|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 《数据库原理》实验报告 | | | | |
| 题目：实验八  数据库综合实验 | 学号 | 姓名 | 班级 | 日期 |
| **2022302543**  **2022302546** | **屈胤龙**  **李昊轩** | **10012202**  **10012202** | **2024/12/19**  **2024/12/19** |

1. 分工详细内容：本项目由两位同学合作完成

屈胤龙：项目需求分析设计，设计预期的功能，数据库表格设计，前端网页的链接，数据库与后端的架构设计与实现，触发器的设计优化，使用文档撰写。

李昊轩：项目需求分析设计，数据库概念设计，ER图绘制，本地环境的搭建，前端的界面美化，后端代码的更改调整，演示视频的展示，代码修复工作。

为什么不使用金仓数据库：金仓数据库的实现框架非常的固定，没有办法灵活的进行修改，在前端设计方向上设计上可以参考的资料很少，加上我和组员在vue以及flask框架上有一些初步的了解，作中采用的是以下的设计

整体设计：

前端：编辑器：vscode

框架：vue

后端：编辑器：pycharm

语言：python

框架：flask

数据库：MySQL

1. 线下实验：实验内容、步骤以及结果

实验内容：通过完成从用户需求分析、概念结构设计、逻辑结构设计等一系列的数据库设计到编程、调试和应用等全过程，进一步理解和掌握教材中的相关内容。

实验题目：调查驾校的相关业务，根据具体情况设计驾校考试预约管理系统。主要功能有：

1. 基础信息管理：教练信息，学员信息，场地信息，考试信息等。

2. 考试预约管理：考试预约登记，查询与修改等。

3. 练车场地预约管理：练车场地预约等。

**需求分析：**

#### 数据库需求：

**教练表**：包括教练的基本信息（姓名、性别、联系电话）

**学员表**：包括学员的基本信息（姓名、性别、年龄、电话、地址）

**场地表**：包括场地信息（场地编号、名称、容量、位置）

**考试表**：包括考试信息（考试编号、科目、时间、地点、状态）

**考试预约表**：包括学员预约的信息（预约编号、学员编号、考试编号、预约时间）

**练车场地预约表**：学员预约练车场地的记录（场地编号、学员编号、预约时间）

#### 界面需求：

**登录界面**：管理员、用户的登录入口

**用户主页**：学员查看和管理自己的考试、课程、练车场地预约信息。

**管理员主页**：管理员查看并管理学员、教练、场地、考试等信息。

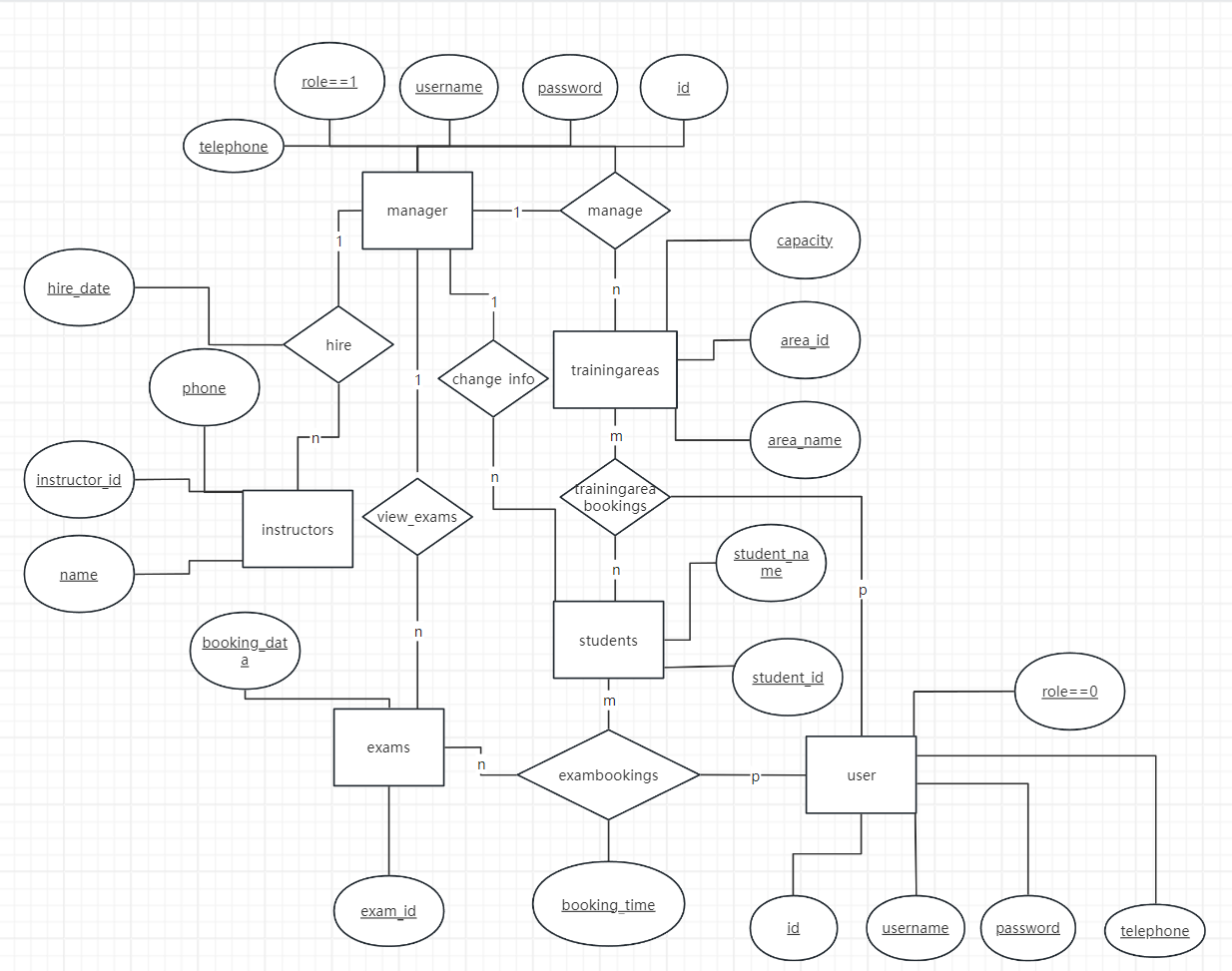
#### 权限管理：

用户：可以预约考试、场地、查看自己的预约信息。

**管理员**：管理所有信息，包括学员、教练、场地、考试安排。

**概念设计：**

ER图如下所示



**逻辑模式：**

关系模式：

user(**id**, username, password, telephone, role)

user\_msg(**id**, real\_name, sex, age, mail, phone, user\_name)

students(**student\_id**, student\_name, gender, phone, registration\_date)

trainingareas(**area\_id**, area\_name, capacity, avail\_capacity)

instructors(**instructor\_id**, name, phone, hire\_date)

trainingareabookings(**student\_id**, **area\_id**, areabooking\_date, areabooking\_time, instruct\_name)

exams(**exam\_id**, exam\_name, exam\_date, exam\_time, location, total\_slots, available\_slots)

表设计：

### 1. user 表

| 字段名 | 类型 | 约束 | 索引 |
| --- | --- | --- | --- |
| id | int unsigned | NOT NULL AUTO\_INCREMENT, PRIMARY KEY | PRIMARY KEY |
| username | varchar(20) | NOT NULL | UNIQUE, BTREE |
| password | varchar(20) | NOT NULL | BTREE |
| telephone | varchar(20) | NOT NULL | BTREE |
| role | int | NOT NULL | BTREE |

**范式**: 满足 3NF，所有非主键字段直接依赖于主键，没有传递依赖。

### 2. user\_msg 表

| 字段名 | 类型 | 约束 | 索引 |
| --- | --- | --- | --- |
| id | int unsigned | DEFAULT NULL, PRIMARY KEY (id) | userid (BTREE), |
| real\_name | varchar(50) | NOT NULL | BTREE |
| sex | varchar(50) | NOT NULL | BTREE |
| age | int | NOT NULL | BTREE |
| mail | varchar(50) | NOT NULL | BTREE |
| phone | varchar(50) | NOT NULL | BTREE |
| user\_name | varchar(50) | NOT NULL | BTREE |

**范式**:**3NF**: 满足 3NF，没有传递依赖。

### 3. trainingareabookings 表

| 字段名 | 类型 | 约束 | 索引 |
| --- | --- | --- | --- |
| student\_id | int | NOT NULL | PRIMARY KEY (student\_id, area\_id), |
| area\_id | int | NOT NULL | trainingareabookings\_ibfk\_2 (area\_id) |
| areabooking\_date | date | DEFAULT NULL |  |
| areabooking\_time | time | DEFAULT NULL |  |
| instruct\_name | varchar(45) | DEFAULT NULL |  |

**范式**:**3NF**: 满足 3NF，所有字段直接依赖于主键，没有传递依赖。

### 4. students 表

| 字段名 | 类型 | 约束 | 索引 |
| --- | --- | --- | --- |
| student\_id | int | NOT NULL AUTO\_INCREMENT, PRIMARY KEY | PRIMARY KEY (student\_id) |
| student\_name | varchar(100) | NOT NULL |  |
| gender | varchar(10) | DEFAULT NULL |  |
| phone | varchar(15) | DEFAULT NULL |  |
| registration\_date | date | DEFAULT NULL |  |

**范式**:**3NF**: 满足 3NF，没有传递依赖。

### 5. trainingareas 表

| 字段名 | 类型 | 约束 | 索引 |
| --- | --- | --- | --- |
| area\_id | int | NOT NULL AUTO\_INCREMENT, PRIMARY KEY | PRIMARY KEY (area\_id) |
| area\_name | varchar(100) | NOT NULL |  |
| capacity | int | DEFAULT NULL |  |
| avail\_capacity | int | DEFAULT NULL |  |

**范式3NF**: 满足 3NF，没有传递依赖。

### 6. instructors 表

| 字段名 | 类型 | 约束 | 索引 |
| --- | --- | --- | --- |
| instructor\_id | int | NOT NULL AUTO\_INCREMENT, PRIMARY KEY | PRIMARY KEY (instructor\_id) |
| name | varchar(100) | NOT NULL |  |
| phone | varchar(15) | DEFAULT NULL |  |
| hire\_date | date | DEFAULT NULL |  |

**范式**:**3NF**: 满足 3NF，没有传递依赖。

### 7. exambookings 表

| 字段名 | 类型 | 约束 | 索引 |
| --- | --- | --- | --- |
| student\_id | int | NOT NULL DEFAULT '0' | PRIMARY KEY (student\_id, exam\_id), |
| exam\_id | int | NOT NULL DEFAULT '0' | exambookings\_ibfk\_2 (exam\_id) |
| booking\_date | date | DEFAULT NULL |  |
| status | enum('Booked','Cancelled','Completed') | DEFAULT 'Booked' |  |
| booking\_time | time | DEFAULT NULL |  |

**范式**:**3NF**: 满足 3NF，所有字段直接依赖于主键，没有传递依赖。

分析来看，数据库结构 符合第三范式 ，已经消除了部分依赖和传递依赖，结构设计非常规范

**物理设计：**

触发器设计：

trg\_exam\_booking\_limit：向 exambookings 表插入新记录时，触发器首先会查询 exams 表中相应考试的 available\_slots（剩余名额）。如果该考试的剩余名额小于等于 0，则会通过 SIGNAL SQLSTATE '45000' 抛出一个异常

trg\_exam\_booking\_insert：当在 exambookings 表插入一条新的考试预约记录时，触发器会更新 exams 表中相应的考试，减少该考试的 available\_slots（剩余名额）数目。这样保证了每次成功的预约都能够减少对应考试的可预约名额。

trg\_instructor\_prevent\_duplicate：用于 instructors 表，确保不会插入重复的 phone（电话号码）。如果在插入新教练记录时，发现电话号码已经存在，触发器会通过 SIGNAL SQLSTATE '45000' 抛出异常，阻止插入并返回消息

trg\_area\_booking\_limit：当向 trainingareabookings 表插入新记录时，触发器首先会查询 trainingareas 表中相应的训练区域的 avail\_capacity（剩余容量）。如果该区域的剩余容量小于等于 0，则会抛出一个异常，阻止插入操作并返回消息

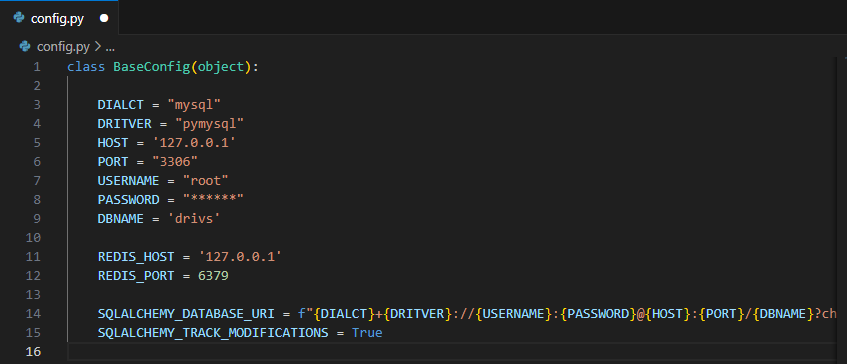
trg\_area\_booking\_insert：表插入一条新的训练场地预约记录时，触发器会更新 trainingareas 表中相应的训练区域，减少该区域的 avail\_capacity（剩余容量）

索引设计：为了提高查找的速度，对于每一个表都对其主键建立unique索引，在其余每个属性上都建立了normal索引，索引方法选择BTREE

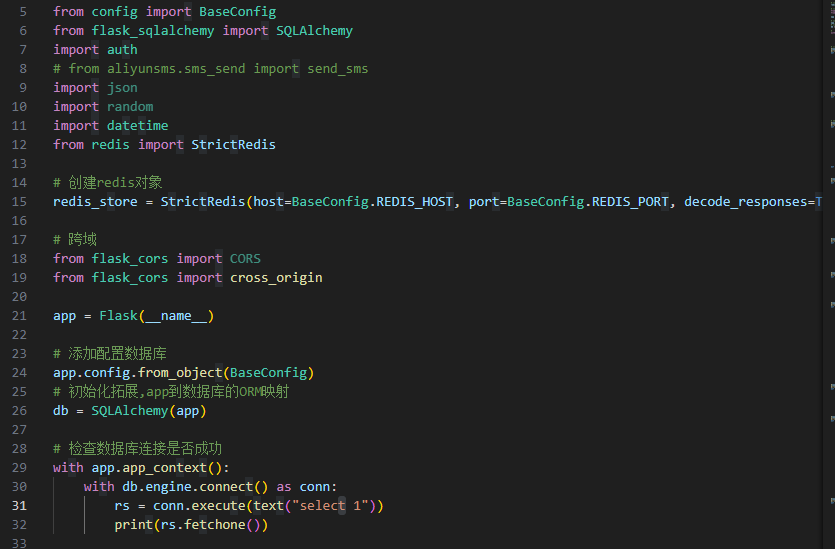
**系统设计以及代码实现：**

后端链接数据库：

在config.py文件中创建配置数据库连接的类



在主文件app.py中导入配置



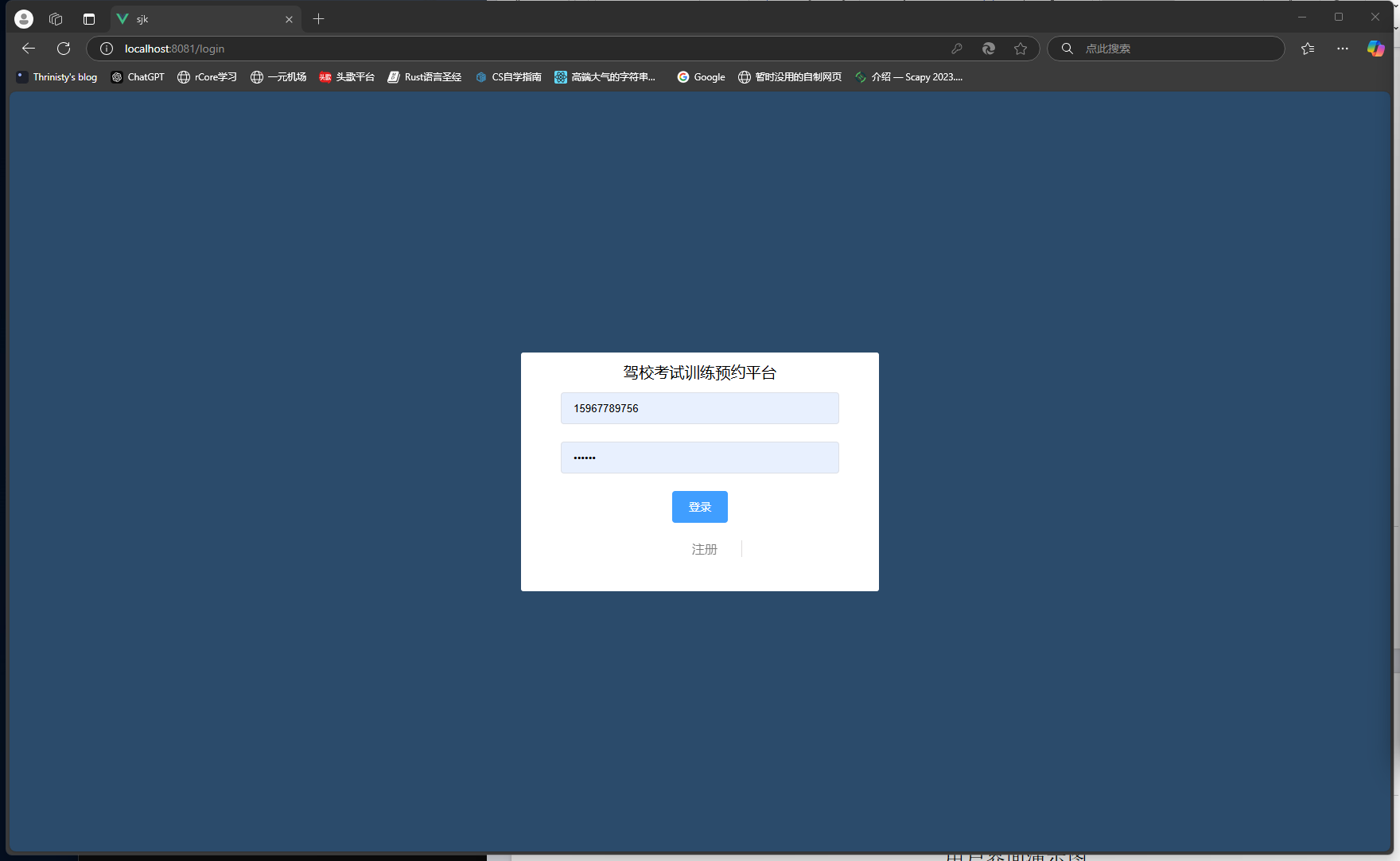
前端页面设计：运用主流的Vue框架设计，主页面有三个，登录页面，用户页面，管理员页面

界面功能演示：（详见演示视频）

登录界面：**登录界面**：管理员、用户的登录入口

登录页面使用的是电话号码与密码的组合登录方式

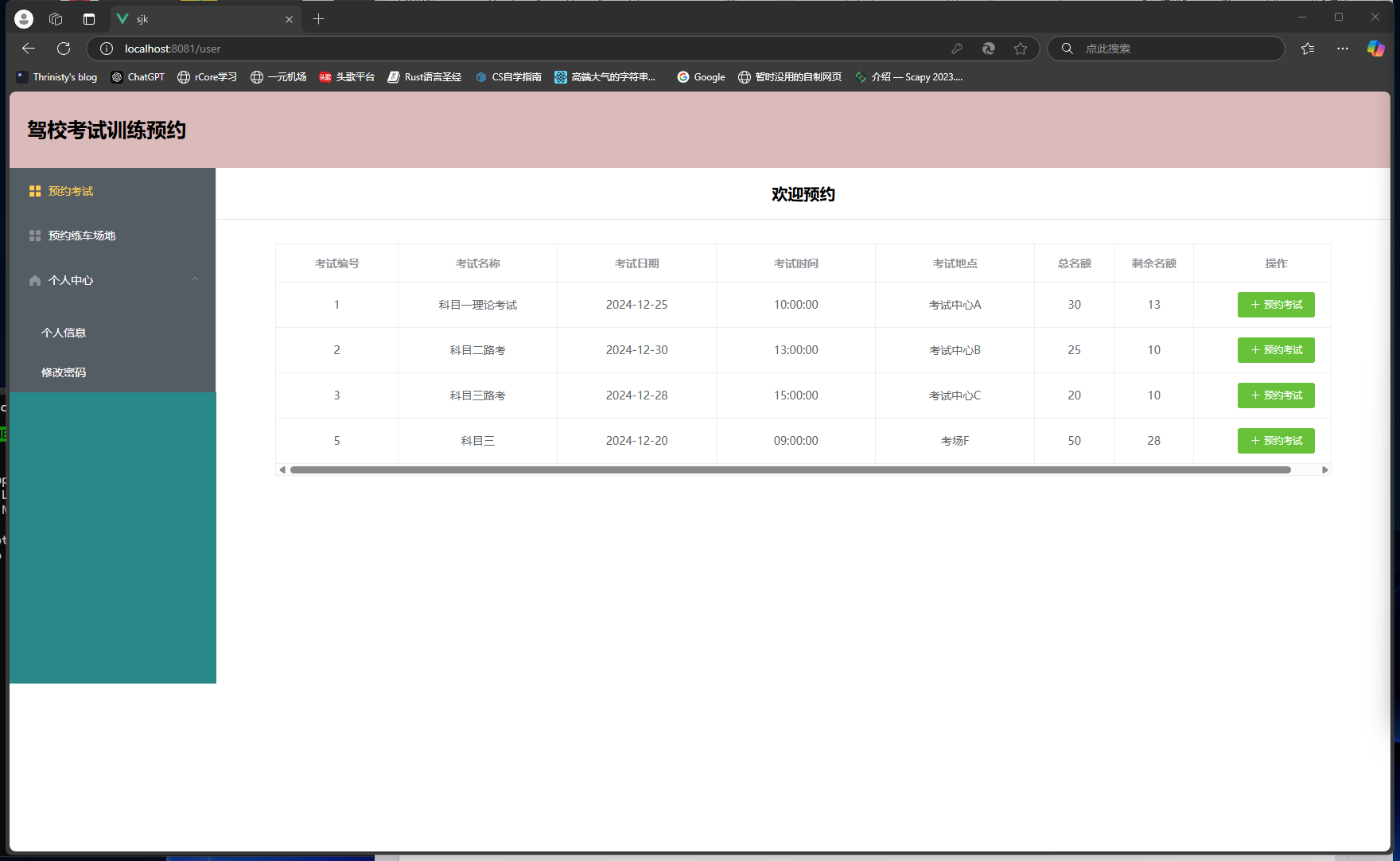
通过了链接了的数据库可以获取到对应的电话以及适配的密码用于用户的验证。



登录界面演示图

用户界面：**用户主页**：学员查看和管理自己的考试、课程、练车场地预约信息

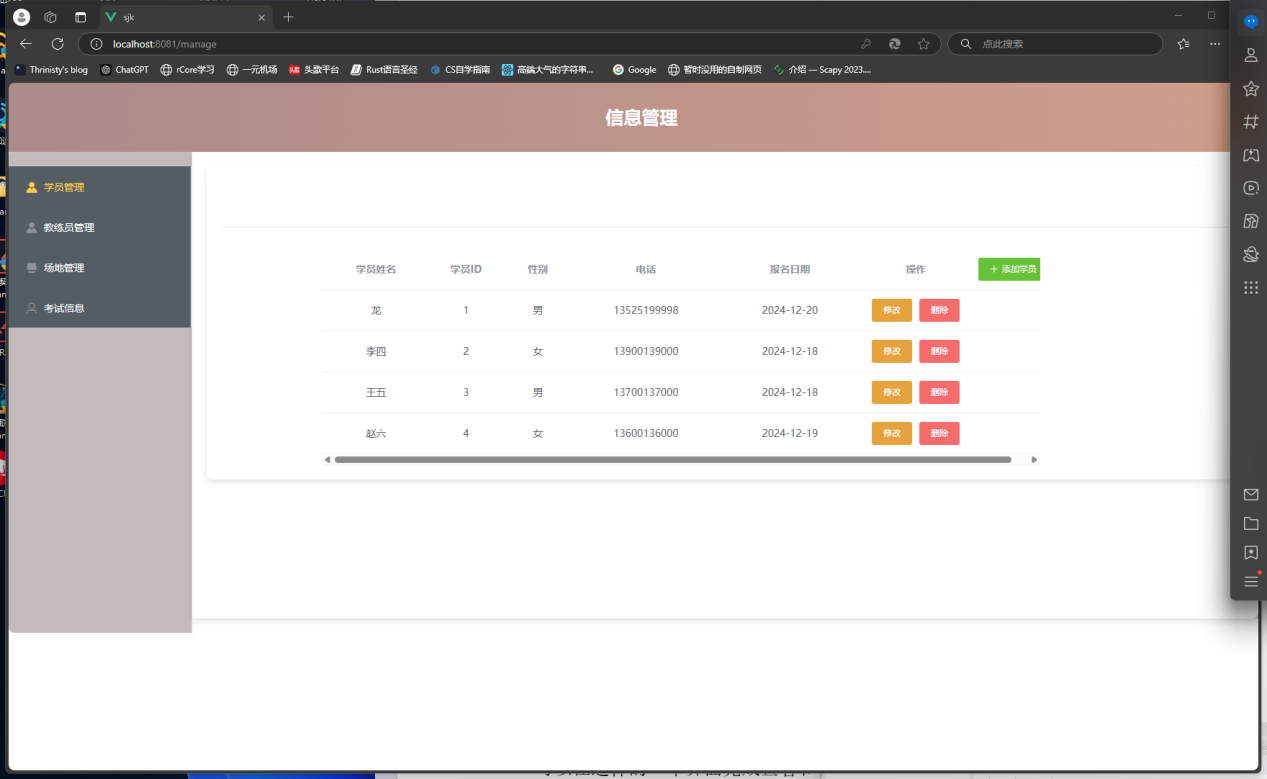
用户界面主要有如下子界面：预约场地，预约考试，以及个人信息与密码修改



用户界面演示图

可以在这样的一个界面完成查看和管理自己的考试、课程、练车场地预约信息的动作，后端会根据api接口用以修改连接了的数据库，在“个人中心”里用户可以查看和修改自己的个人信息以及密码，子级页面的功能演示详见演示视频。

管理员界面：**管理员主页**：管理员查看并管理学员、教练、场地、考试信息



管理员界面演示图

管理员界面主要有如下子界面：场地信息管理，考试信息查看页面，学员信息与教练信息的管理界面，分别可以对应各个表格的增删改查，具体功能详见演示视频。

**程序源代码：**

见打包压缩文件

1. 实验收获以及心得体会

心得体会：通过这一次课程设计，我不仅复习并深入研究了数据库中的 SQL 知识，还提升了自己的编程能力，亲自设计了页面并实现了功能。

在整个系统设计与实现的过程中，我熟悉了后端的 Flask 框架，并且熟练掌握了 MySQL 数据库的运用，例如通过索引加快查询速度，通过视图扩展显示属性，以及利用触发器实现表之间的联动操作。

设计的数据库已基本满足需求。同时，通过制作网站来完成数据库课设，我对网站设计与开发过程有了更加深入的理解。