**软件设计与体系架构实验报告**

——食堂饭菜评价系统

## 一、主要功能

1. 项目简介

本系统为学生更好了解食堂饭菜提供了一个信息交换共享的平台，用户可以在平台上查看武大各个食堂的丰富菜品及评价，根据自己的用餐体验，对系统中已有菜品进行评论。用户还可以对系统中还未更新的菜品进行添加并评论。

2. 具体功能：

（1）实现了用户、食堂、菜品和评价四个界面。

（2）实现了对四个界面信息的数据库读取显示及增删改查功能。

## 二、实现方法

1. 前端使用Vue框架。

2. 使用Springboot，SpringMVC和mybatis。

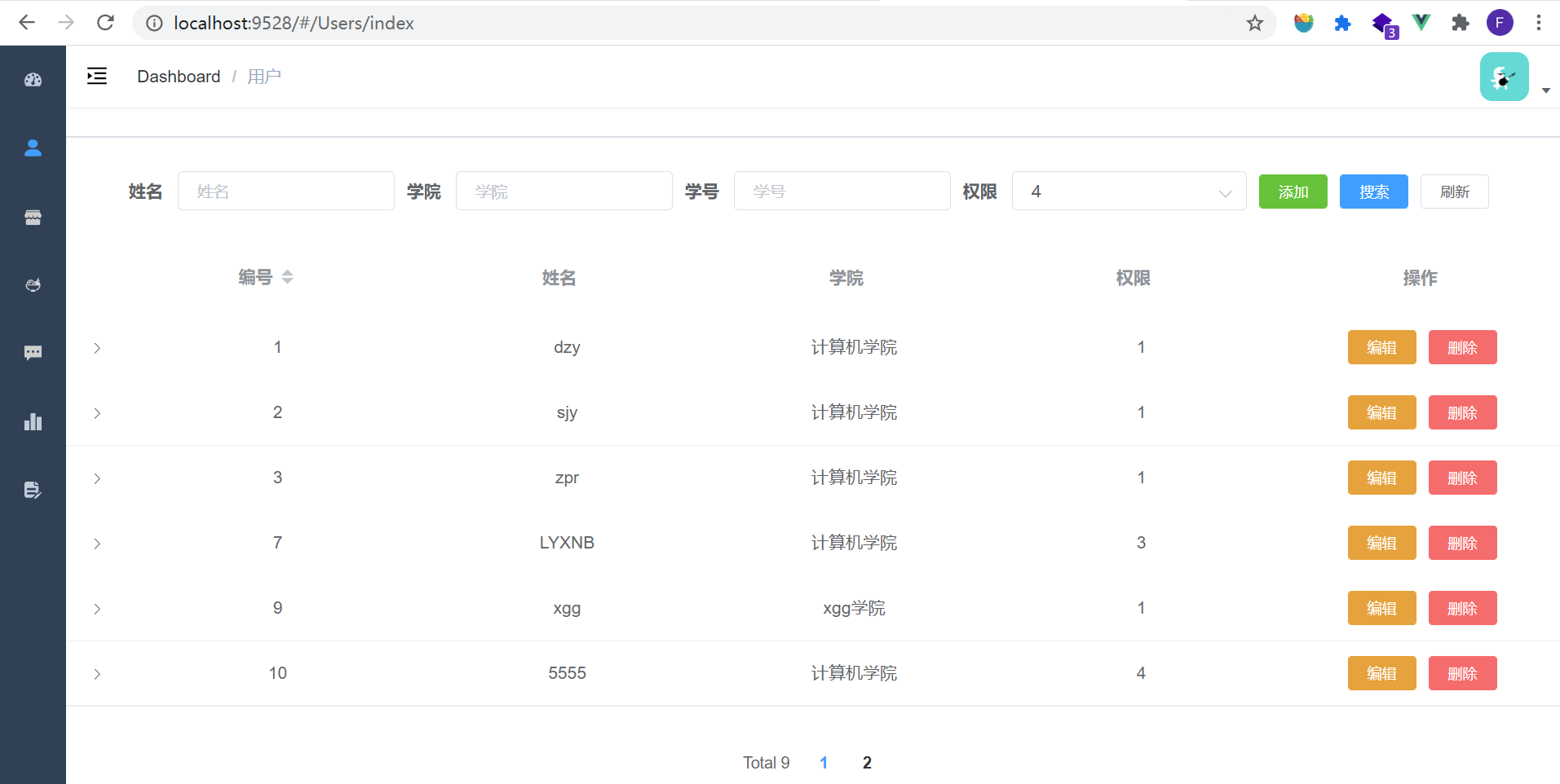
3. 使用基于SwaggerUI可视化 API。

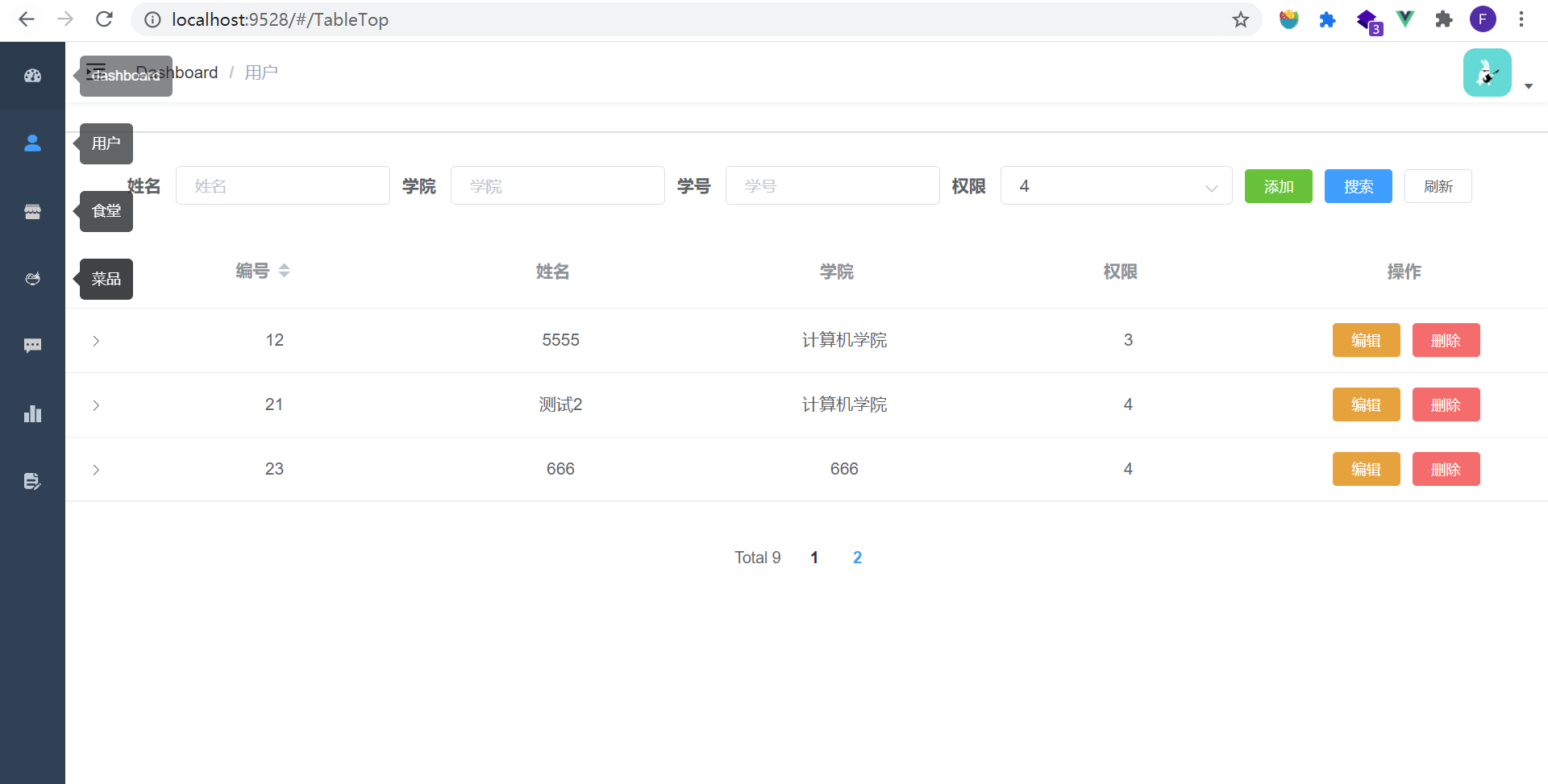
## 三、效果展示

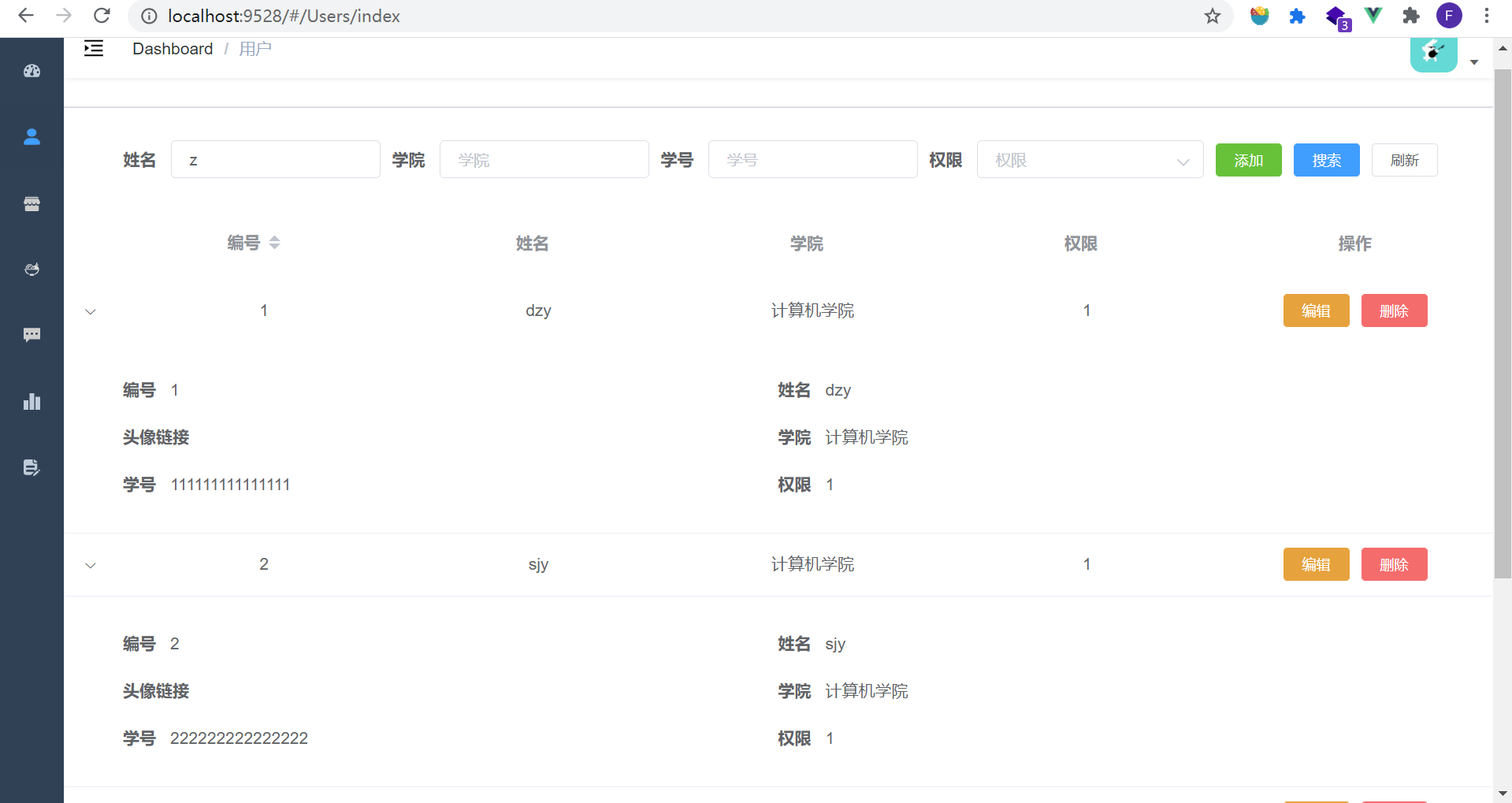
**前端**

### 1. 用户界面

（1）查看：此界面可以查看系统所有用户基本资料及其权限，点击用户可以展开用户详细信息。



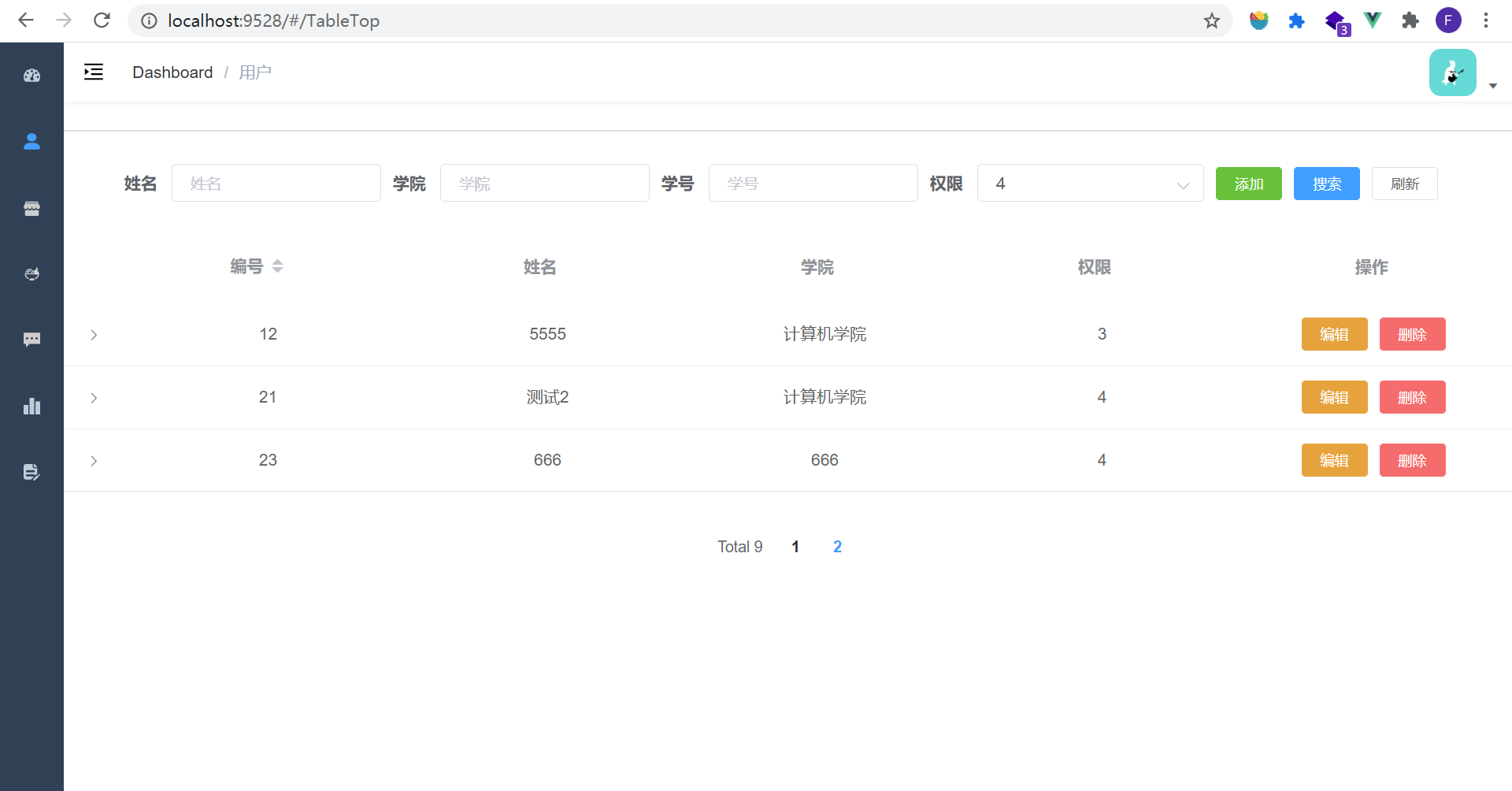




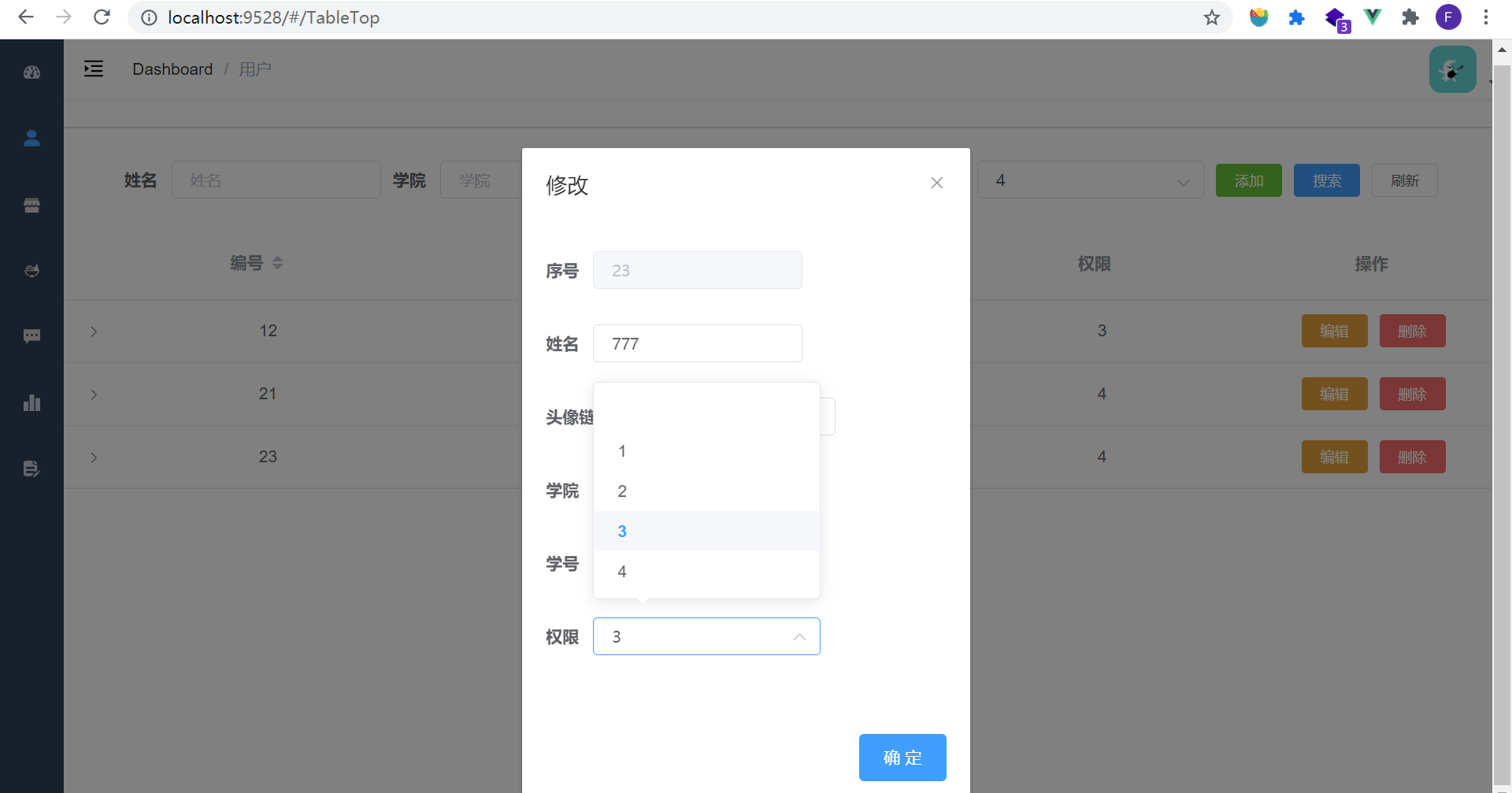
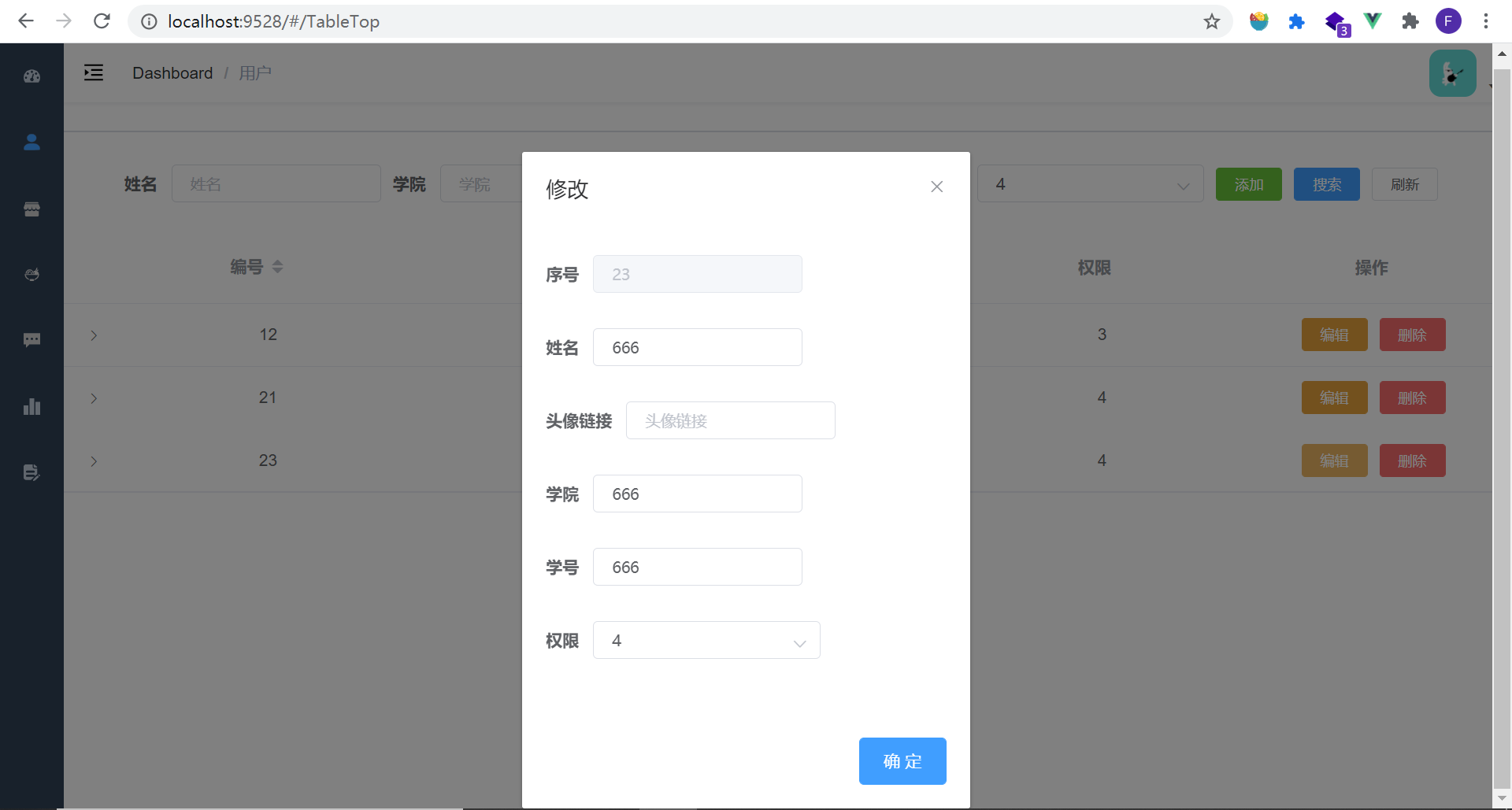
（2）添加：在文本框中输入姓名、学号、学院和权限，并点击添加，可以实现对新用户的添加，编号顺延。如下图所示，增添了XXXX用户。



（3）删除：点击用户后面操作中的删除按键，可以将该用户删除。如下图删除了之前添加的XXXX用户。

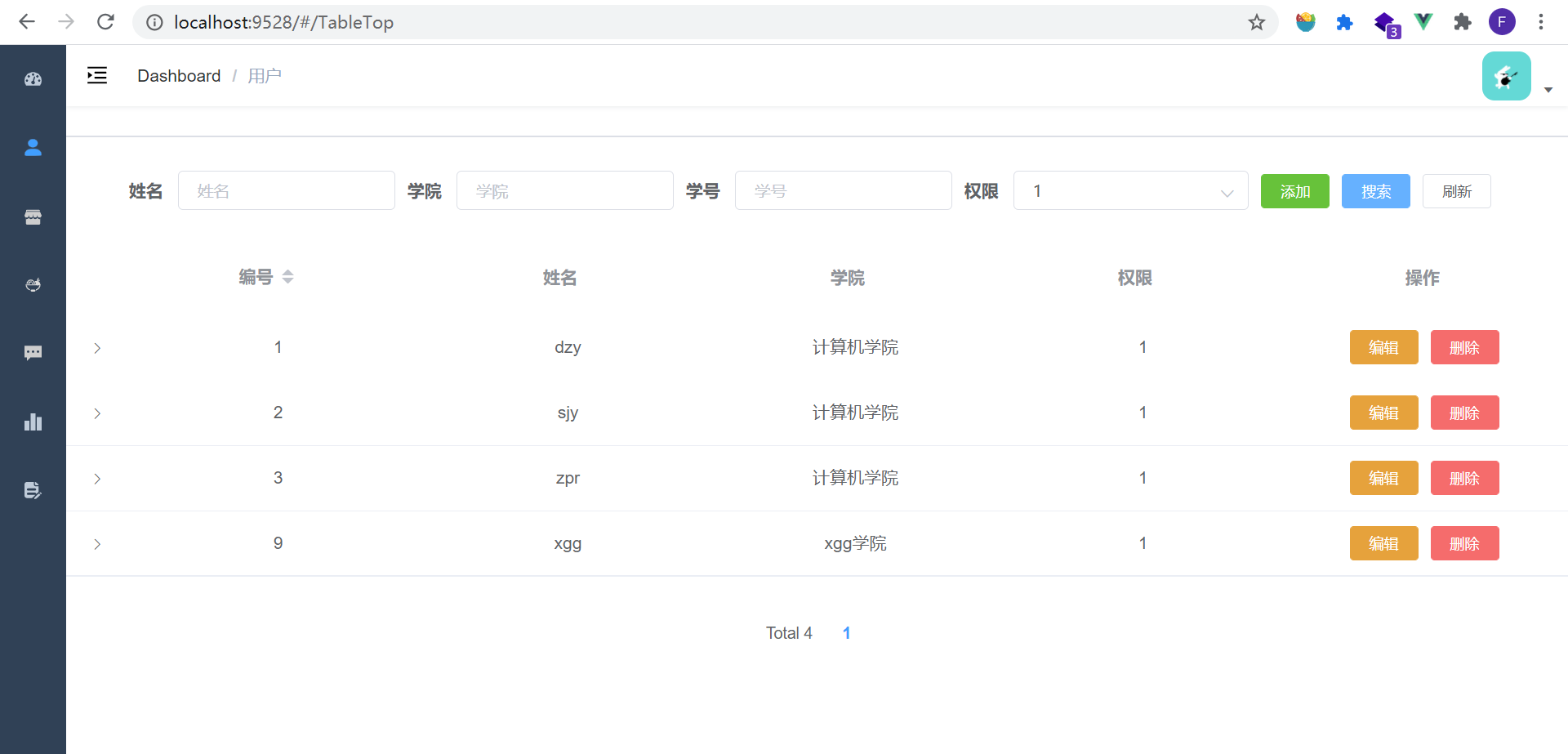


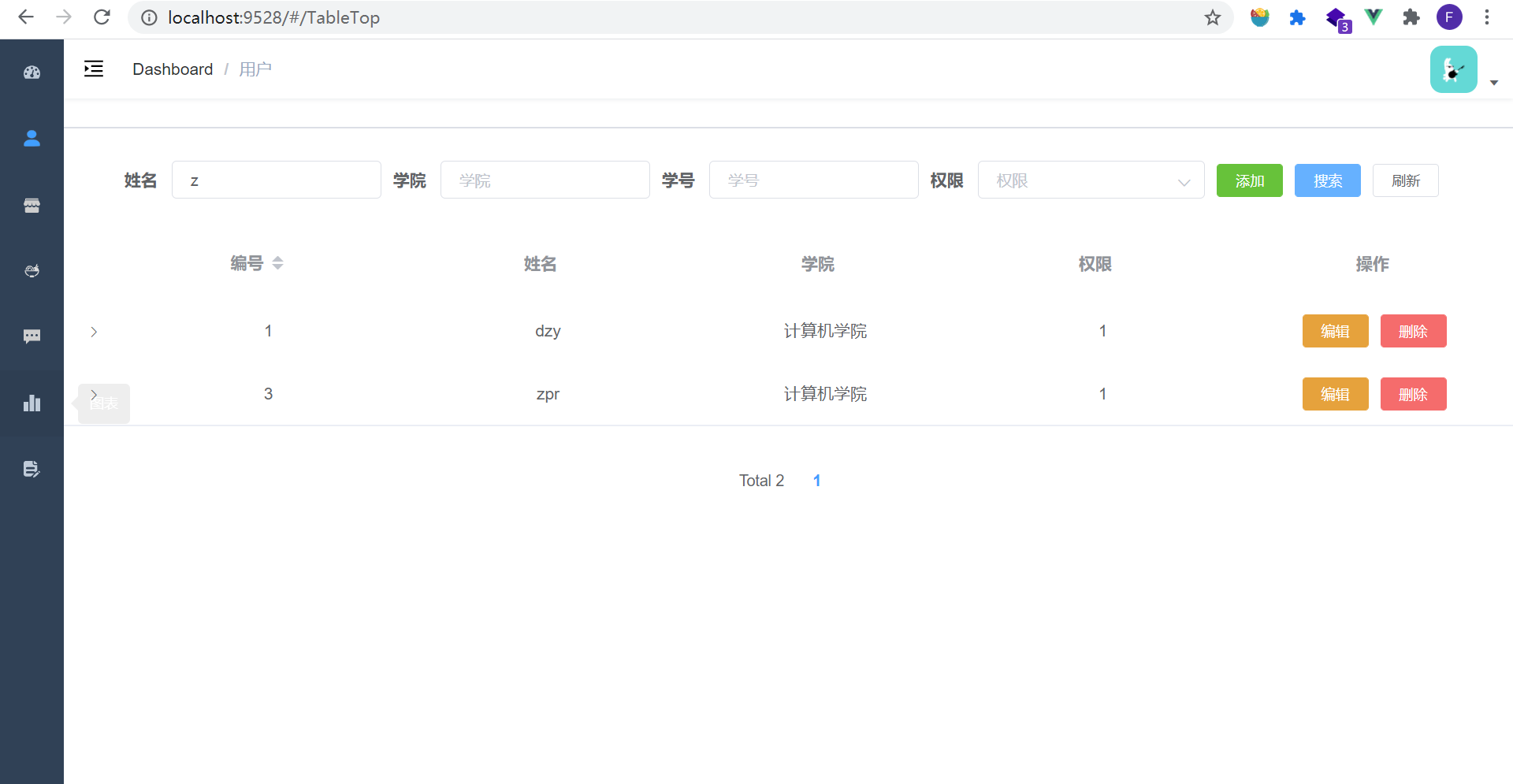
（4）编辑：点击操作中的编辑按键，可以对该用户除序号外的其他属性进行修改。如下图，将姓名为666的用户更改了姓名、学院、权限等信息。





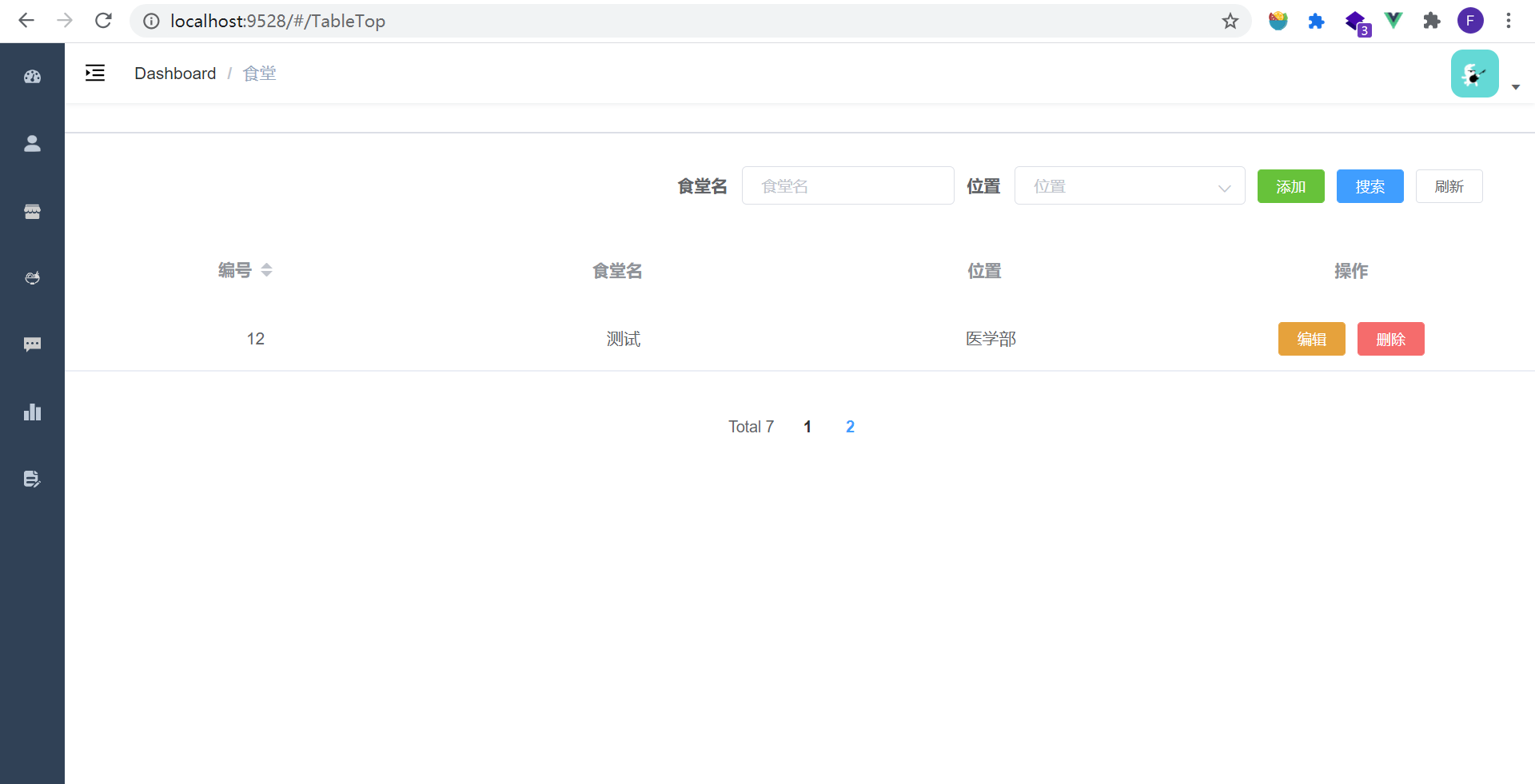
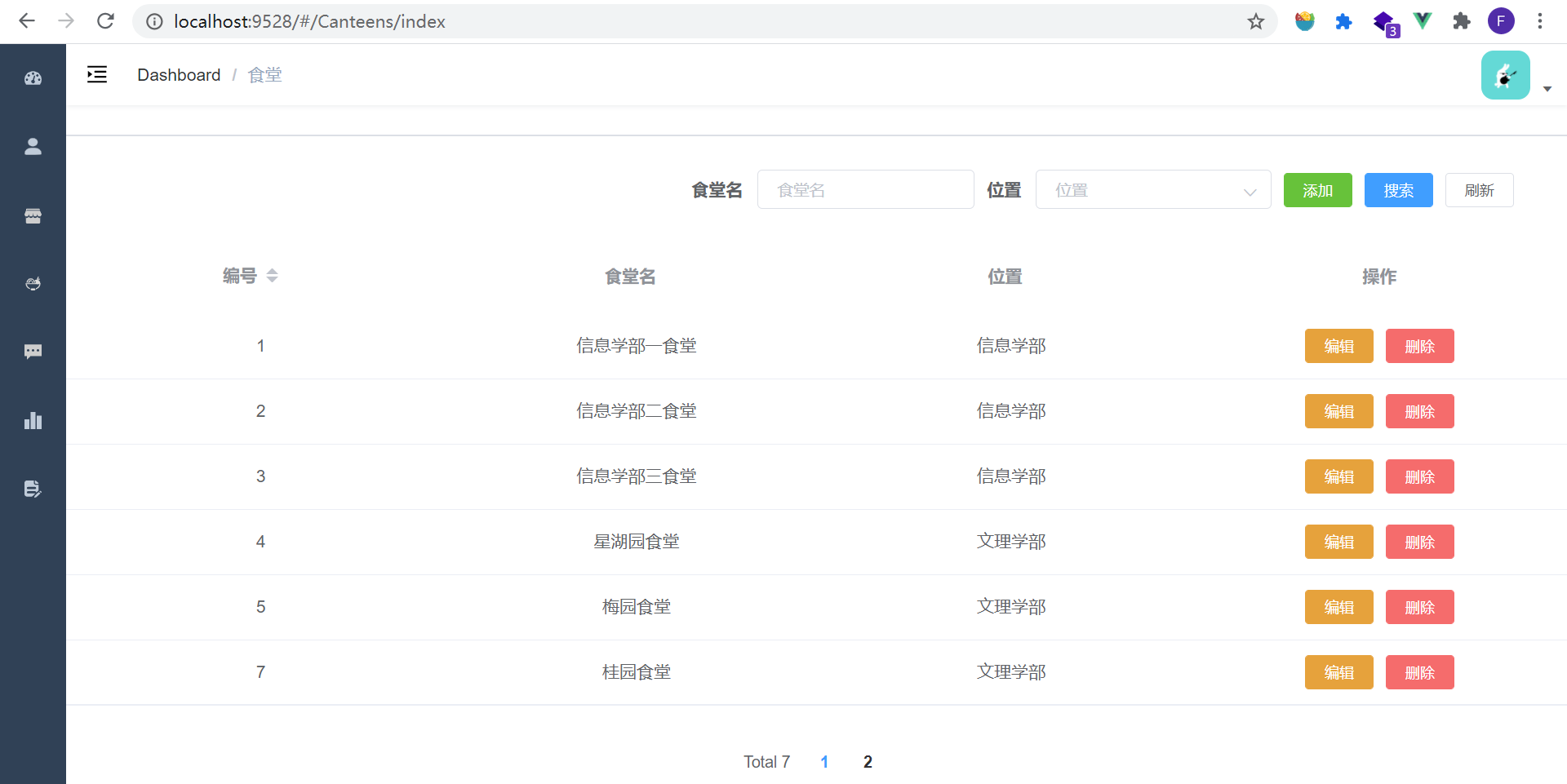
（5）搜索：在文本框中，可以针对任何一个属性进行搜索，如下图，分别筛选出权限为1的所有用户和姓名中含字母z的所有用户。

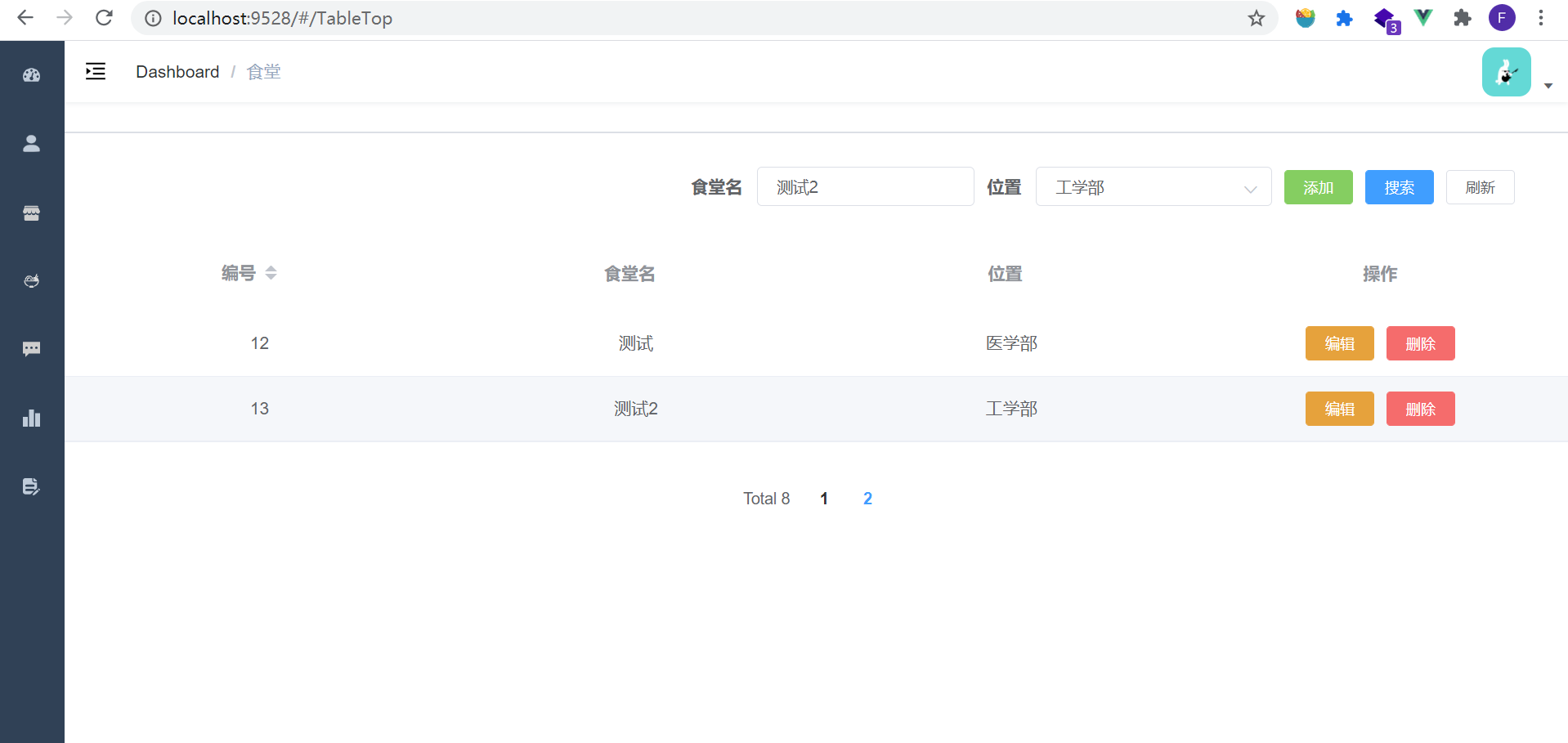


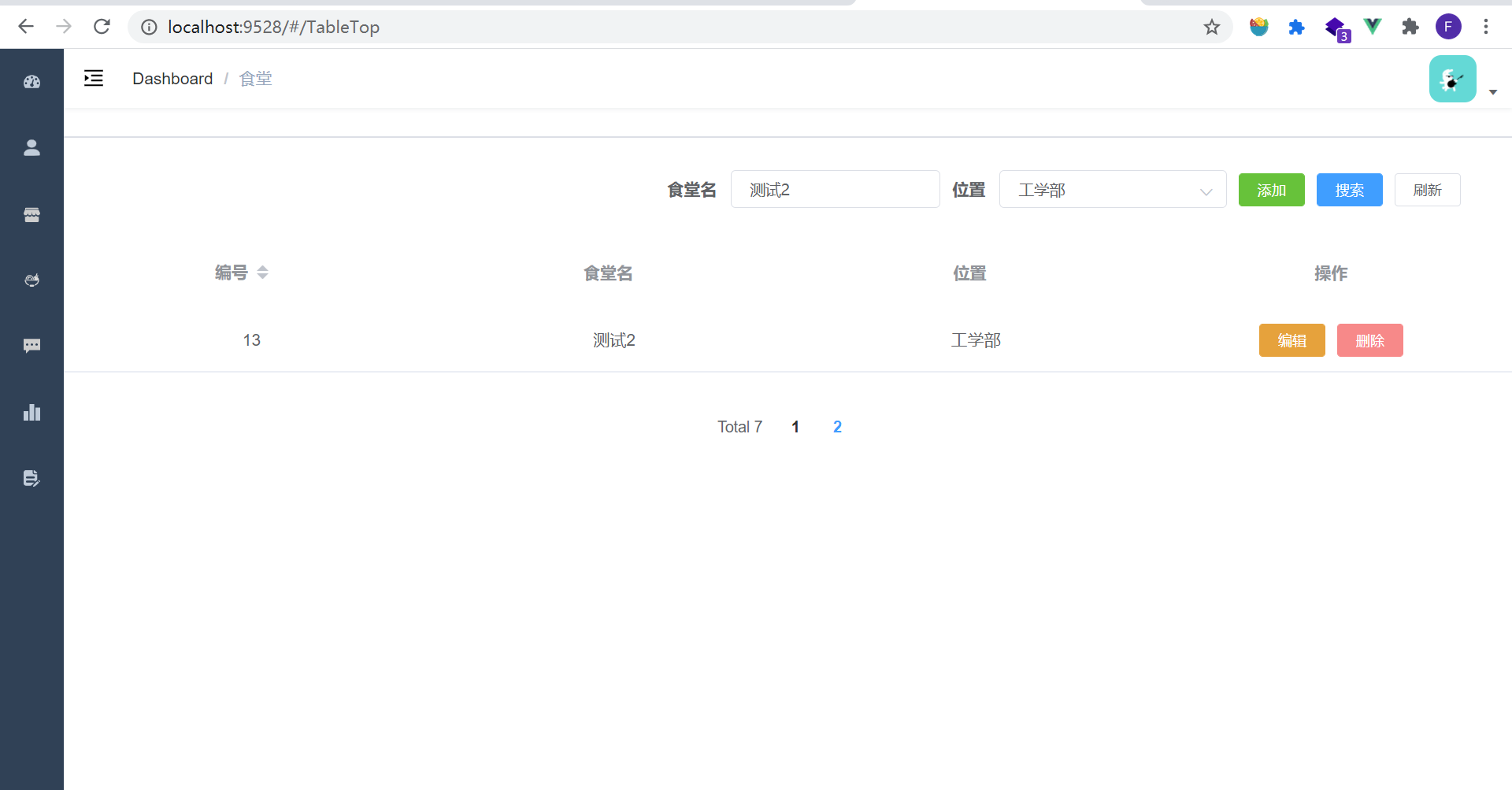


### 2. 食堂界面

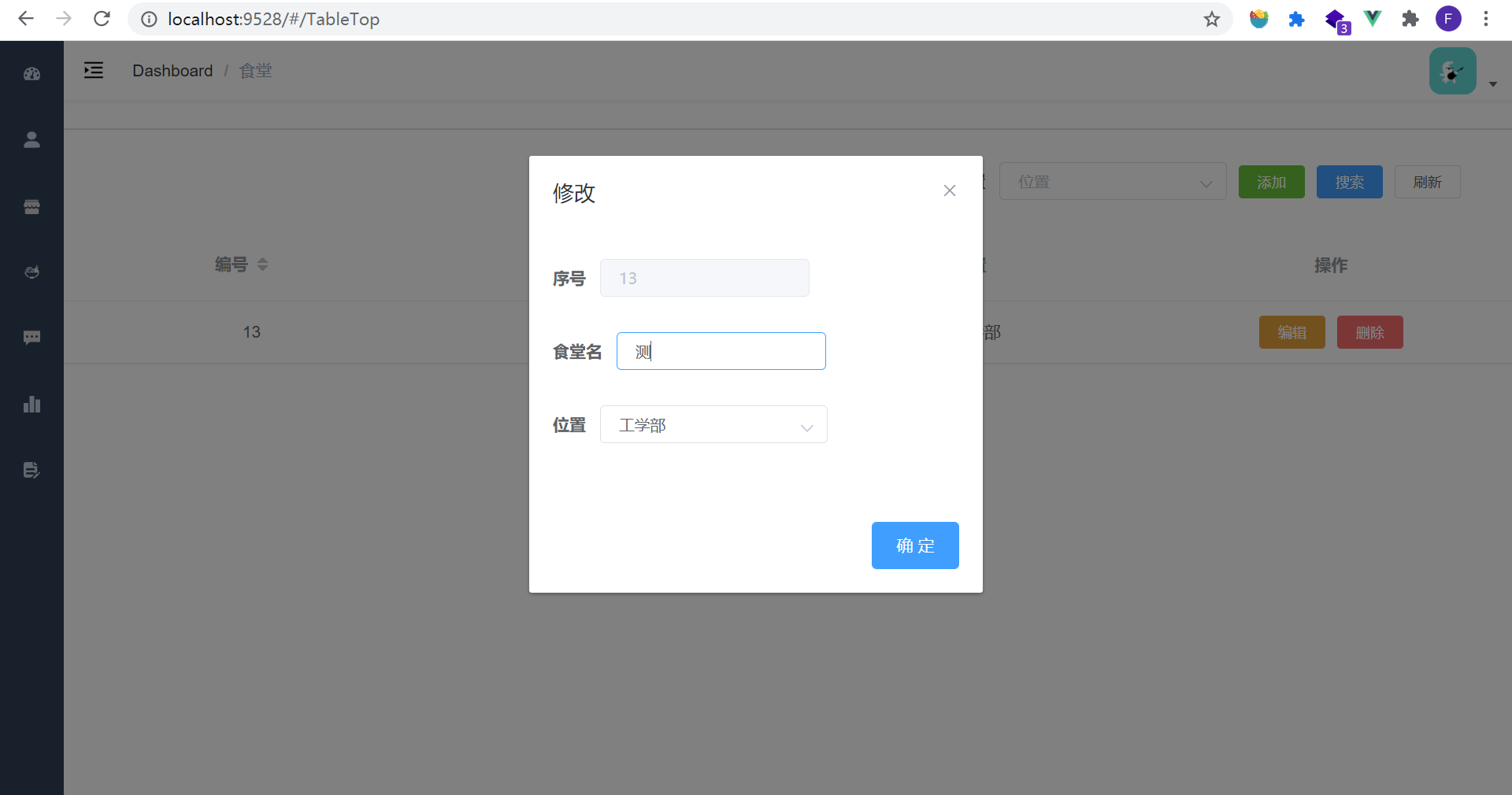
（1）查看：此界面可以查看所有食堂及其所在位置。

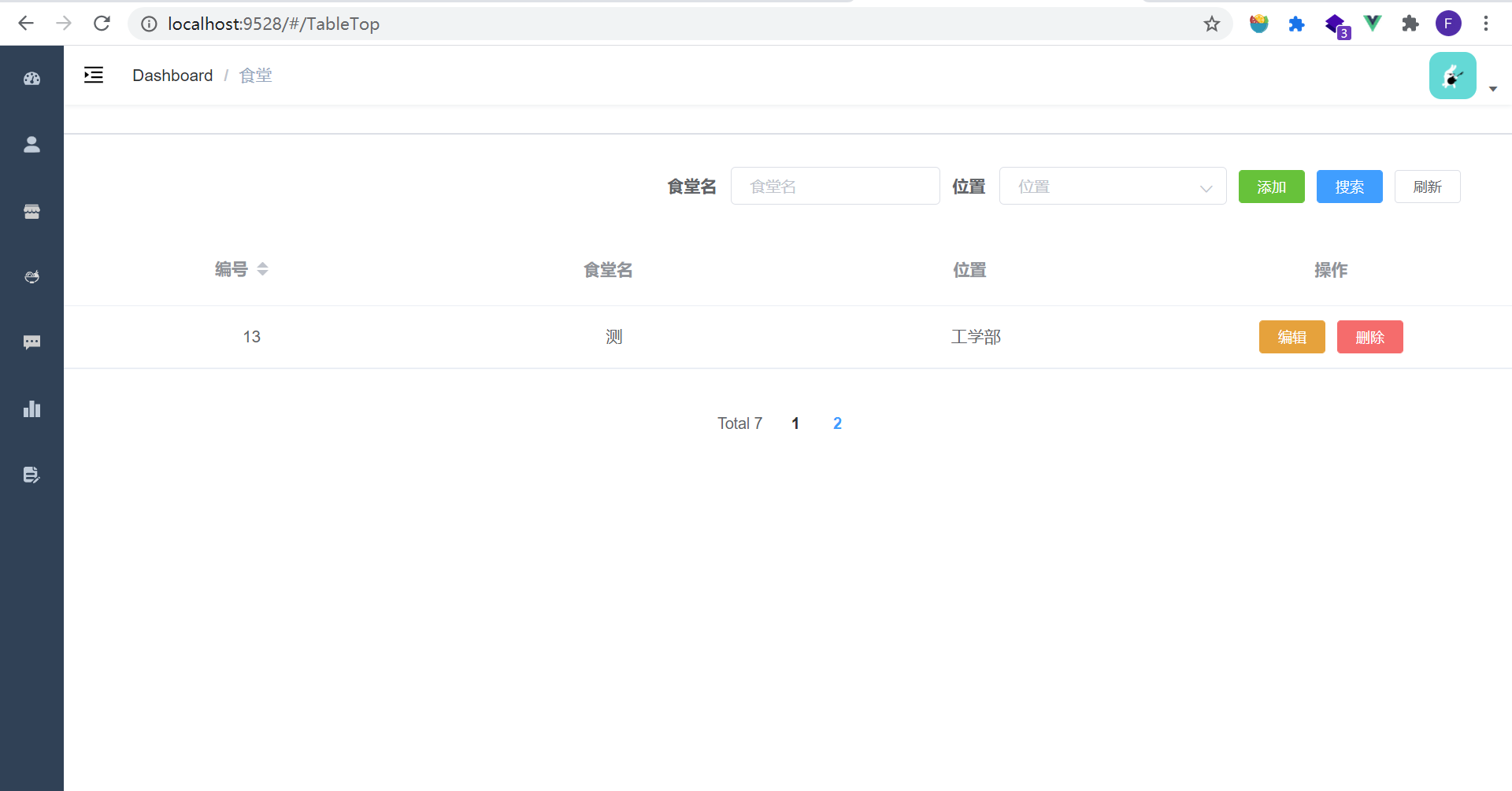
（2）添加：可以通过文本框输入食堂名和位置，并点击“添加”对食堂进行添加。如下图，添加了位于工学部，名为“测试2”的食堂。



（3）删除：点击食堂后的“删除”按键，即可将对应食堂删除。如下图删除了编号为12的食堂。

（4）编辑：点击食堂后面的“编辑”按键即可对食堂的基本信息进行编辑。如下图，将13号食堂的食堂名由“测试2”改为了“测”。





（5）搜索：可以按照食堂名或食堂位置进行搜索，如下图，筛选出了位于信息学部的所有食堂。

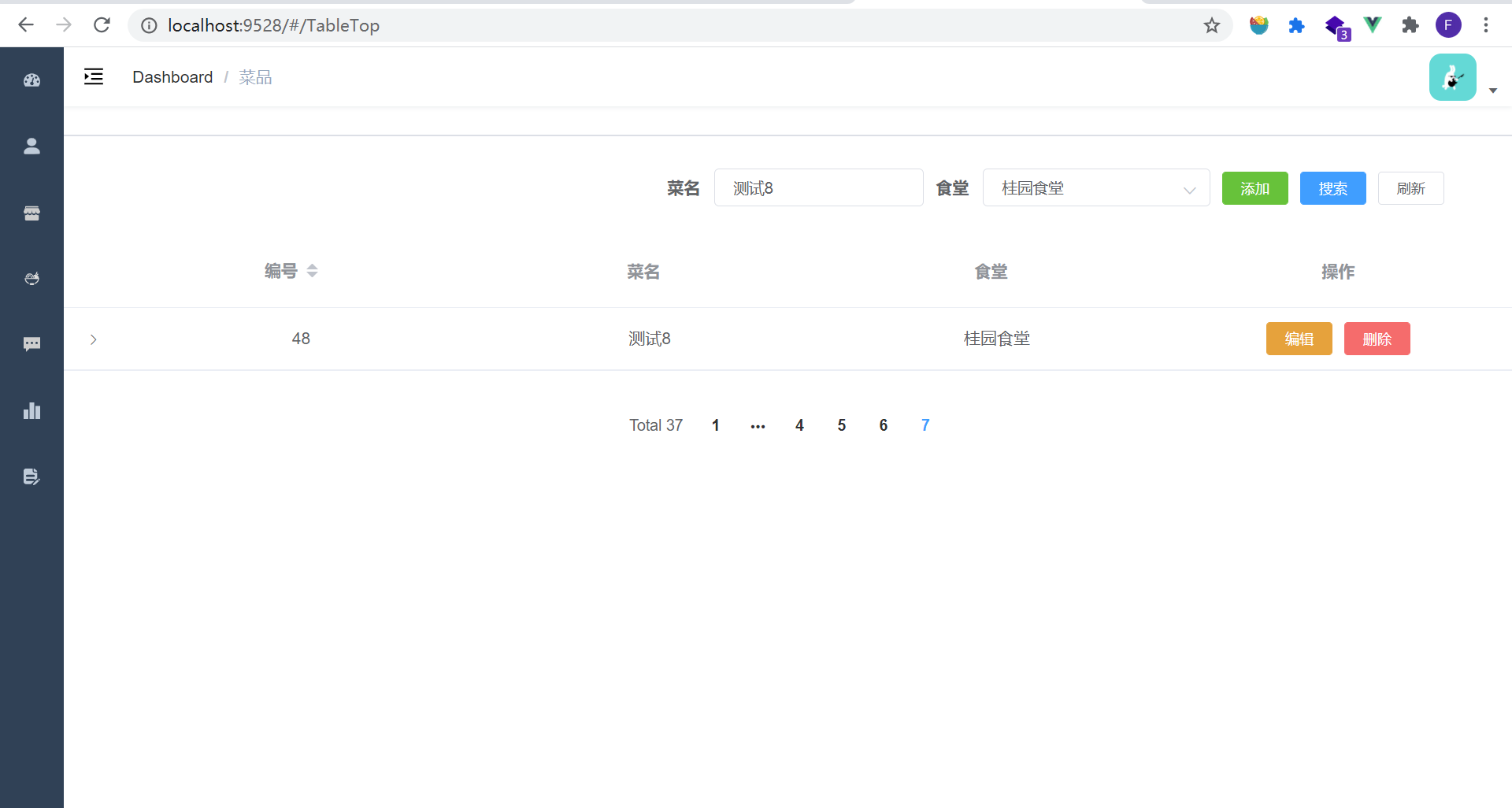


### 3. 菜品界面

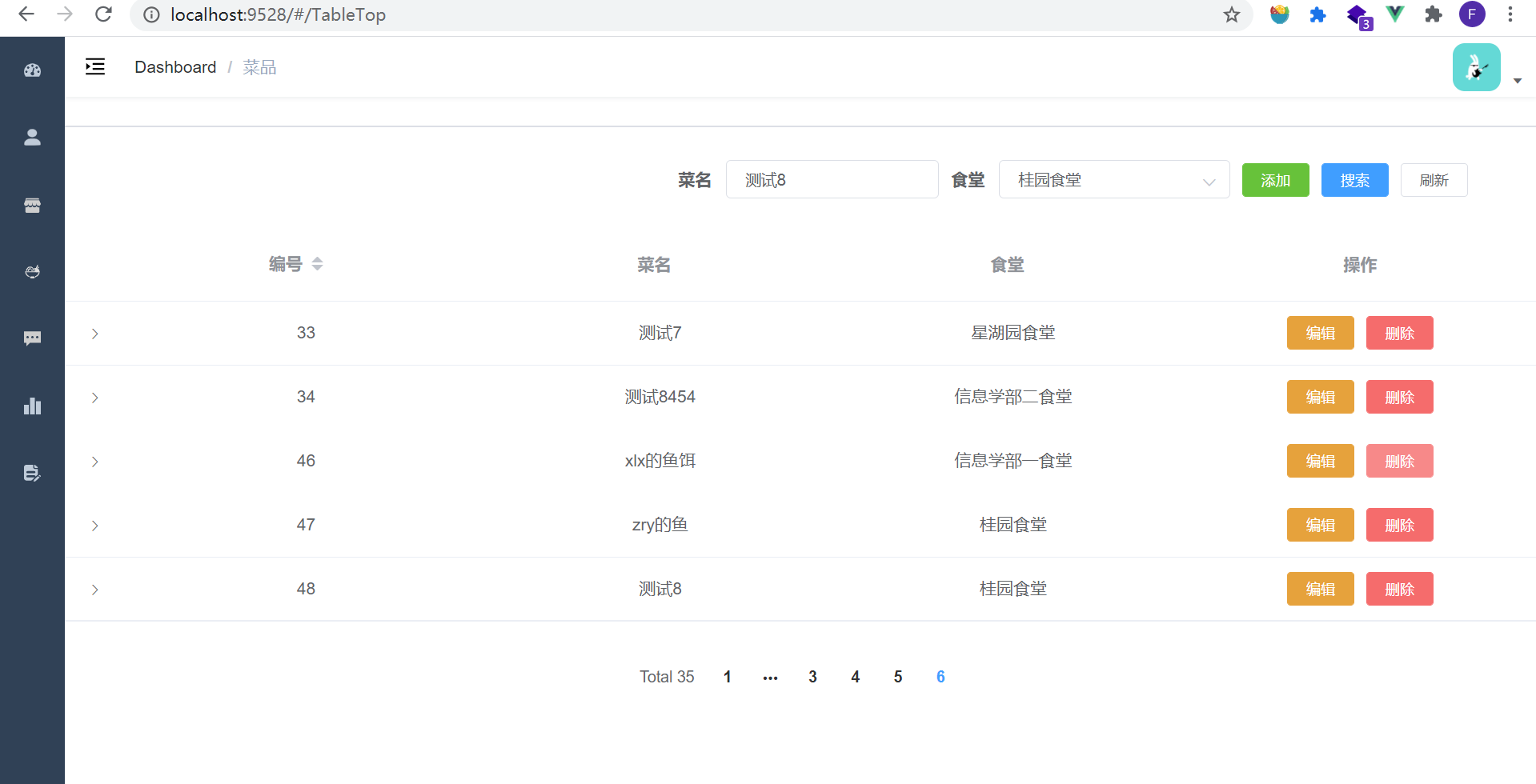
（1）查看：此界面可以查看所有菜品的基本信息。

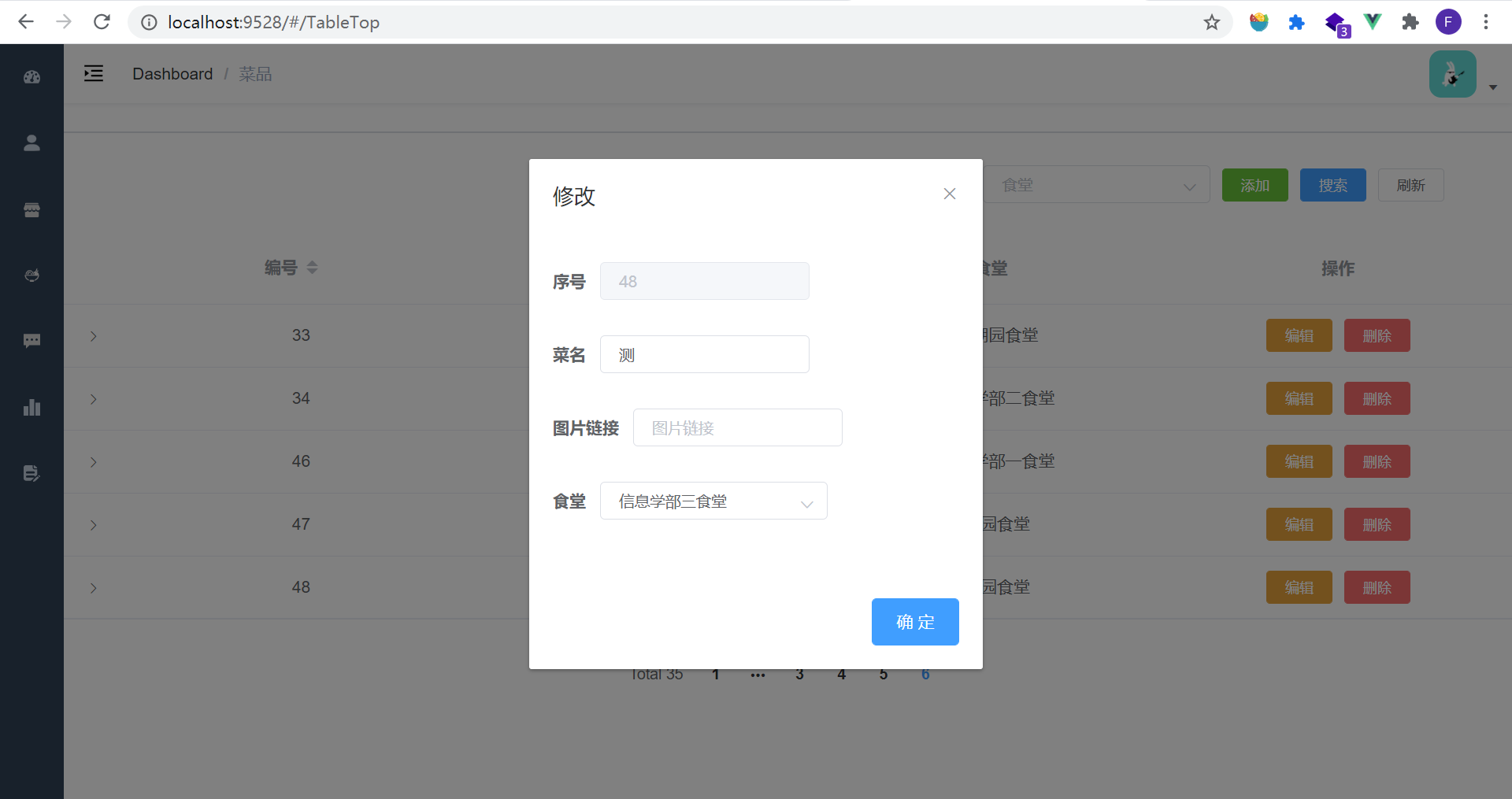


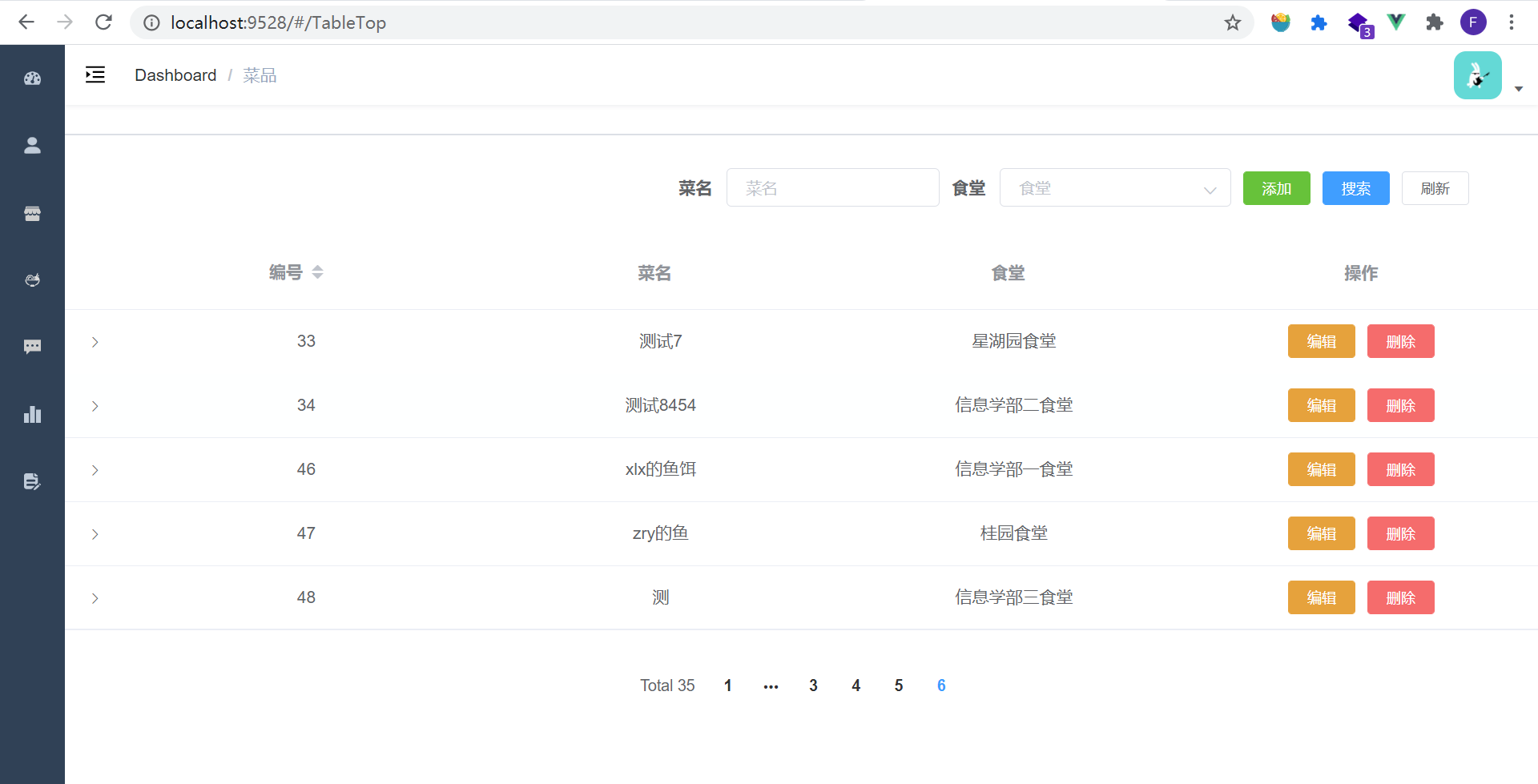
（2）添加：在文本框中输入菜名和食堂并点击“添加”即可对菜品进行添加，序号顺延。如下图，添加了桂园食堂的“测试8”这一菜品。



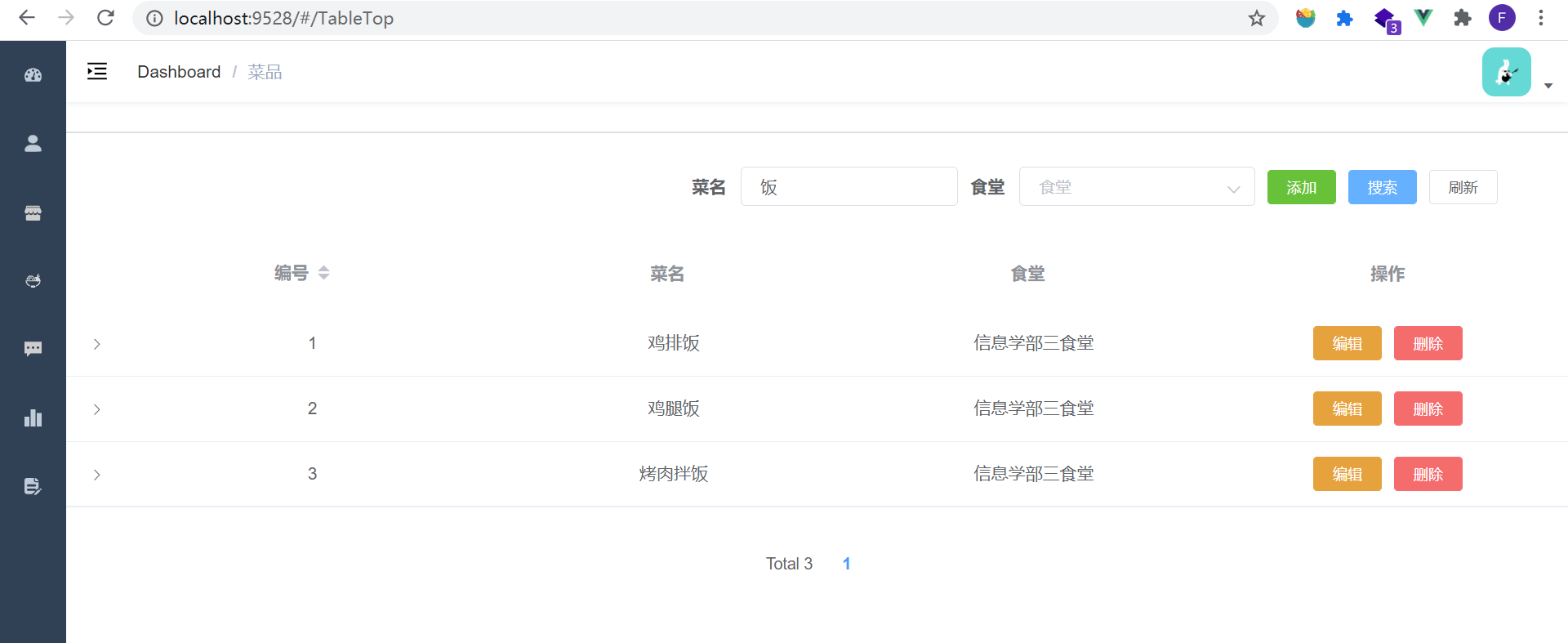
（3）删除：点击菜品后面的删除键，可将菜品删除。



（4）编辑：点击菜品后面的“编辑”按键即可对菜品的详细信息进行编辑。如下图，更改了编号为48的菜品的名称。



（5）搜索：可以在文本框按照菜名或食堂进行搜索。如下图，筛选出了所有名称中带“饭”的菜品。



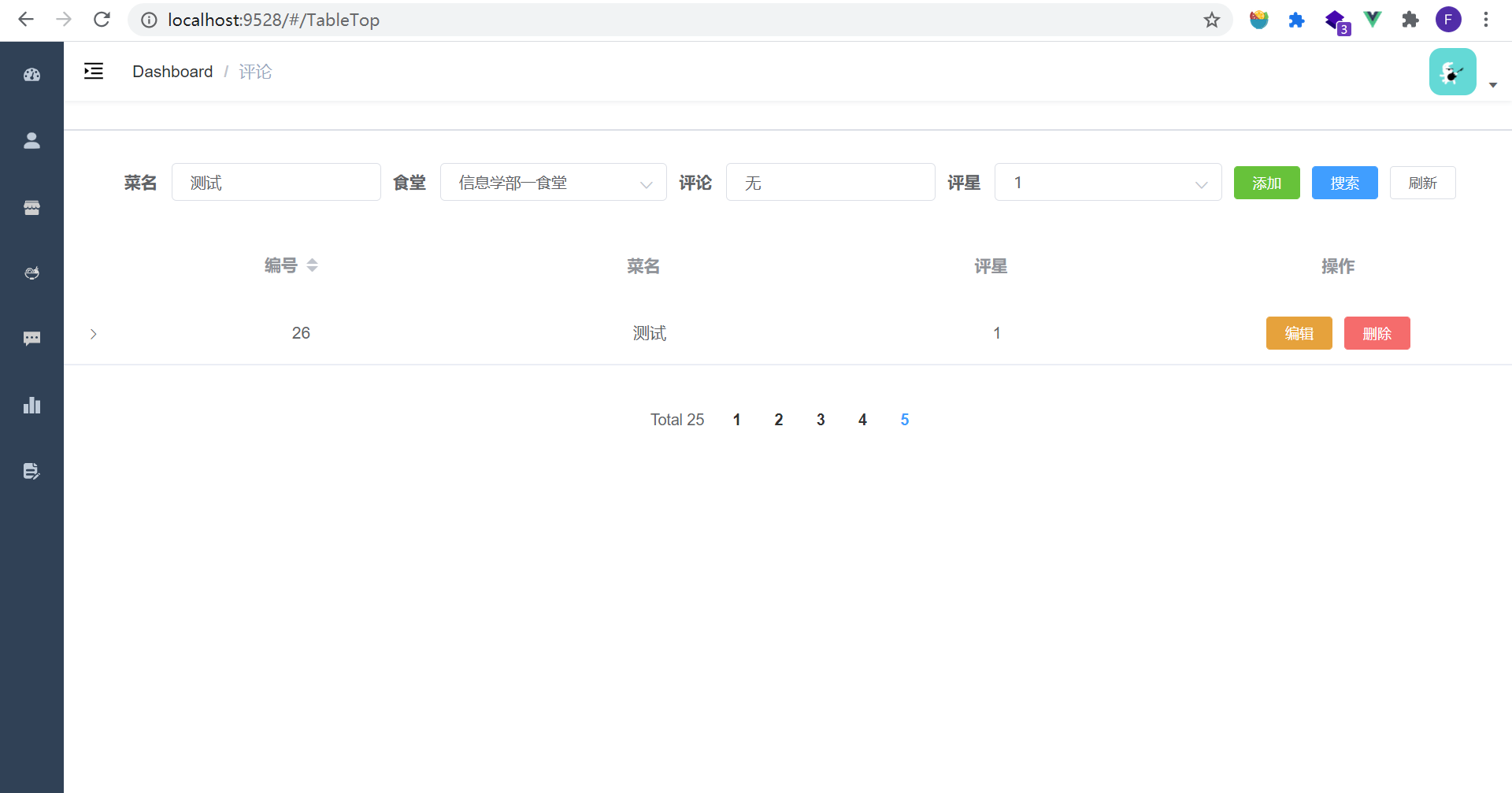
### 4. 评价界面

（1）查看：此界面可以查看所有用户对所有菜品做出的评价。展开可以查看详细内容，包括菜名、所在食堂、评分和评价内容。





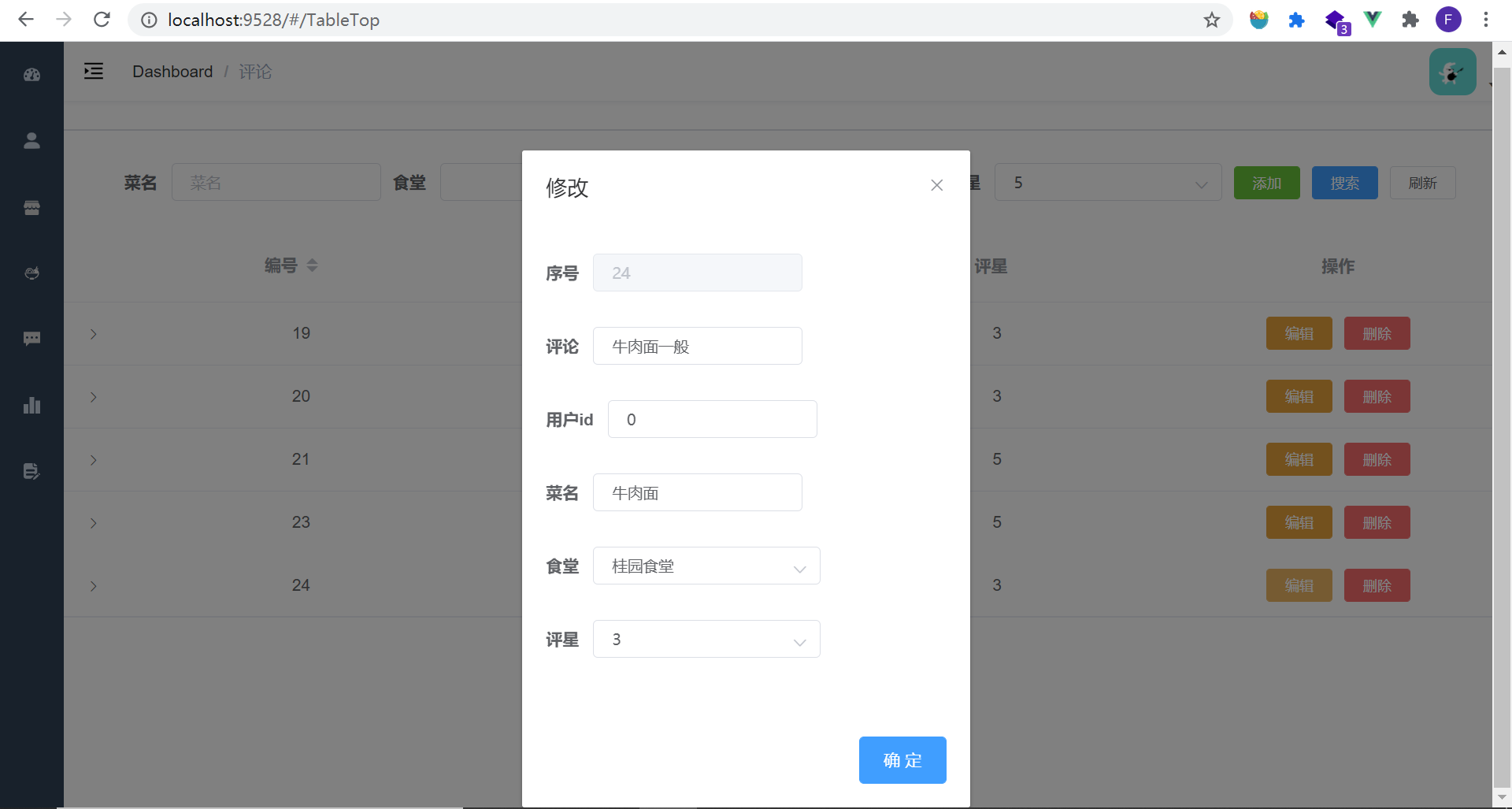
（2）添加：通过在文本框输入内容进行评价的添加。输入后，点击“添加”按键即可。



（3）删除：点击具体评价后面的“删除”按键即可将评论删除。如下图，删除了刚刚添加到26号评论。



（4）修改：点击“编辑”即可对评价内容进行修改。如下图，将牛肉面的评分改为5。

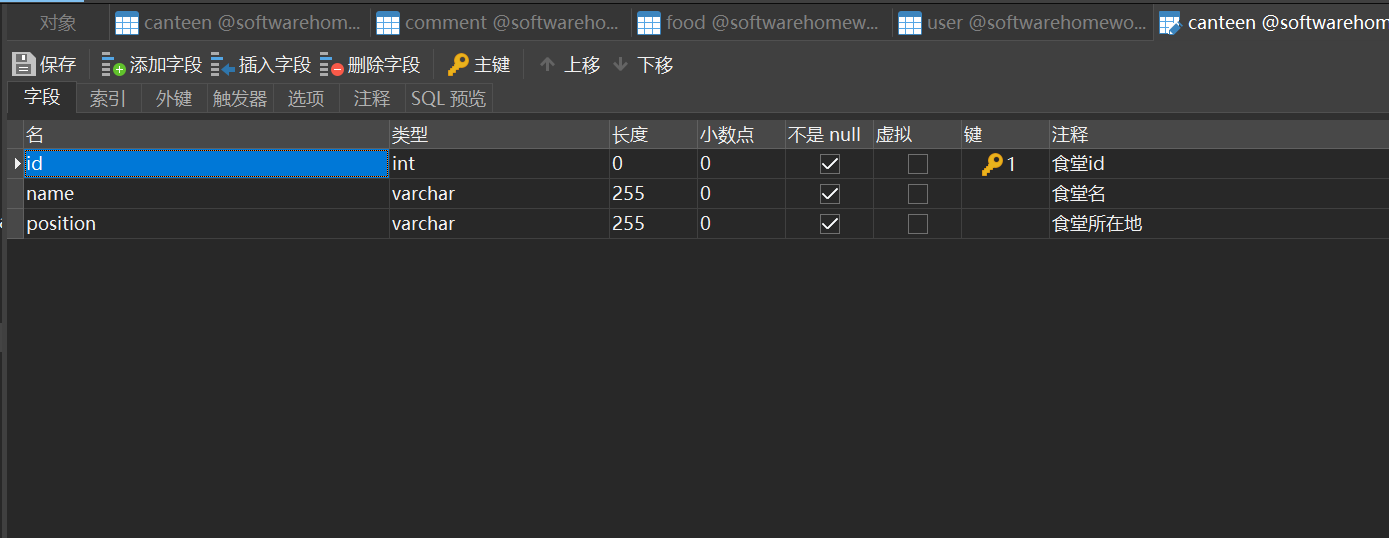


（5）搜索：可以通过菜名、食堂、评论内容和评星对评价内容进行搜索。如下图，筛选出评星为3的所有菜品。



**后端**

### 1. 数据库

图1-1 食堂信息对应的数据库

这里是食堂信息对应的数据库。可以看到表里一共有三个列为id, name和position,分别对应食堂的id，食堂的名字和食堂的位置。其中食堂id是int类型，name和position均为varchar。

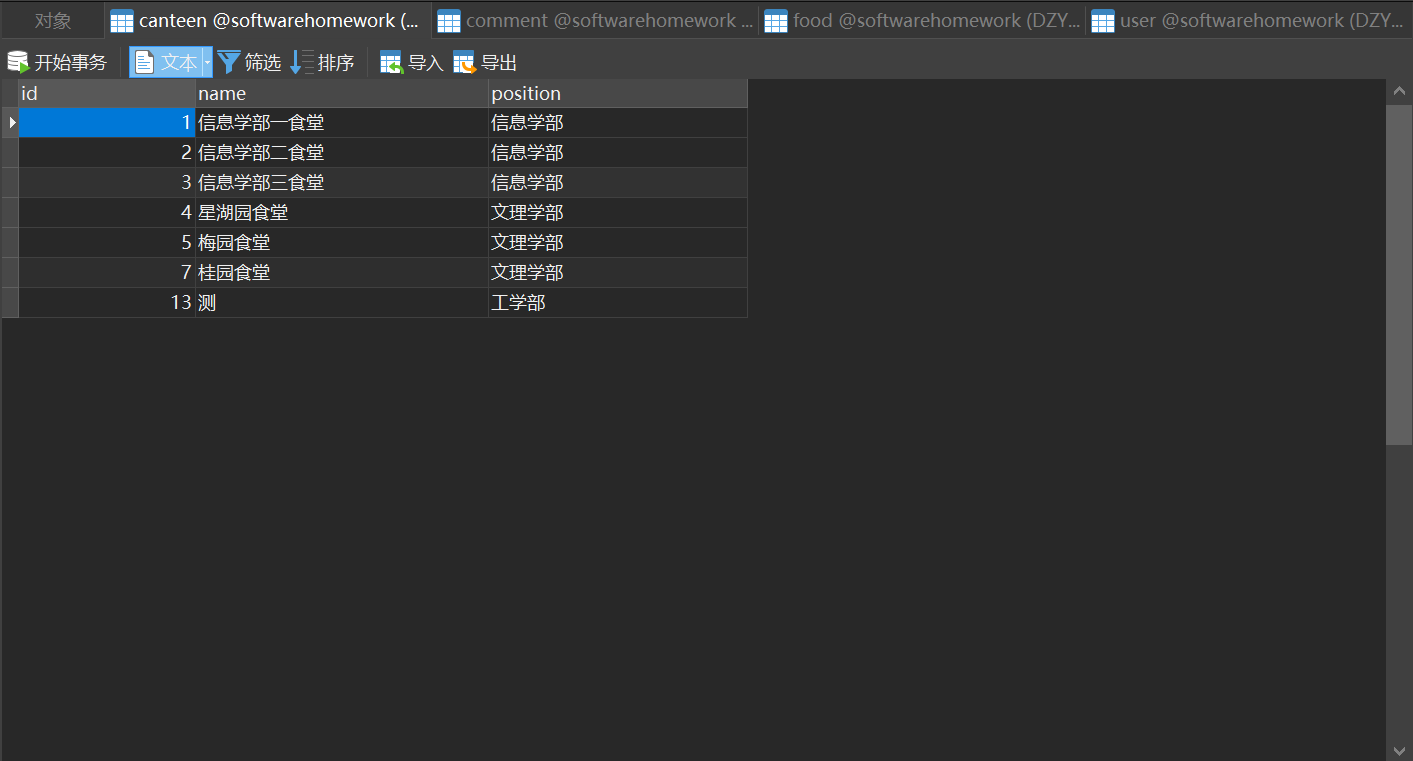


图1-2 数据库的表

这里是食堂数据库里现有的数据项。

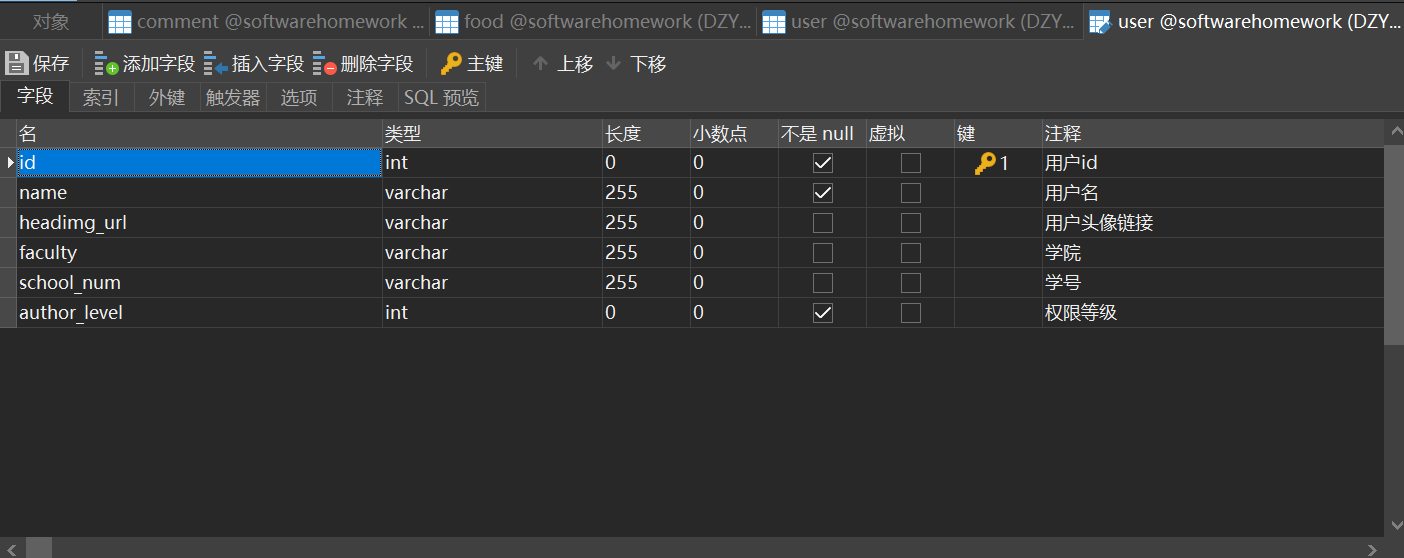
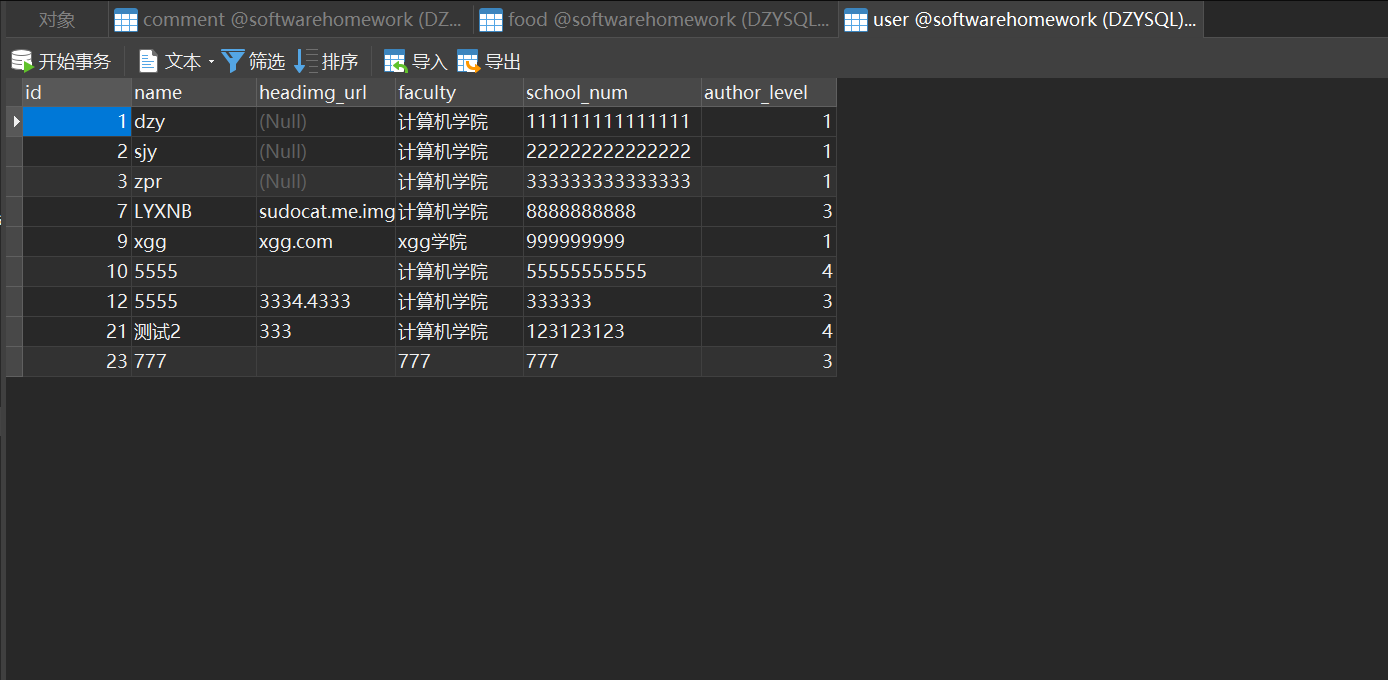
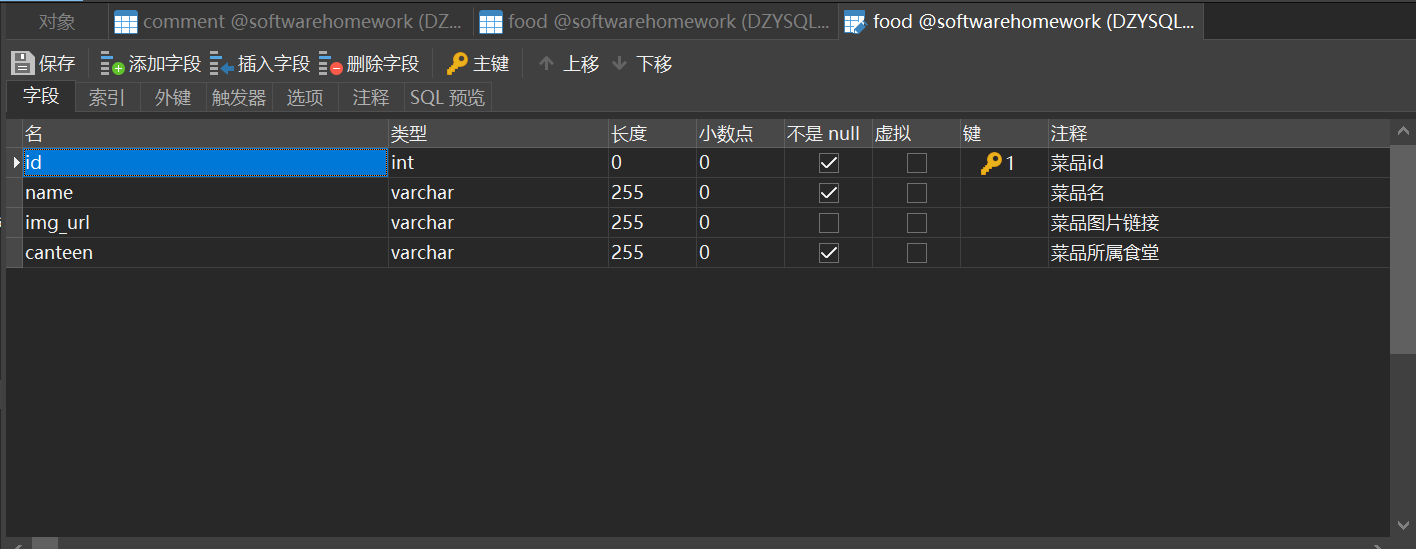


图1-3 用户表

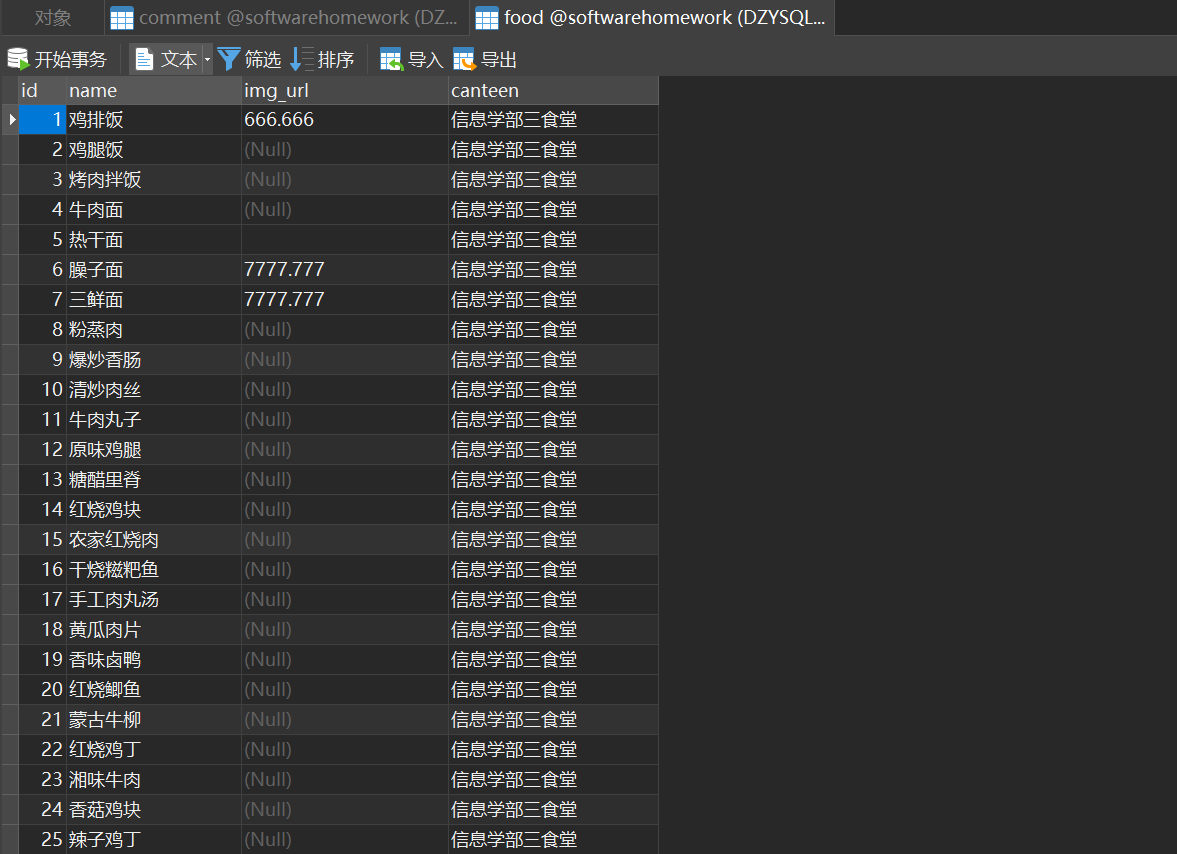
这个表是数据库的用户表。可以看到一共有六列，分别是id, name, heading \_url, faculty, school\_num和author\_level,分别表征用户id,用户名，用户的头像链接，用户所在学院，用户的学号以及 用户的权限等级。其中id和author\_level为int类型，其他的均为varchar类型。

图1-4 用户表的数据

这里是用户表的现有数据

图1-5 菜品表

这个表是菜品的表。该表有四列，分别为id,name, img\_url和canteen，分别表示菜品id，菜品名，菜品图片链接和菜品所属食堂。除id是int类型以外其他的都是varchar类型。

图1-6 菜品表数据

这个是菜品表中所有的数据

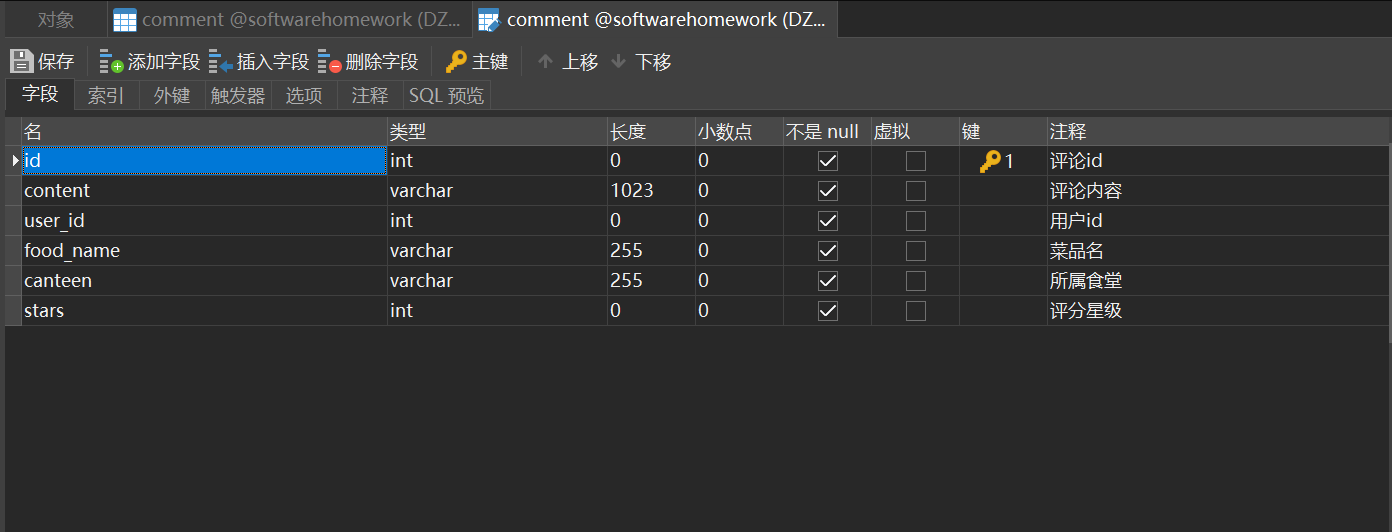
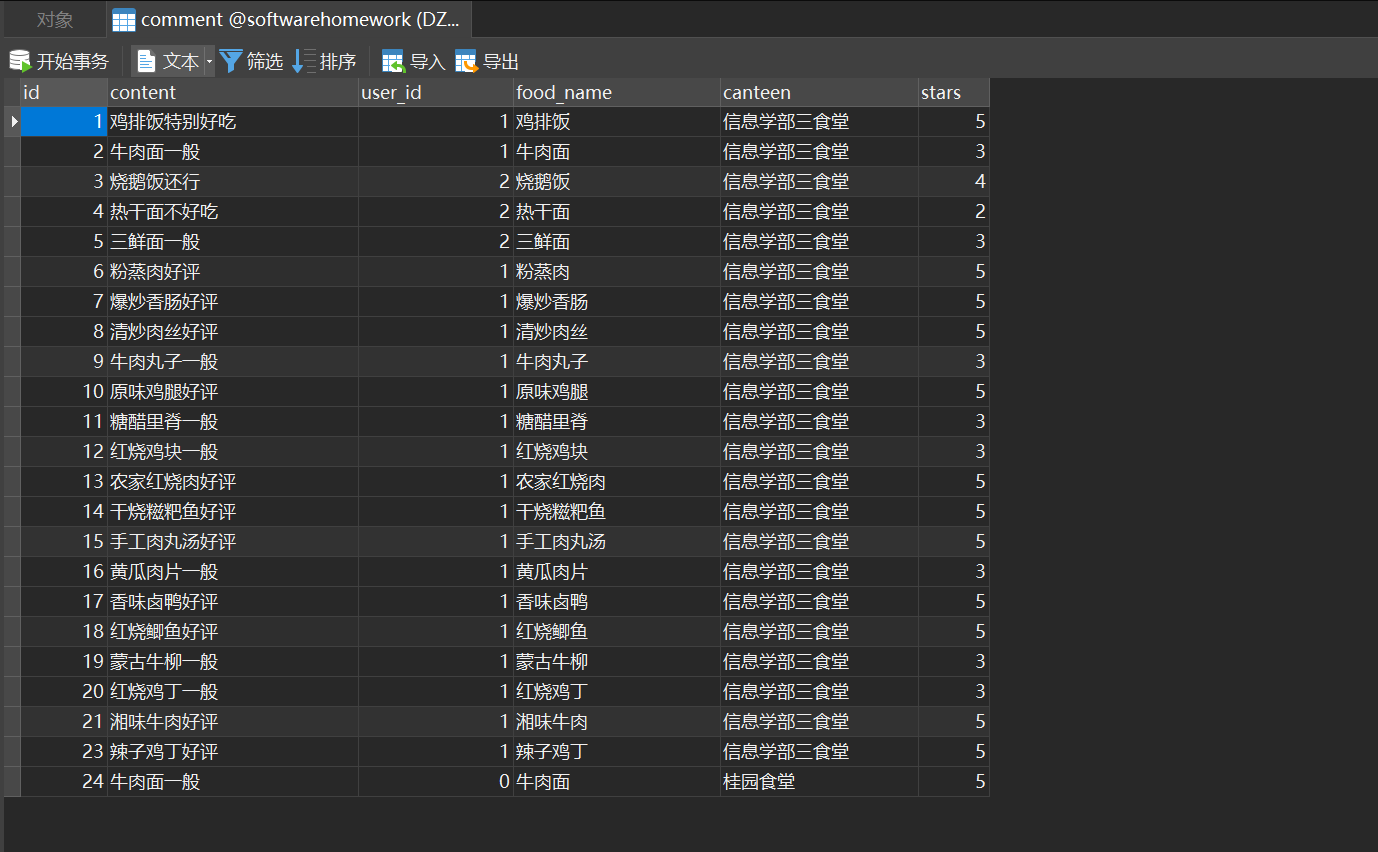


图1-7 评价表

这个表是对食物的评价表。里面有id，content，user\_id，food\_name，canteen，stars这六个属性，分别对应评论id，评论内容，用户id，菜品名，所属食堂和评分星级，id，user\_id和stars为int，其他均为varchar。

图1-8 评价表数据

这张图是食物评价表里现有的数据。

### 2. 后端API

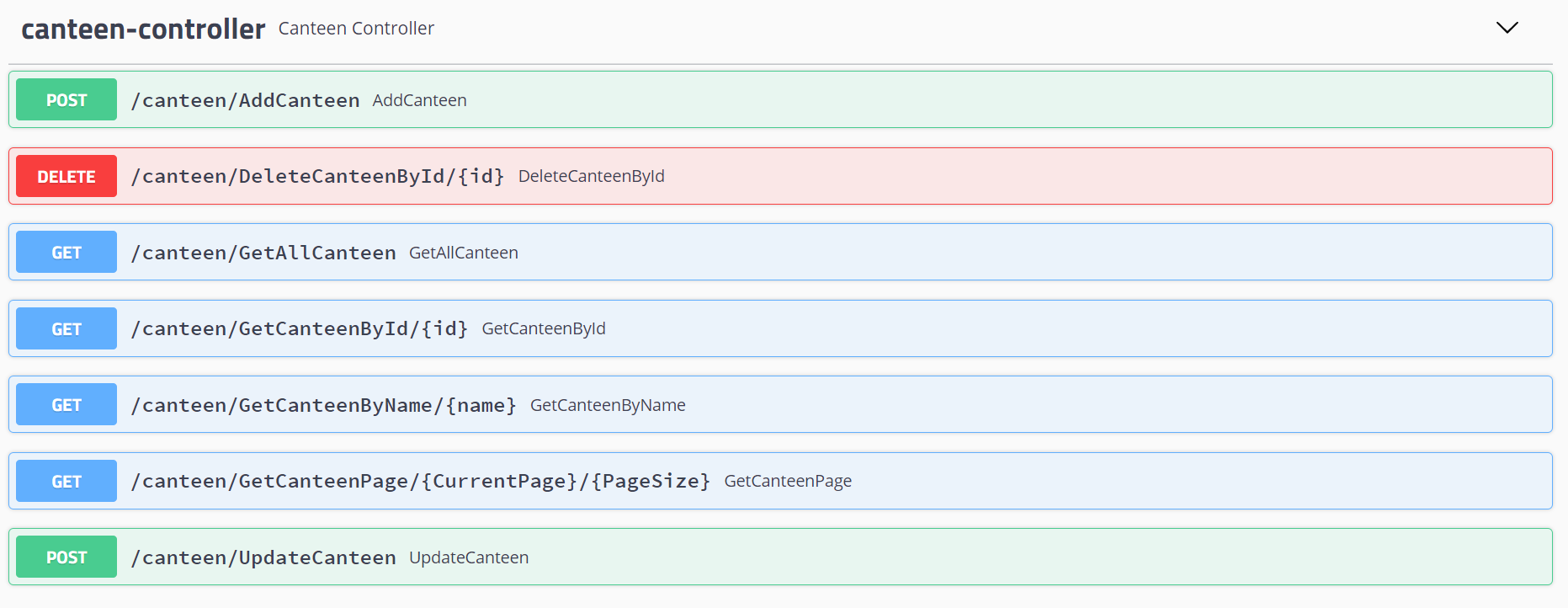


图2-1 食堂控制API

这里展示了对食堂控制的API

GetAllCanteen是获取所有的食堂信息

GetCanteenById是按照id获取对应的食堂信息

GetCanteenByName是按照食堂名获取对应的食堂

GetCanteenPage是获取对应页的食堂

UpdateCanteen负责更新食堂数据

AddCanteen负责添加对应的食堂数据

DeleteCanteenById是根据id删除对应的食堂数据

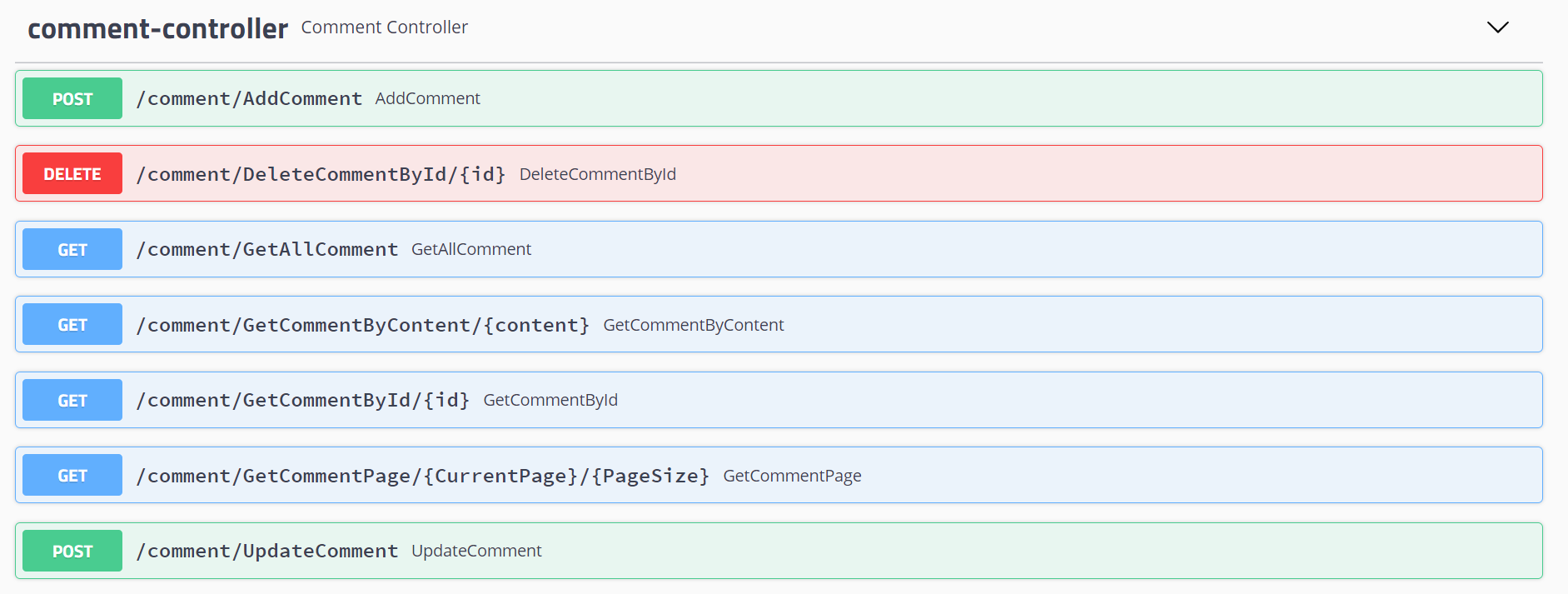


图2-2 评论控制API

这里展示了对评论控制的API

GetAllComment是获取所有的评论信息

GetCommentByContent是按照内容获取对应的评论信息

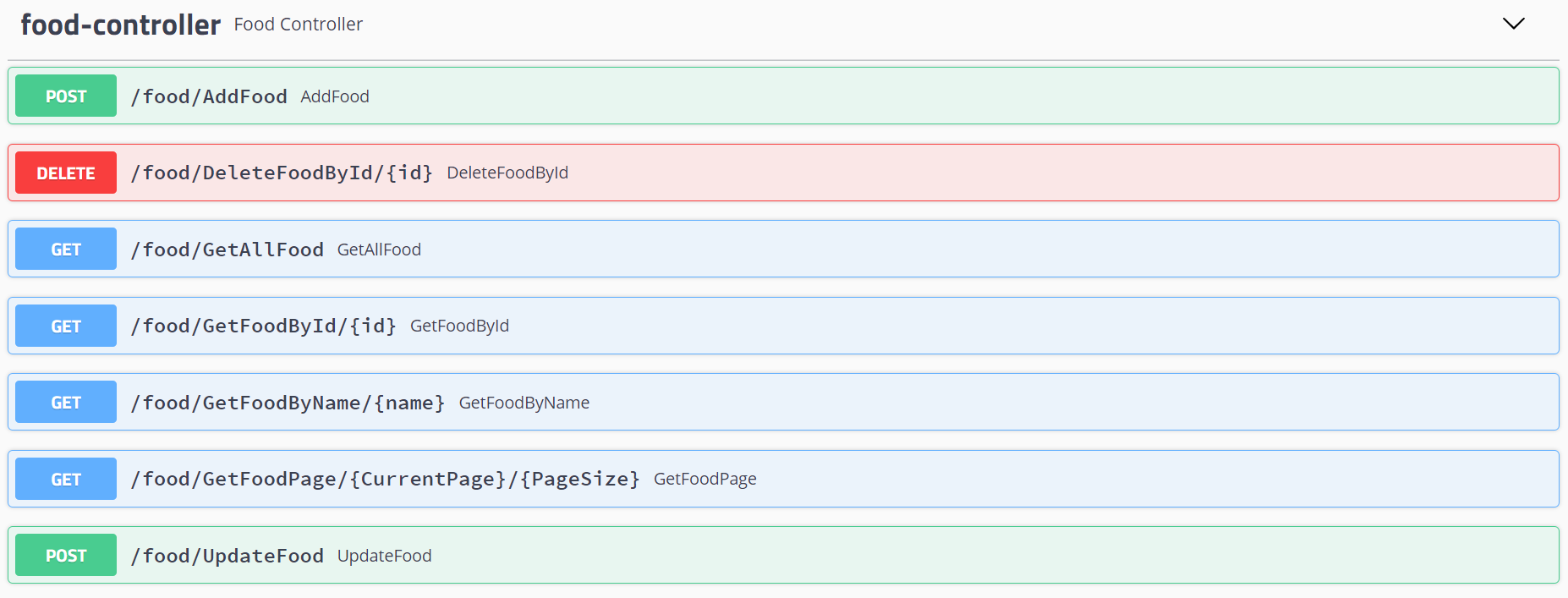
GetCommentById是按照评论id获取对应的评论

GetCommentPage是获取对应页的评论

UpdateComment负责更新评论数据

AddComment负责添加对应的评论数据

DeleteCommentById是根据id删除对应的评论数据

图2-3 食品控制API

这里展示了对食品控制的API。

GetAllFood是获取所有的食物信息

GetFoodByName是按照食物名字获取对应的食物信息

GetFoodById是按照食物id获取对应的食物

GetFoodPage是获取对应页的食物

UpdateFood负责更新食物数据

AddFood负责添加对应的食物数据

DeleteFoodById是根据id删除对应的食物数据。

图2-4 用户控制API

这里展示了对用户控制的API。

GetAllUser是获取所有的用户信息

GetUserByName是按照用户名字获取对应的用户信息

GetUserById是按照用户id获取对应的用户

GetUserPage是获取对应页的用户

UpdateUser负责更新用户数据

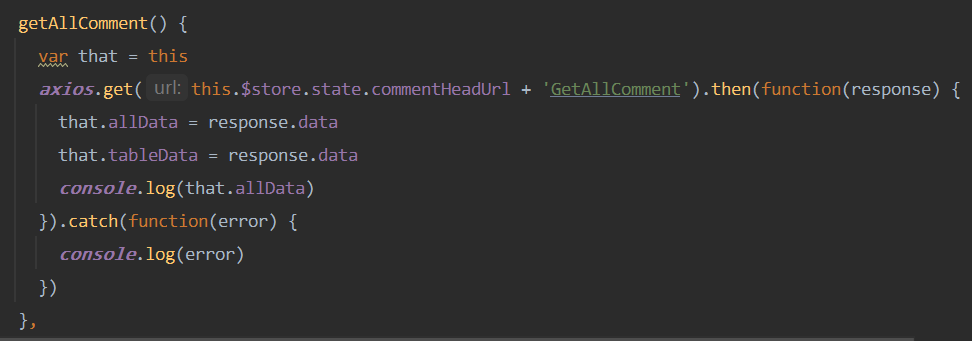
AddUser负责添加对应的用户数据

最后DeleteUserById是根据id删除对应的用户数据

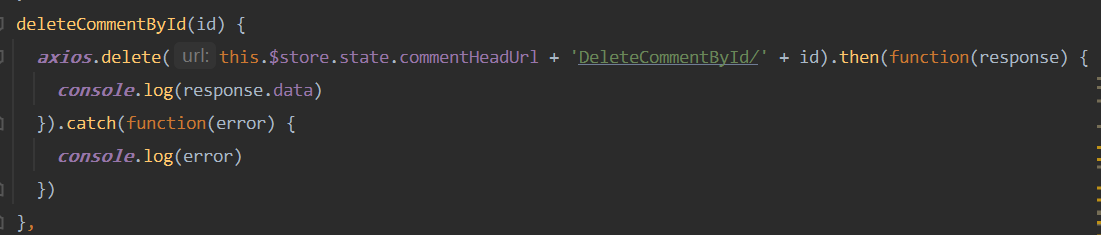
Redo是撤销上一步操作

### 3. 前端交互代码

（1）向后端查询对象数据并展示到前端

图3-1 前端的获取所有评论的API

（2）向后端发送请求删除该id对象

图3-2 前端按id删除对应评论的API

（3）向后端发送请求添加对象



图3-3 后端添加评论对象的API

（4）按输入框搜索对象

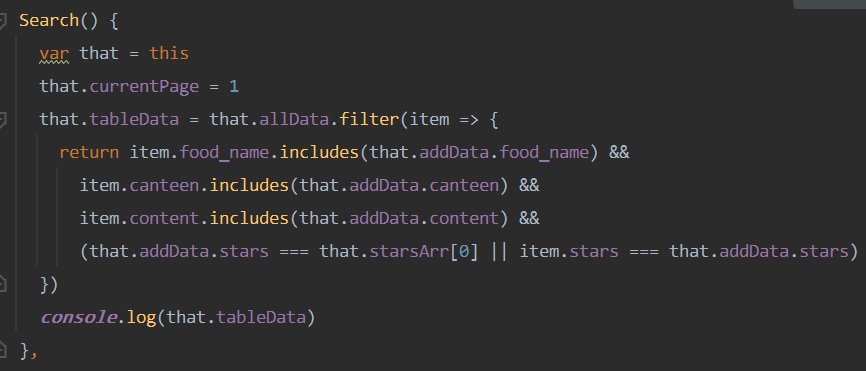


图3-4 前端搜索对象的API