

**《数据库系统原理》**

|  |  |  |
| --- | --- | --- |
| **作业名称：** | **系统详细设计说明书** | |
| **姓 名：** | | **戴斌斌** | |
| **学 号：** | | **20281239** | |
| **专 业：** | | **物联网工程2001** | |

2023年6月17日

目录

[一、系统功能概述 3](#_Toc137938553)

[二、系统功能模块结构 3](#_Toc137938554)

[三、系统界面设计 4](#_Toc137938555)

[3.1未登录界面 4](#_Toc137938556)

[3.2注册、登录界面 5](#_Toc137938557)

[3.3管理员界面 6](#_Toc137938558)

[3.4顾客界面 6](#_Toc137938559)

[3.5商家界面 8](#_Toc137938560)

[四、系统物理模型 9](#_Toc137938561)

[4.1表 9](#_Toc137938562)

[4.2视图 11](#_Toc137938563)

[4.3索引 13](#_Toc137938564)

[五、系统安全体系设计 13](#_Toc137938565)

[六、系统运行环境与部署结构 14](#_Toc137938566)

[七、源代码列表及说明 14](#_Toc137938567)

# 一、系统功能概述

本次课设中我开发了一个网络点餐系统，支持管理员、顾客和商家三种用户进行登录，不同的用户具有不同的功能。

实现的基础功能有：

1. 顾客、商家账号的注册、登录和登出。
2. 账号、密码的存在性和正确性检测、违法数据提示。

管理员功能有：

1. 各个表数据的查看、删除和更新
2. 用户密码的重置
3. 用户数据查询功能
4. 数据库的备份和恢复

顾客用户功能有：

1. 在各个商家点餐加入购物车
2. 进入购物车对餐品进行添加修改
3. 查看历史订单以及订单详情
4. 查看、更新个人信息

餐厅用户功能有：

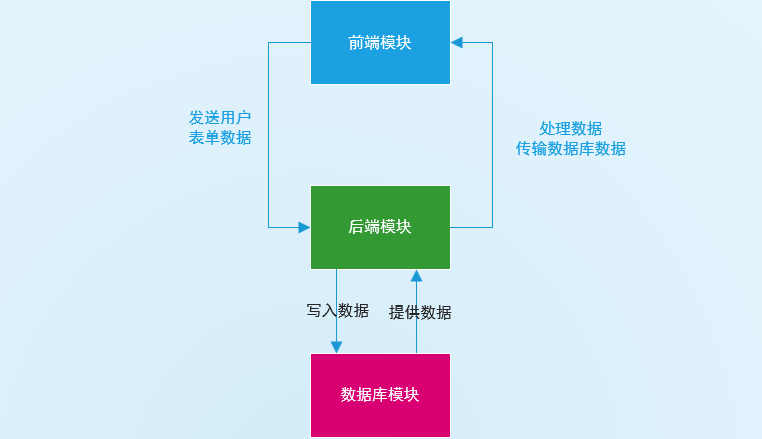
1. 添加、删除菜品
2. 更新菜品数据
3. 查看历史订单以及订单详情
4. 查看和更新个人信息

总的来说，本次开发了一个较完整的网络点餐系统，支持数据查看，菜品下单、菜品添加等基础功能，可支持多用户（商家、顾客）的并发操作。

# 二、系统功能模块结构

系统主要分为前端模块、web服务器模块、数据库模块。

* 前端模块：基础界面使用html语言，使用了bootstrap框架加速页面的开发，动态效果以及一些错误信息的提示使用javascript语言，其中主要使用的为jQuery库。前后端的数据传递使用的时jQuery封装的Ajax方式。
* 后端模块：后端主要使用python语言以及flask web框架开发，主要功能为处理前端的请求，从数据库中读取数据进行相关操作后返回结果给前端。以及获取前端数据，进行分析处理后写入数据库中，并完成展示。起到前端和数据库之间的桥梁作用
* 数据库模块：数据库采用的是mysql数据库，对使用的数据进行管理，python中可以使用pyodbc库来对该数据库进行操作，完成前端操作和数据库的联动。



具体的功能在用户手册和使用手册中进行说明

# 三、系统界面设计

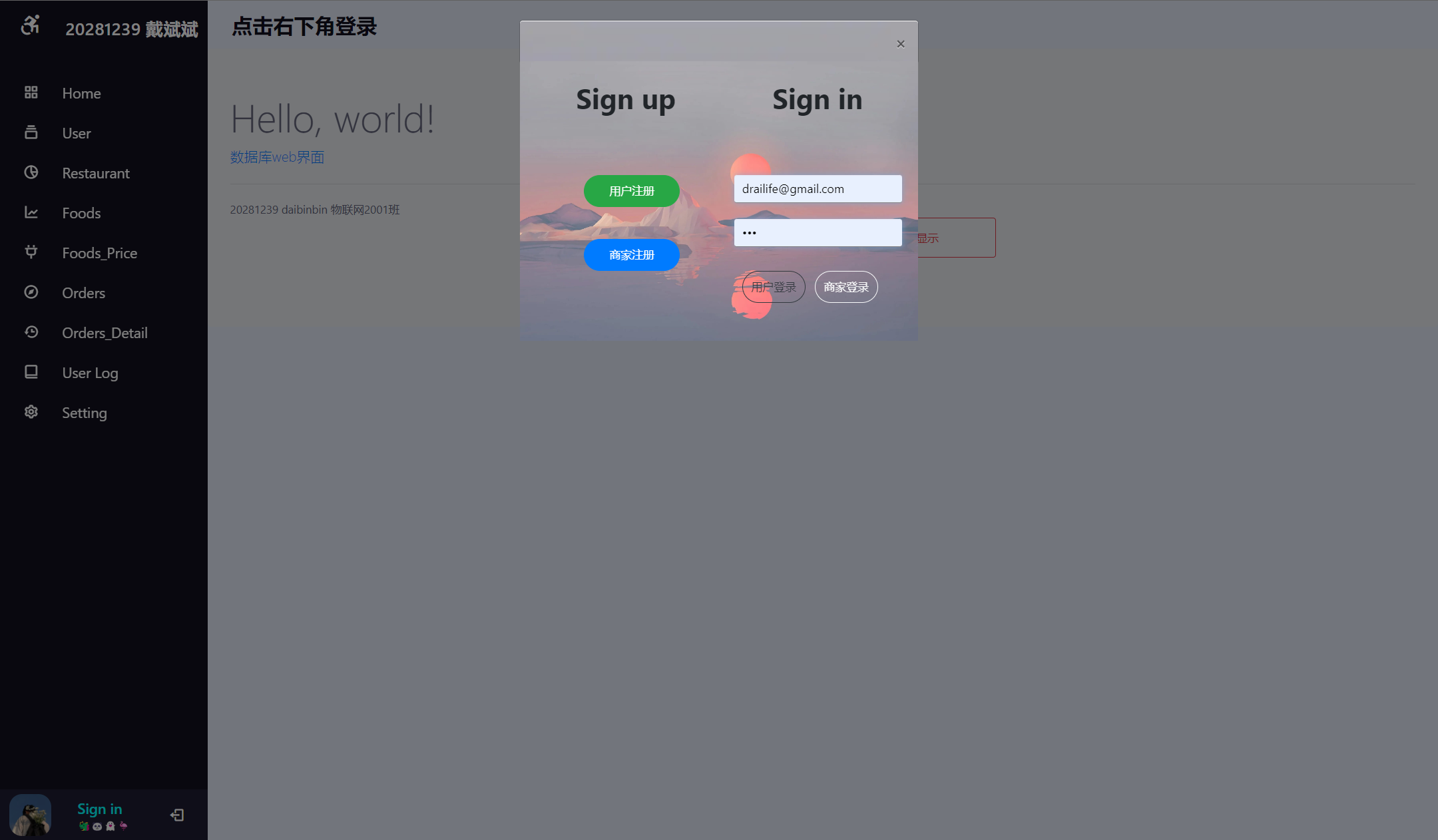
## 3.1未登录界面

展示基本信息，提示需要登录，可点击左下角登录。

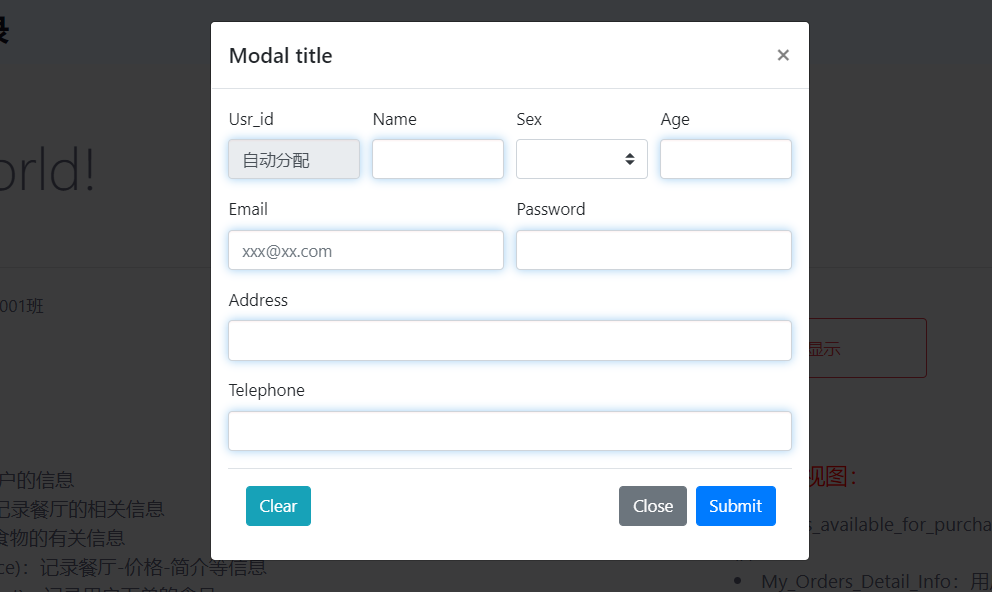


## 3.2注册、登录界面

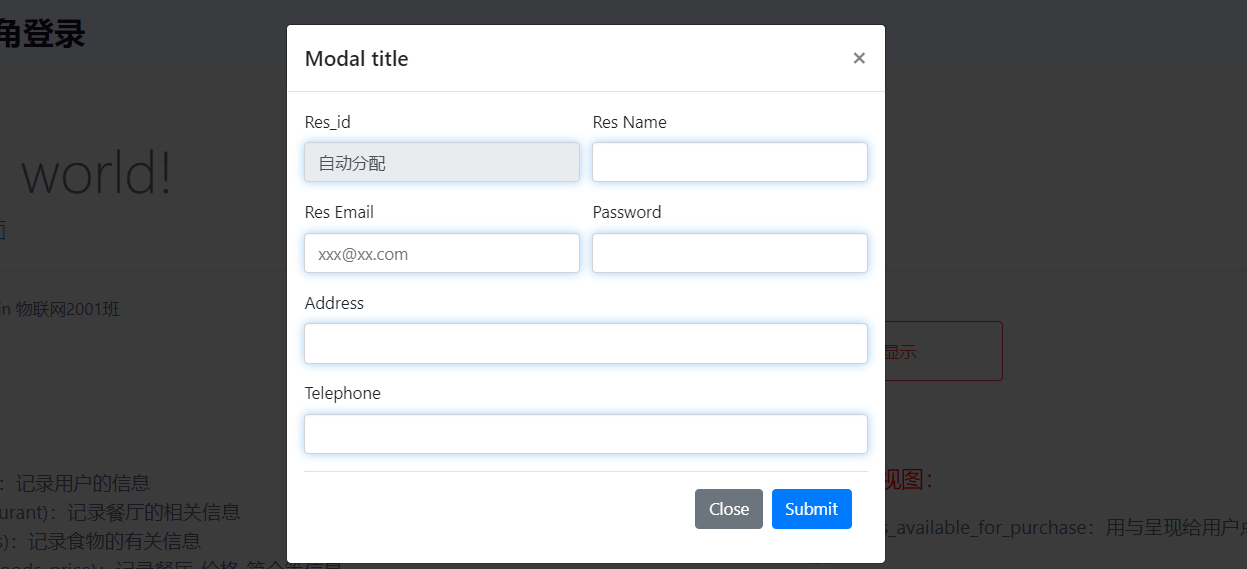
商家、顾客各有不同的注册页面、以及登录按钮，登录页面如下图



点击左边的“用户注册”弹出用户从注册页面



点击“商家注册”弹出商家注册页面



## 3.3管理员界面

管理员可点击左侧导航查看各个数据



## 3.4顾客界面

点餐界面进行点餐



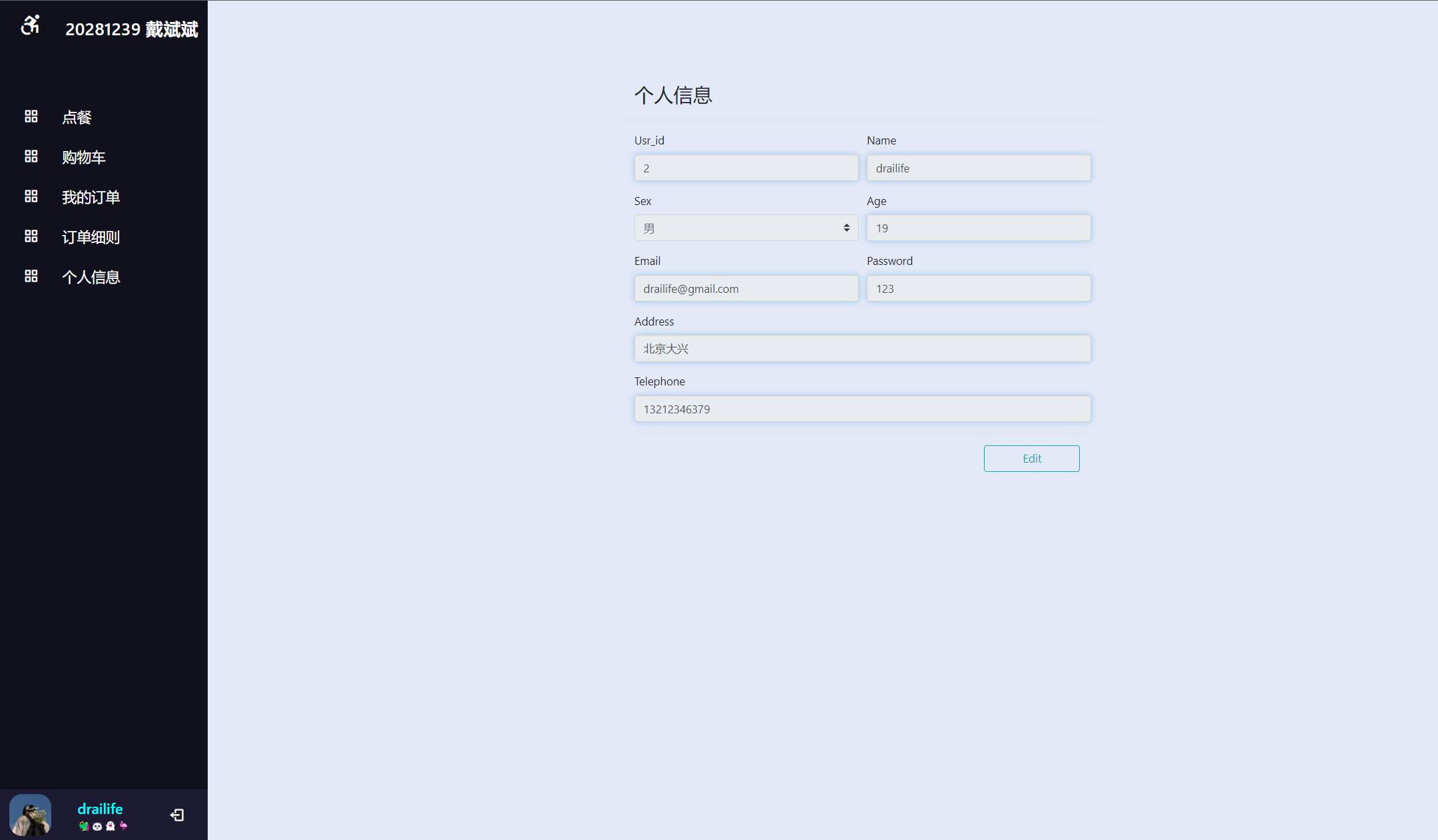
购物车界面查看所选择物品



我的订单界面查看已下单物品



个人信息界面查看或修改个人信息

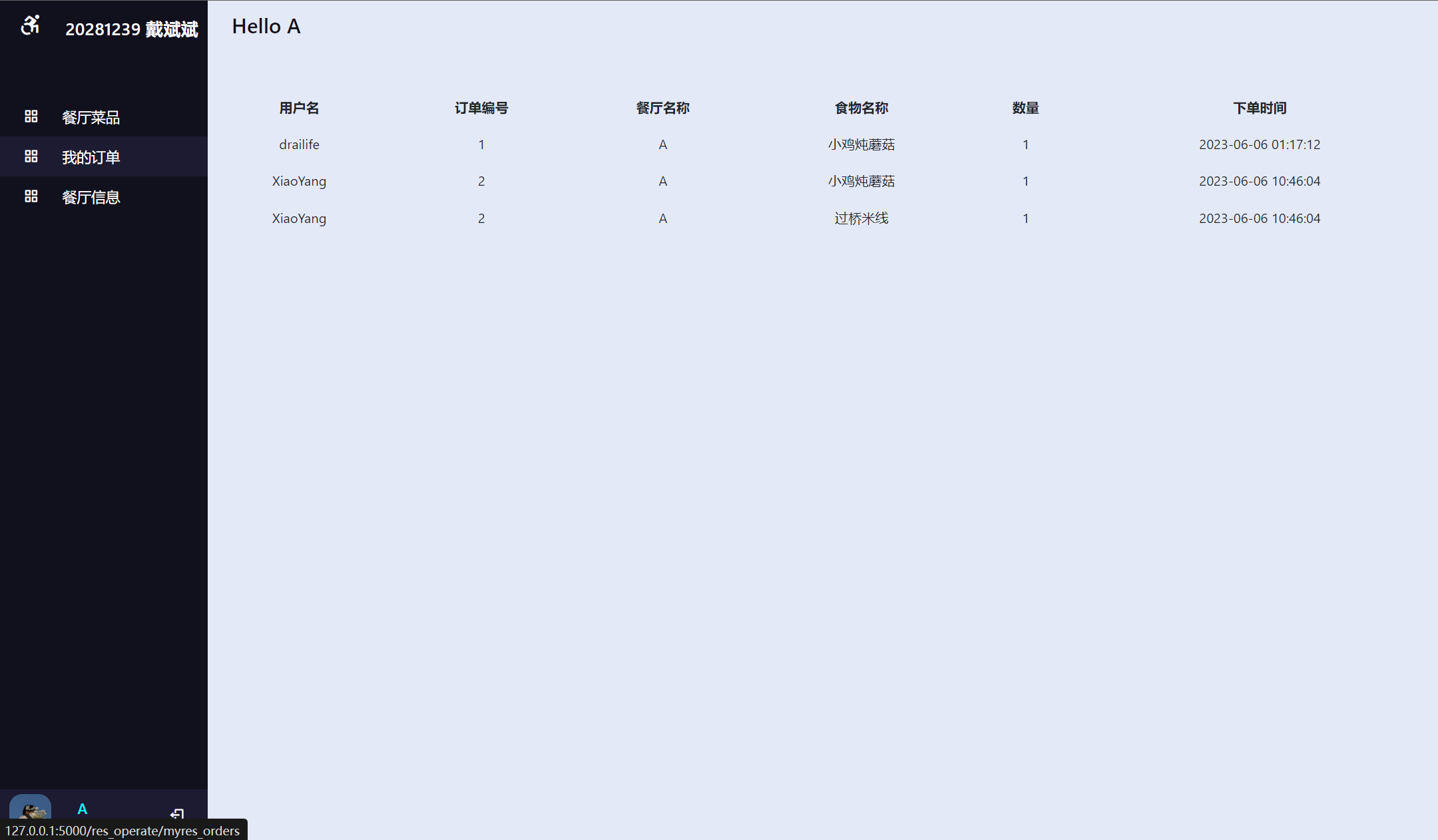


## 3.5商家界面

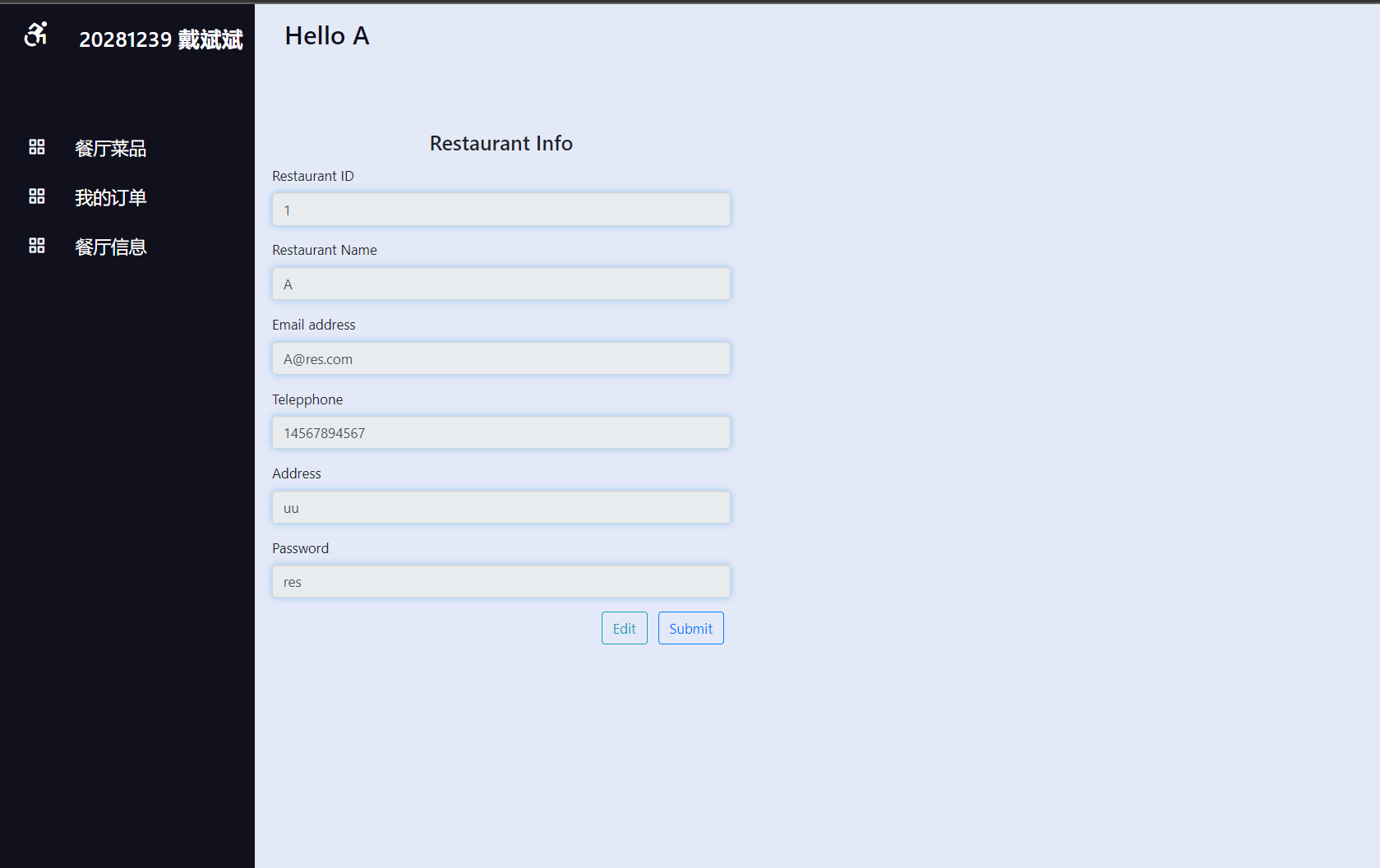
在我的餐厅界面查看餐厅内的食品数据



在我的订单界面查看顾客在该商店下单的食物详情



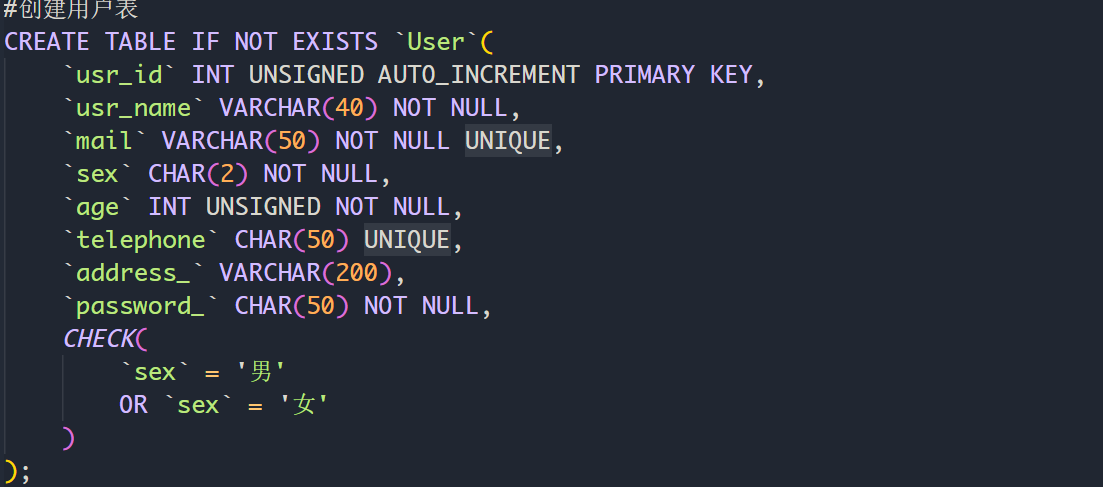
在餐厅信息界面查看商店具体信息



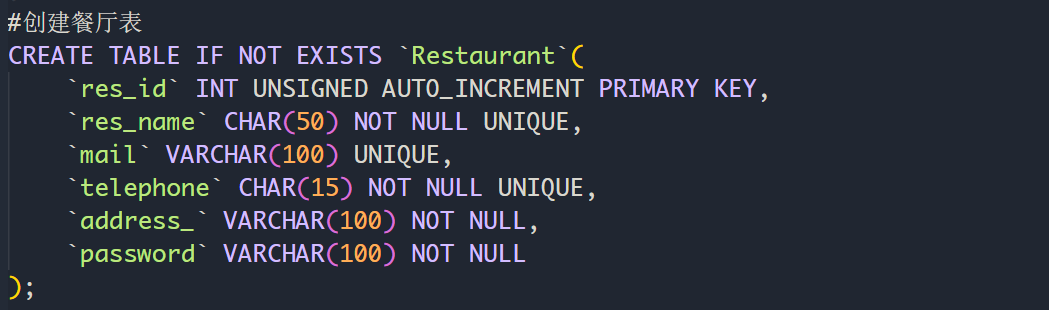
# 四、系统物理模型

## 4.1表

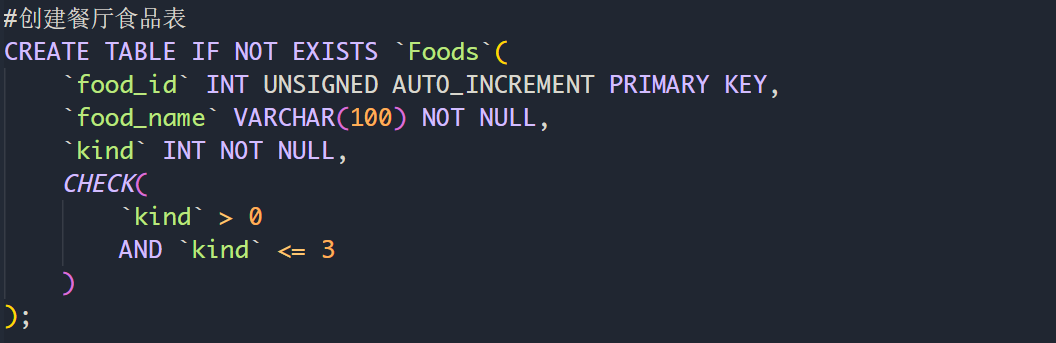
1.用户表(user)：记录用户的信息



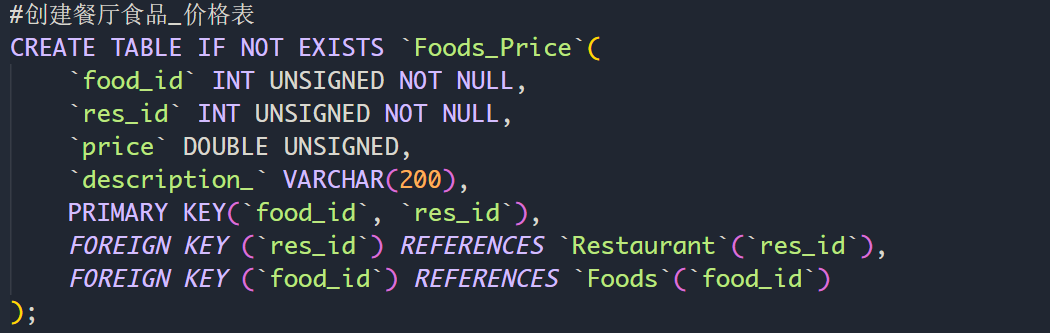
2.餐厅表(reataurant)：记录餐厅的相关信息



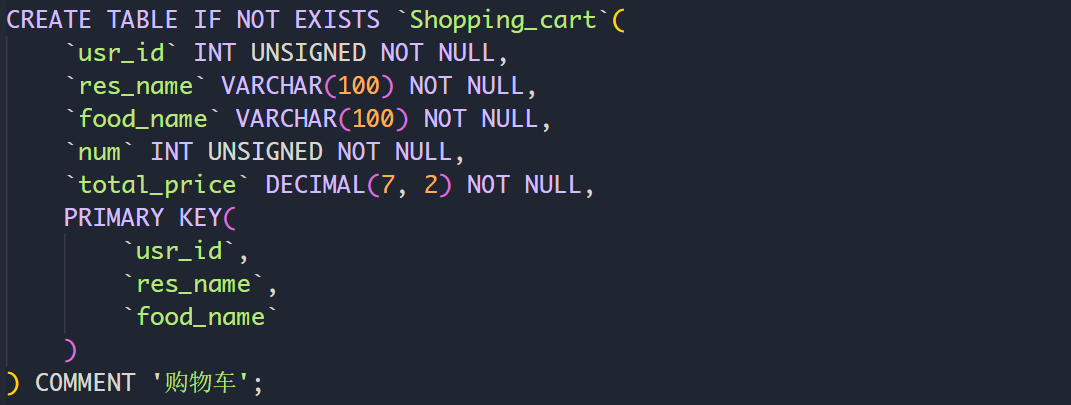
1. 食品表(foods)：记录食物的有关信息



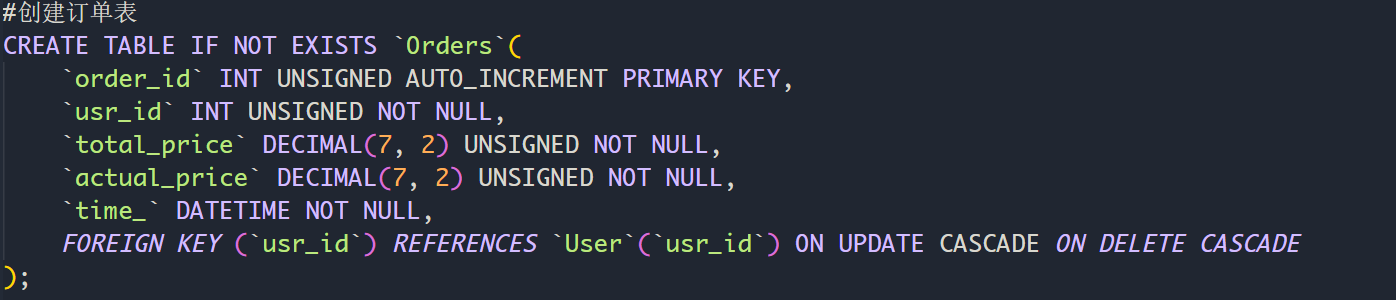
1. 食品价格表(foods\_price)：记录餐厅-价格-简介等信息



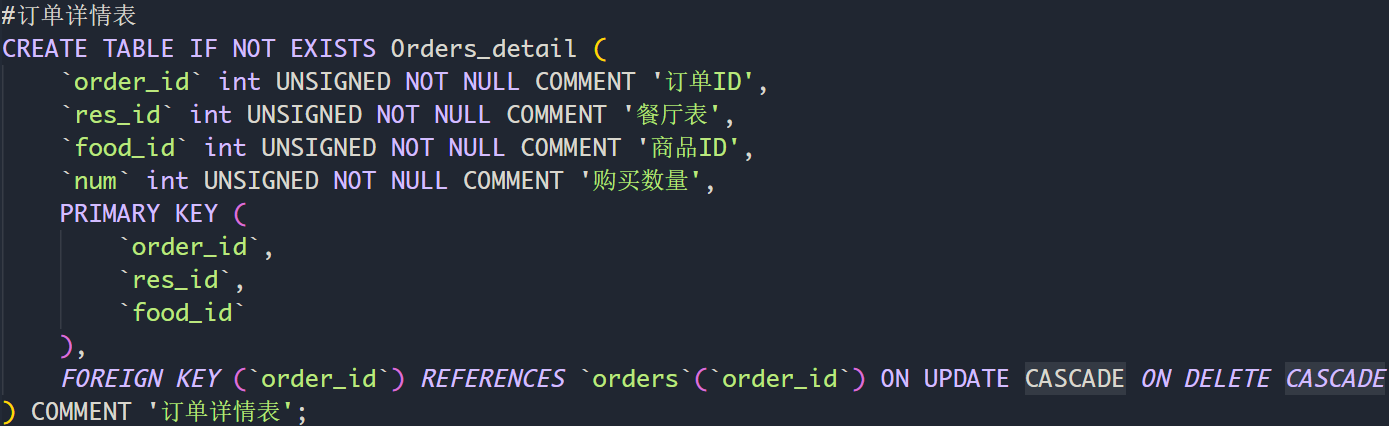
1. 购物车表(shopping\_curt)：记录用户下单的食品



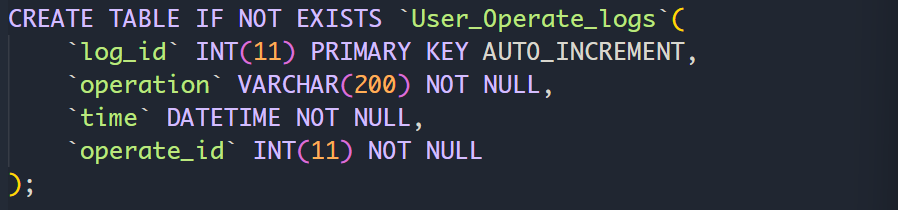
1. 订单表(orders)：记录每个订单的信息



1. 订单详情表(orders\_detail)：记录订单包含的详细信息

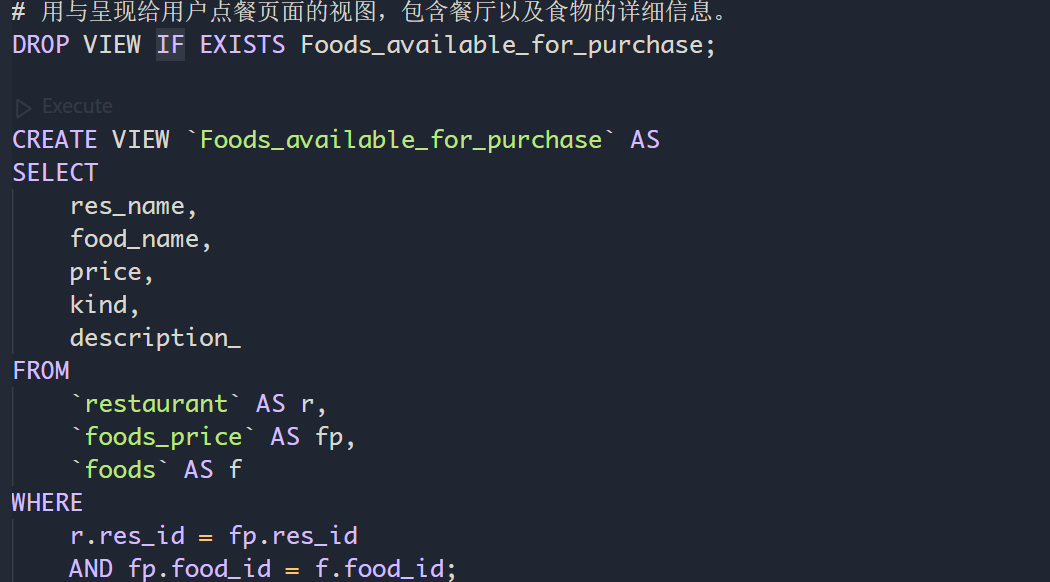


1. 操作日志表(User\_Operate\_logs):记录用户添加，删除和更新的操作

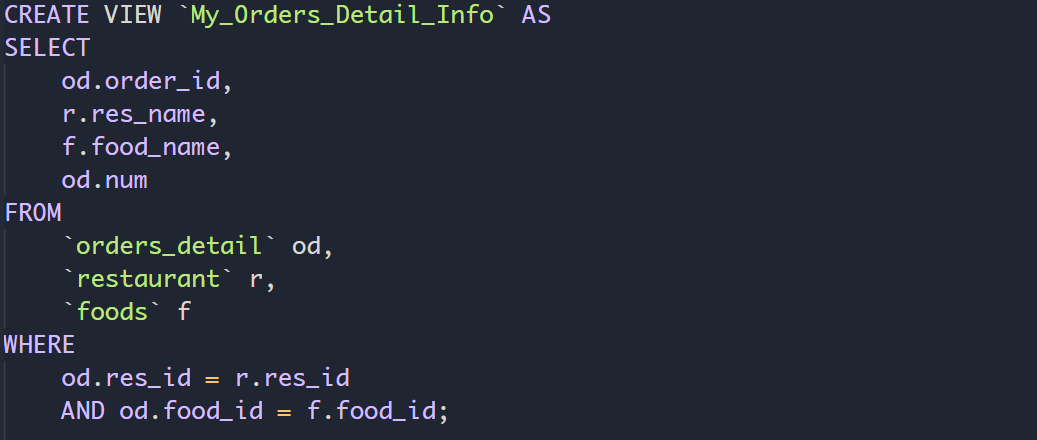


## 4.2视图

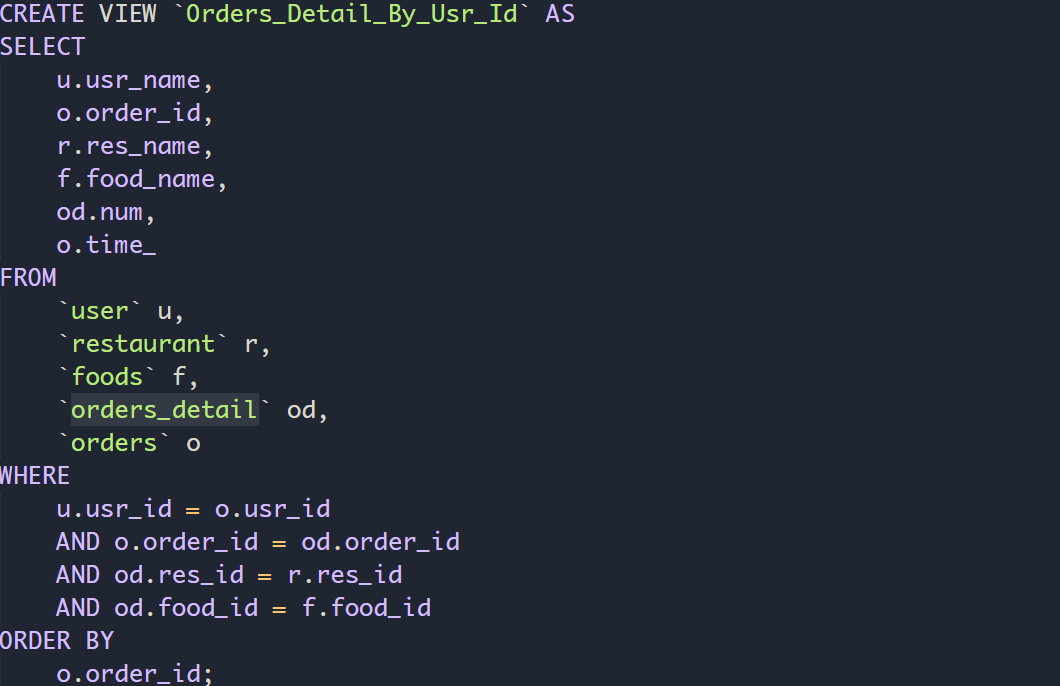
1. Foods\_available\_for\_purchase：用与呈现给用户点餐页面的视图，包含餐厅以及食物的详细信息。



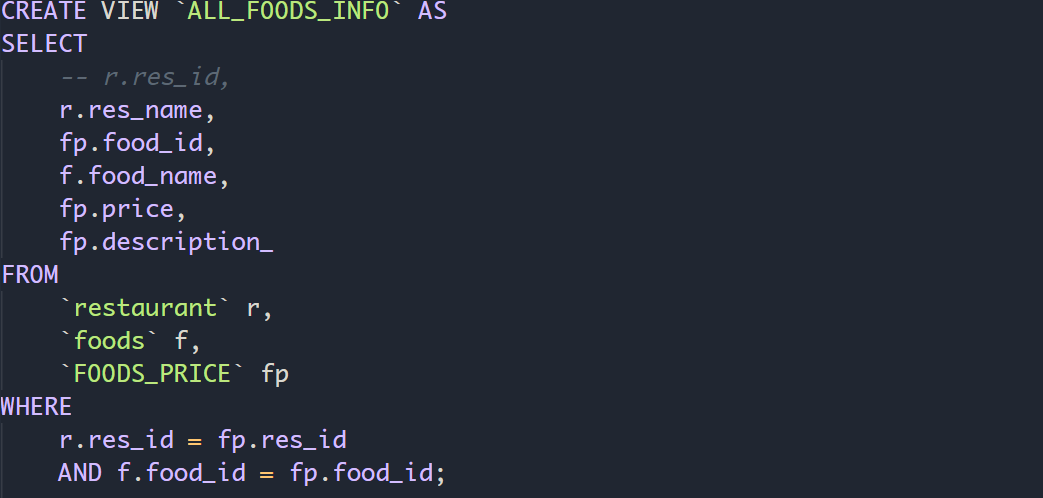
2.My\_Orders\_Detail\_Info：用户餐厅的具体信息展示



3.Orders\_Detail\_By\_Usr\_Id：用户订单细则



1. ALL\_FOODS\_INFO：餐厅的所有的餐品信息



## 4.3索引

在用户表和餐厅表的’usr\_name’和’res\_name’字段添加普通索引，因为经常需要根据名字查询相关信息。

# 五、系统安全体系设计

在登录时，后端会调用数据库检查用户的账号和密码是否匹配，防止没有权限的用户进入到系统中。



在系统中，只分配给了管理员对于所有数据的权限，对于普通的顾客用户和餐厅用户，只能使用较少的功能，防止对数据库的数据造成破坏。

# 六、系统运行环境与部署结构

目前系统运行在windows 11 环境中，后续可将项目传输到服务器并分配相应网址进行部署。

# 七、源代码列表及说明

所用的到的代码和脚本如下所示：

├────\_\_init\_\_.py //设置环境变量地址

├────app.py //项目入口文件

├────backup.exe //数据库恢复exe文件

├────backup.vbs //数据库恢复脚本文件

├────BluePrints/ //蓝图文件夹

│ ├────\_\_init\_\_.py //设置环境变量地址

│ ├────InfoShow.py //在管理员页面显示数据

│ ├────Login.py //登录功能实现

│ ├────Res\_Operate.py //餐厅用户功能实现

│ ├────Update.py //管理员更新数据操作实现

│ └────User\_Operate.py //顾客用户操作实现

├────config.py //flask的配置文件

├────exts.py //插件文件的设置

├────README.md //文件介绍

├────SqlFIle/ //SQL文件夹

│ ├────backup.sql //数据库恢复的sql文件

│ ├────Create.sql //数据库表创建的文件

│ ├────Delete.sql //用于删除表数据

│ ├────Query.sql //用于查询表数据

│ └────视图-存储过程-触发器.sql //存储视图存储过程触发器

├────SQLModel.py //ORM映射模型的表

├────static/ //静态文件

│ ├────img/ //图片文件

│ └────js/ //js文件

│ │ ├────bootstrap.min.js //bootstrap框架

│ │ ├────bootstrap.min.js.map

│ │ └────jquery-3.6.4.min.js //jquery库

└────templates/ //html的文件

│ ├────0\_base.html //网站基本框架

│ ├────10\_orders\_detail.html //订单详情表数据

│ ├────11\_my\_restaurant.html //餐厅用户展示

│ ├────12\_User\_Purchase\_From\_Res.html //购物车展示

│ ├────1\_index.html //开始界面

│ ├────2\_setting.html //设置界面

│ ├────4\_user.html //用户界面

│ ├────5\_user\_log.html //用户日志表数据

│ ├────6\_Restaurant.html //餐厅表数据

│ ├────7\_foods.html //食品表数据

│ ├────8\_foods\_price.html //食品价格表数据

│ ├────9\_orders.html //订单表数据