 完成课程管理员相关业务逻辑
2. 对课程相关管理员相关的后端接口进行debug
3. 撰写系统设计报告

## 刘冠良（组员）

1. 完成学生相关业务逻辑
2. 对学生相关的后端接口进行debug
3. 撰写系统设计报告