

# Tongji校园助手

部署:

01 项目介绍

02 技术(需要补充, 更详细些)

03 功能描述

用户界面

后台 SimpleUI

数据库 MySQL

03 特色与创新

04 未完善

05 参考

## 部署:

Tongji 校园助手

先打开setting.py修改数据库信息, 账号密码名称等文件夹下有我的数据库数据eco.sql, 可以使用navicat复制到你的数据库中。

1.Anconda创建虚拟环境

conda creat -n eco python=3.8

2.激活虚拟环境

conda activate eco

3.安装依赖文件

pip install -r requirements.txt

使用后台可能还需要pip install simpleUI

4.迁移文件

python

manage.py makemigrations

python

manage.py migrate

5.运行

py

`manage.py runserver`

**127.0.0.1:8000 前端界面**

**127.0.0.1::8000/admin 后台界面**

## 01 项目介绍

Tongji校园助手旨在为学校社区提供一个平台，学生和教职工可以在这里发布和接取跑腿任务，如代拿快递/外卖，出售学习资料，失物招领等。同时实现了一些交互功能，例如评论，用户注册（邮箱验证），信息发布等

分为四个板块:前端用户界面，后端处理，数据库以及后台管理员面板

REQ：用户请求

OD：接受任务

**角色：**

- 用户：可以在前端页面同时充当REQer和ODer，派出和接受任务。
- 管理员：可以从127.0.0.1:8000/admin进入后台界面，可以实现用户/REQ/评论/地址等的增删改。

## 02 技术(需要补充，更详细些)

本应用后端采用Django体系结构（基于模型Model-模版Template-视图-View模式的开源框架）

数据库采用MySQL数据库，使用Django进行数据库的交互，可以通过POST和GET请求，在前端和数据库之间进行数据传输，数据库与前端信息的修改是同步的。

后台使用轻量化框架SimpleUI-admin实现，并进行了页面，功能的修改，提升了交互性。

- 前端：HTML+CSS+JavaScript，利用 Ajax 实现页面的动态交互(不太清楚，应该是Ajax，没有vue,好像还有Bootstrap-简洁、直观、强悍的前端开发框架，让web开发更迅速、简单。)
- 后端：Django 框架，用于处理数据交互和业务逻辑。
- 数据库：MySQL，存储用户资料、任务信息、地址和交易记录等

## 03 功能描述

### 用户界面

- 用户注册（SMTP邮箱验证）
- 用户登录
- 密码重置
- 账户个人信息管理(联系方式/地址/简介等)
- 添加和删除REQ(图片等显示)
- 添加到OD List
- 添加到收藏夹
- 按名称搜索REQ
- 按类别筛选
- 作为ODer接受REQ，选择交易方式和地址等
- 发布评论（任何人评论/ODer评论）
- 查看REQ状态（确认完成/评论/是否派出等）
- 查看REQer个人简介信息
- 部分功能登录用户可用(增强逻辑性)

等，功能很多，也有些小bug，可以多试试

### 后台 SimpleUI

- 后台登录
- 管理REQ类别，增/删/改
- 管理用户
- 管理OD
- 修改密码
- 日志记录，查看管理员最近动作

- 管理交易方式
- 认证和授权，增加用户/管理员分组，权限分配
- 增删改评论

等，功能挺完善的

## 数据库 MySQL

- 增删改字段
- 日志记录

等，navicat打开数据库截个图，挨个表介绍下吧

## 03 特色与创新

- 安全性：CSRF跨站请求伪造漏洞；邮箱注册，极大程度阻挡非法用户，后续可以考虑仅同济邮箱注册；密码，用户名等文本输入字段的合法性检查，防止SQL注入攻击
- 前后端与数据库分离，信息处理不暴露给前端，保护用户数据
- 可扩展性：可以使用Django模版新建许多app，使得功能更强大
- 美观：界面使用CSS美化
- 模块化，面向对象编程，构建了许多类，比如评论类，商品类等，可维护性强
- 等等等，还是挺多的

## 04 未完善

- 付款机制
- 退款机制
- 申诉机制

- 订单跟进，例如邮件通知等
- 信息在数据库中明文存储，后续可以添加RSA等算法进行加密
- ip记录以及请求头检查还待完善
- 等等等，扯扯就行

## 05 参考

参考开源教学项目：**Django E-commerce Tutorial Series**