



数据库课设 期末汇报

2025/01/05



目录



概述

当下外卖行业蓬勃发展，本项目旨在为外卖系统搭建一个用户端网站，实现基本的外卖逻辑。

1



项目框架

本项目使用经典的B/S架构，前端使用Vue，后端使用Flask，数据库使用MySQL。在Ubuntu下，使用VScode编写而成。

2



功能展示

本项目实现了注册登录、食物浏览、订单下单、订单汇总以及个人中心功能。

3



Part 1

概述

选题背景：随着外卖行业的迅速发展，餐饮服务领域的需求日益增长，特别是在数字化转型过程中，外卖软件成为人们日常生活中不可或缺的工具。外卖平台不仅仅为用户提供便捷的餐饮选择，更需要支持高效的订单处理、用户管理和商户协作。因此，开发一个稳定、高效的数据库系统，使之能在处理海量订单数据的同时，保证系统的可扩展性和稳定性，是外卖软件开发的核心任务之一。



问题描述：在本项目中，主要需实现外卖平台中的用户端功能。其中，最重要的点单功能属于一个典型的 OLTP 型业务。该种类型的业务主要处理大量简单的事务操作，通常具有以下特点：

1. 高频次：处理大量的短事务请求，例如下单、支付、修改订单状态等。
2. 实时性：要求数据的快速写入和读取，以便为用户提供实时的反馈。
3. 并发性：需要支持大量用户的并发访问。
4. 数据完整性：在高并发下确保数据的一致性。

因此，在本项目中设计的数据库需要满足实时响应和高并发事务处理的需求。

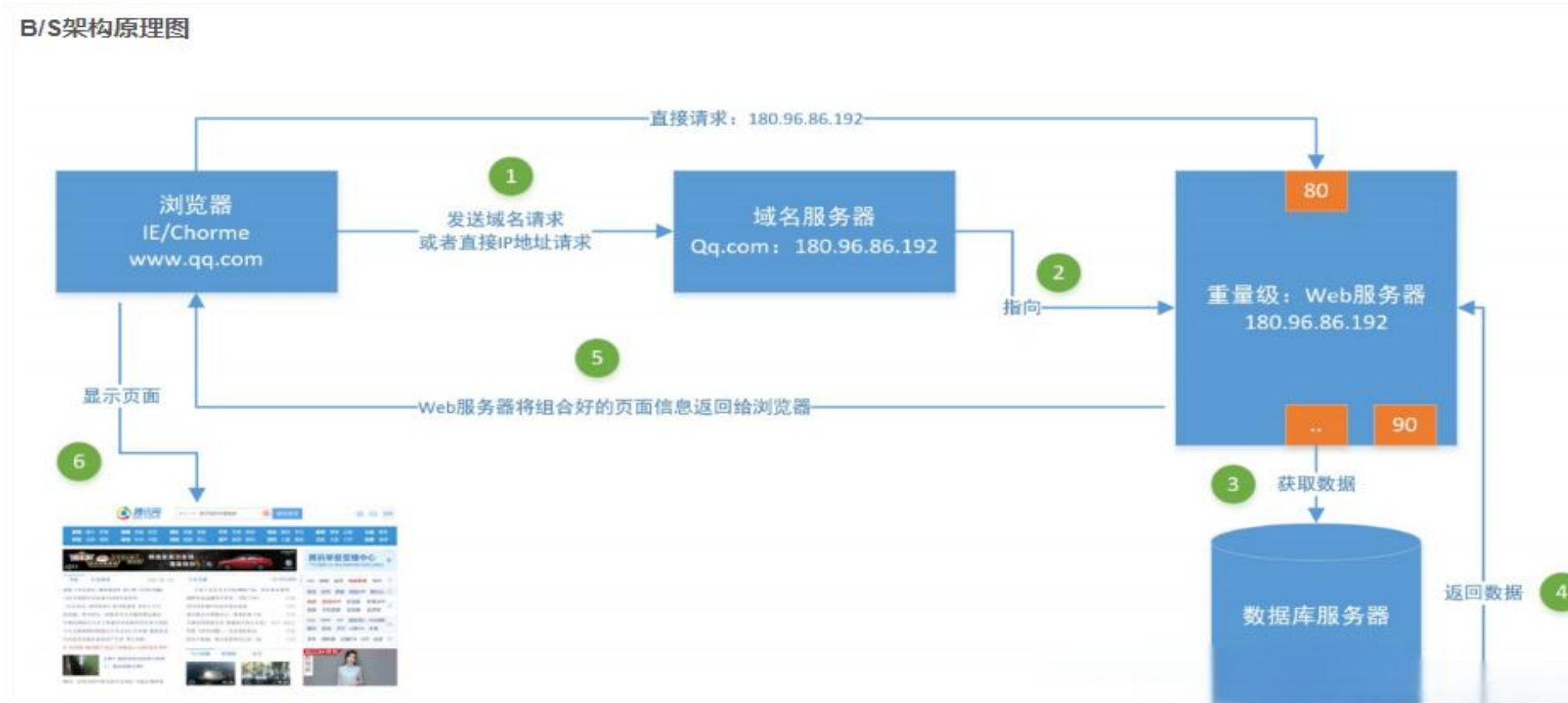




Part 2

项目框架

B/S 架构



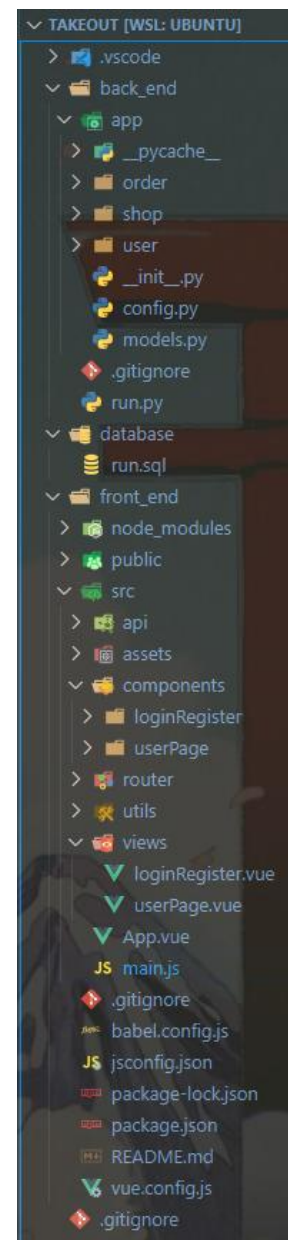
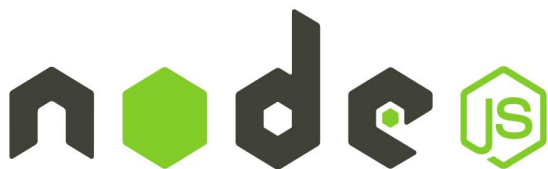
本项目使用经典的B/S架构，将软件分为：前端层、后端层以及数据库层。其中前端层负责页面展示和用户交互；后端层负责处理业务逻辑、数据管理以及与前端的接口交互；数据库层负责使用MySQL管理系统来存储核心数据。

解决方案

本项目使用 Vue + Flask + MySQL 开发，前端使用 Vue 框架管理，后端使用 Flask 管理，并在 NaviCat 中进行数据库的设计。

项目的所有代码均在 VScode 中编写，整个项目存储在 github 上，使用 git 作为版本控制工具。

本项目的开发过程跨平台进行，在 Ubuntu 中编写前后端代码，在 Windows 中进行数据库的设计。





Part 3

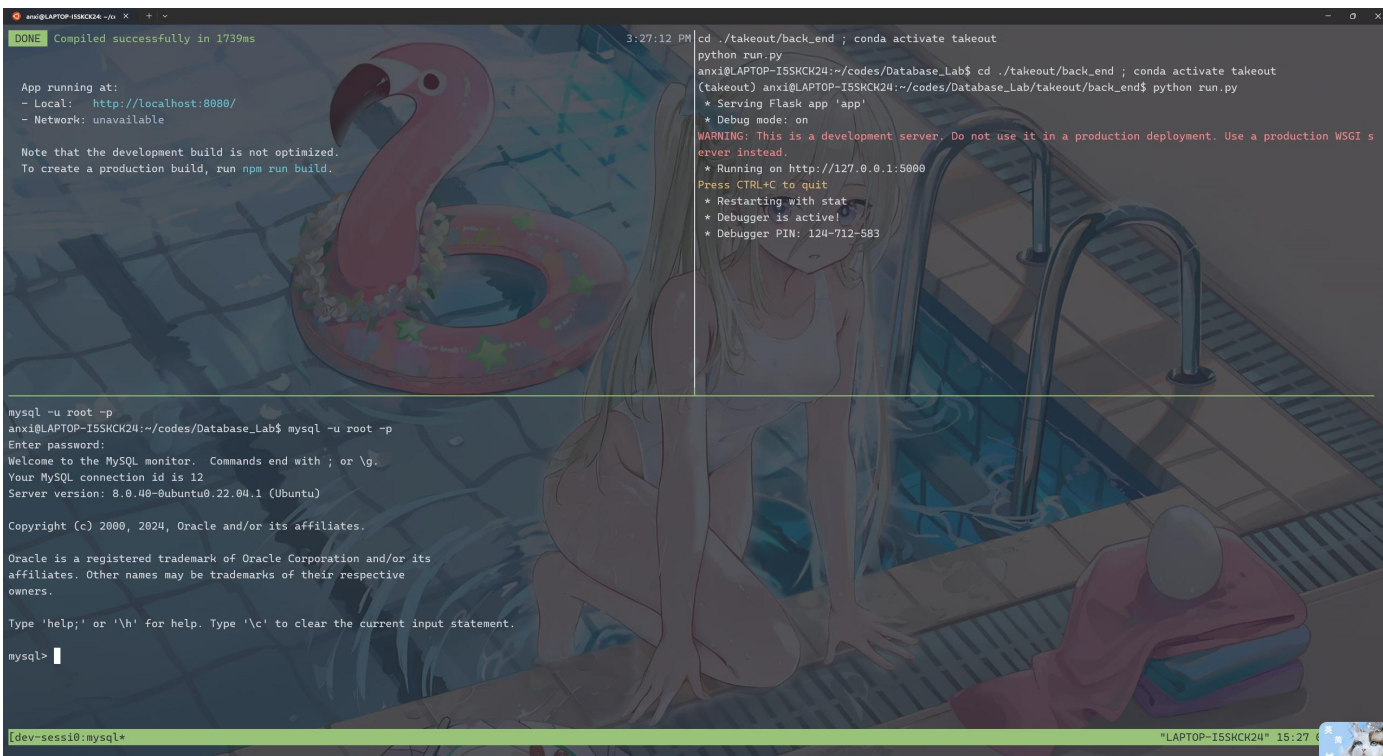
功能展示

项目启动及运行-p1

本项目在 Ubuntu 上完成，因此我编写了一个脚本文件 run.sh 用于项目一键启动。

这里使用了 tmux 对窗口进行划分，便于同一个窗口管理前端、后端和数据库三者。

具体的脚本代码如右图所示，使用 ./run.sh 即可运行本项目，结果如下图：



```
anxi@LAPTOP-I5SKCK24:~/codes/Database_Lab$ ./run.sh
DONE Compiled successfully in 1739ms

App running at:
- Local: http://localhost:8080/
- Network: unavailable

Note that the development build is not optimized.
To create a production build, run npm run build.

3:27:12 PM cd ./takeout/back_end ; conda activate takeout
python run.py
anxi@LAPTOP-I5SKCK24:~/codes/Database_Lab$ cd ./takeout/back_end ; conda activate takeout
(takeout) anxi@LAPTOP-I5SKCK24:~/codes/Database_Lab/takeout/back_end$ python run.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI s
erver instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 124-712-583

mysql -u root -p
anxi@LAPTOP-I5SKCK24:~/codes/Database_Lab$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.40-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

```
anxi@LAPTOP-I5SKCK24:~/codes/Database_Lab$ ls
README.md  doc  run.sh  takeout
anxi@LAPTOP-I5SKCK24:~/codes/Database_Lab$ cat run.sh
#!/bin/bash

# 创建一个新的 tmux 会话，命名为 'dev-session'
tmux new-session -d -s dev-session

# 在第一个窗口中运行前端代码
tmux send-keys -t dev-session 'cd ./takeout/front_end ; npm run serve' C-m

# 分割窗口，创建第二个窗口
tmux split-window -v
tmux send-keys 'mysql -u root -p' C-m

tmux select-pane -U

# 再次分割窗口，创建第三个窗口
tmux split-window -h
tmux send-keys 'cd ./takeout/back_end ; conda activate takeout' C-m
tmux send-keys 'python run.py' C-m

# 选择第二个窗口
tmux select-pane -t 2

# 附加到会话
tmux attach-session -t dev-session
```


项目启动及运行-p2

在确保 MySQL 服务器处于 activate 的情况下，在 Windows 中打开 NaviCat 连接上 Ubuntu 中的 MySQL 服务器对数据库进行可视化的操作。

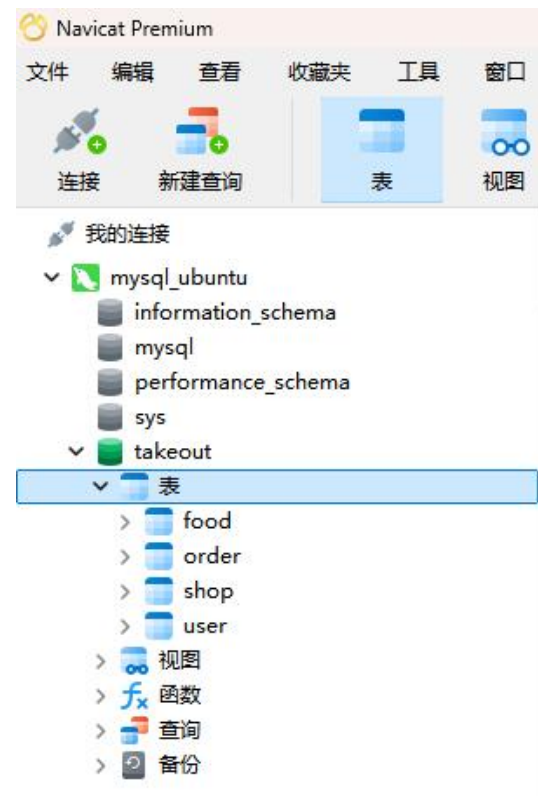
检查方式如下图所示，连接结果如右图：

```
anxi@LAPTOP-I5SKCK24:~/codes/Database_Lab$ service mysql status
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2025-01-05 13:36:15 CST; 1h 49min ago
     Process: 270 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
    Main PID: 376 (mysqld)
      Status: "Server is operational"
        Tasks: 41 (limit: 18861)
       Memory: 10.6M
          CPU: 34.561s
       CGroup: /system.slice/mysql.service
               └─376 /usr/sbin/mysqld

Jan 05 13:36:15 LAPTOP-I5SKCK24 systemd[1]: Starting MySQL Community Server...
Jan 05 13:36:15 LAPTOP-I5SKCK24 systemd[1]: Started MySQL Community Server.
```

可以看到，MySQL 服务器确实处于运行状态。若未处于运行状态，则可以使用 `service mysql start` 命令启动。

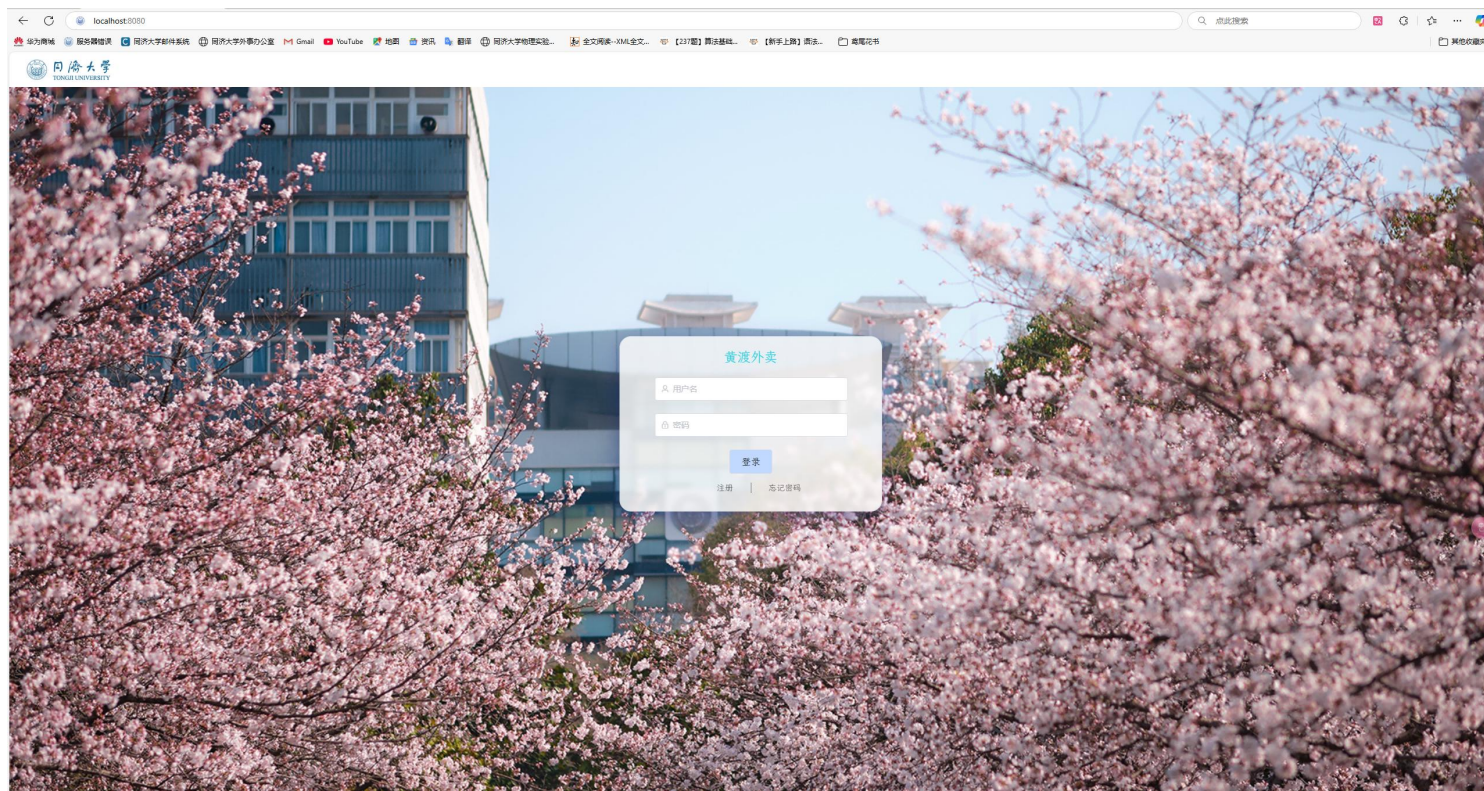
```
anxi@LAPTOP-I5SKCK24:~$ service mysql
Usage: /etc/init.d/mysql start|stop|restart|reload|force-reload|status
```



项目启动及运行-p3

启动之后，可以看到，本项目前端运行在 localhost:8080 端口；后端运行在 localhost:5000 端口；数据库的运行端口在后端代码中配置为 3306 端口。

在浏览器中输入 `http://localhost:8080` 可以自动跳转到登录界面。



```
DONE Compiled successfully in 1739ms

App running at:
- Local:   http://localhost:8080/
- Network: unavailable

Note that the development build is not optimized.
To create a production build, run npm run build.
```

```
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 124-712-583
```

```
You, 上个月 | 1 author (You)
class Config(object):
    """ 后端配置信息 """

    # 数据库的配置
    DIALCT    = "mysql"
    DRITVER   = "pymysql"
    HOST      = '127.0.0.1'
    PORT      = "3306"
    USERNAME  = "root"
    PASSWORD  = '          ' # 数据库的密码
    DBNAME    = 'takeout'
```


登录界面 - 注册 p1

进入登录界面之后可以先注册一个账户。按下注册按钮后，会跳转到注册账户窗口。在表单中填写数据之后，点击注册按钮系统首先会检测表单中的数据是否符合要求，若不符合则会给出提示。符合要求之后则会弹出二次确认按钮。确认注册之后，会根据提交的数据在右上角弹出对应弹窗，提示用户本次注册情况。



登录界面 - 注册 p2

注册之后，进入命令行界面，在 MySQL 服务器中查询 user 表可以发现，刚刚注册的账号已经成功加入了数据库中。

```
mysql> use takeout;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SELECT * FROM user;
+-----+-----+-----+-----+-----+-----+
| sex   | role  | username | realname | telephone | password |
+-----+-----+-----+-----+-----+-----+
|      | user  | anxi    |          | 18720650883 | 111111   |
+-----+-----+-----+-----+-----+-----+
```

成功注册之后，系统会自动跳转到登录界面。用户只需根据刚刚注册的账户进行登录即可。



登录界面 - 登录

第一次登录本系统需要完成账户注册，再次登录时就不需要进行注册。在登录界面中，同样设置了错误提示逻辑，提示用户本系统的密码规则限制，减少部分错误情况的发生。

按下登录按钮之后，同样会在右上角弹出一个窗口用于提示用户本次登录的情况。在登录成功后，会自动跳转到用户端界面。



登录界面 - 忘记密码 p1

若用户忘记密码则可以通过登录表单上的忘记密码按钮跳转到修改密码界面，在该界面中，可以通过预留的手机号进行密码的修改操作。

在该界面中，同样存在表单验证、二次确认、弹窗提示以及自动跳转功能。



登录界面 - 忘记密码 p2

修改密码之后，进入命令行界面，在 MySQL 服务器中查询 user 表可以发现，刚刚的修改已经成功记录在数据库中。

```
mysql> SELECT * FROM user;
+-----+-----+-----+-----+-----+-----+
| sex   | role  | username | realname | telephone | password |
+-----+-----+-----+-----+-----+-----+
|      | user  | anx1     |          | 18720650883 | 222222   |
|      | user  | anx11    |          | 13767580842 | 111111   |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

成功修改密码之后，系统会自动跳转到登录界面。用户只需根据刚刚修改的密码进行登录即可。



用户界面 - 美食广场

在进入用户界面之后，首先跳转到的是美食广场界面。在该界面中，可以通过上方的输入框来查找商家，并选中商家显示其菜单。

在搜索框中实现了关键字联想功能，可以提示用户商家名称，效果如右下图所示。



用户界面 - 待收货订单

本系统将订单分为了三类：待收货订单、已完成订单和未发货订单，首先接收待收货订单。在该界面中，使用了一个表格统计待收货订单信息，并设置了一个数据统计功能，用于统计所有订单的总金额，方便用户查看总支出。

待收货订单

店铺名称	食物名称	订单金额	配送方式	订餐人姓名	送餐地址
豪大大香鸡排	可乐鸡排套餐	21	外卖	王五	友园四号楼
豪大大香鸡排	汉堡套餐	20	外卖	王五	友园四号楼
豪大大香鸡排	香辣鸡排	12	自提	王五	友园四号楼
小麻鲜	麻辣小龙虾套餐	30	外卖	李四	友园六号楼
小麻鲜	香辣烤鱼饭	15	自提	李四	友园六号楼
合计		98			

用户界面 - 已完成订单

该页面和待收货订单类似，同样设置了一个数据统计功能，便于用户查看总支出。

已完成订单

店铺名称	食物名称	订单金额	配送方式	订餐人姓名	送餐地址
奈哥酸菜鱼	酸菜鱼	25	外卖	张三	友园六号楼
奈哥酸菜鱼	麻辣烤鱼	32	自提	李四	友园四号楼
奈哥酸菜鱼	麻辣鱼	25	外卖	张三	友园六号楼
豪大大香鸡排	香辣鸡排	12	外卖	王五	友园五号楼
豪大大香鸡排	原味鸡排	10	自提	赵六	友园三号楼
熊大爷现包饺子	鲜肉饺子	10	外卖	张三	友园四号楼
熊大爷现包饺子	韭菜鸡蛋饺子	10	外卖	张三	友园四号楼
小麻鲜	香辣烤鱼	32	外卖	李四	友园四号楼
合计		156			

用户界面 - 未发货订单

系统将未发货订单分为了两类：未接单和制作中，并在表格中用不同颜色加以标识，所有刚下单的订单都处于未接单状态显示在该页面中。该页面同样添加了数据统计功能。

未发货订单

店铺名称	食物名称	订单金额	配送方式	订餐人姓名	送餐地址	订单状态
熊大爷现包饺子	鲜肉饺子	10	外卖	张三	友园四号楼	未接单
熊大爷现包饺子	韭菜鸡蛋饺子	10	外卖	张三	友园四号楼	制作中
小麻鲜	香辣烤鱼饭	15	外卖	李四	友园四号楼	未接单
小麻鲜	香辣烤鱼饭	15	外卖	李四	友园四号楼	制作中
合计		50				

用户界面 - 个人中心

在个人中心提供了修改个人信息的服务，修改的信息在确认后会同步到数据库中。初始时的信息如右图所示，修改后在命令行中查看 user 表，可以发现数据成功存储到了数据库中。

```
mysql> SELECT * FROM user;
+-----+-----+-----+-----+-----+-----+
| sex  | role | username | realname | telephone | password |
+-----+-----+-----+-----+-----+-----+
| 男   | user | anxi     | 徐宏     | 18720650883 | 222222   |
|      | user | anxil    |          | 13767580842 | 111111   |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

用户名

anxi

姓名

性别

电话

18720650883

开始修改

确认修改

用户名

anxi

姓名

徐宏

性别

男

电话

18720650883

开始修改

确认修改

汇报到此结束
谢谢

