## 项目结构说明

项目技术栈：

Python 3.8

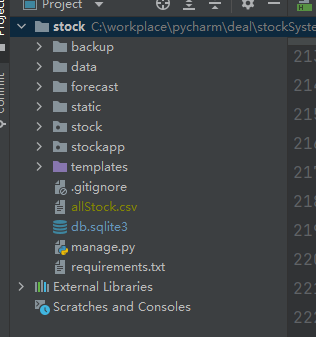
Web框架：Django 3

数据库存储技术：Dataframe文件存储

深度学习框架：keras

前端技术：bootstrap4，jquery,ajax

项目结构：



结构说明：

-stock：Django项目的主app

-stockapp：实现股票展示、预测、更行等操作的app

-forecast:存放各种预测算法（系统用的是其中的lstm进行预测，因为他的效果最好）

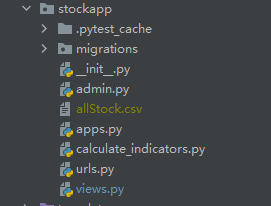
-static：存放系统各种静态资源（js，css）

-templates:存放前端html页面

-data：ts\_code.csv(代码为ts\_code的股票的数据存放文件)，allStock.csv(存放所有股票的简要信息)

-backup：我写代码的适合备份的一些文件，你不用看，没有用处

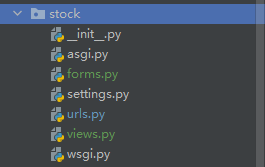
重点说明stockapp下的各种文件作用



.pytest\_cache：系统缓存文件

migrations/：此目录包含Django应用程序的数据库迁移文件。每当您更改数据库表时，都应该创建一个新的迁移文件，并运行migrate命令来将更改应用于数据库。app.py：此文件定义了Django应用程序的配置。您可以在这里指定应用程序的名称，URL配置和其他设置。url.py：此文件包含Django应用程序的URL配置。您可以在这里定义哪些URL应该与哪些视图函数匹配，并将它们映射到适当的视图函数。views.py：此文件包含Django应用程序的视图函数。视图函数处理来自URL的请求，并根据需要从数据库中检索数据，并将其呈现为HTML响应。calculate\_indicators.py：计算股票的各种指标

Stock下各种文件的作用：

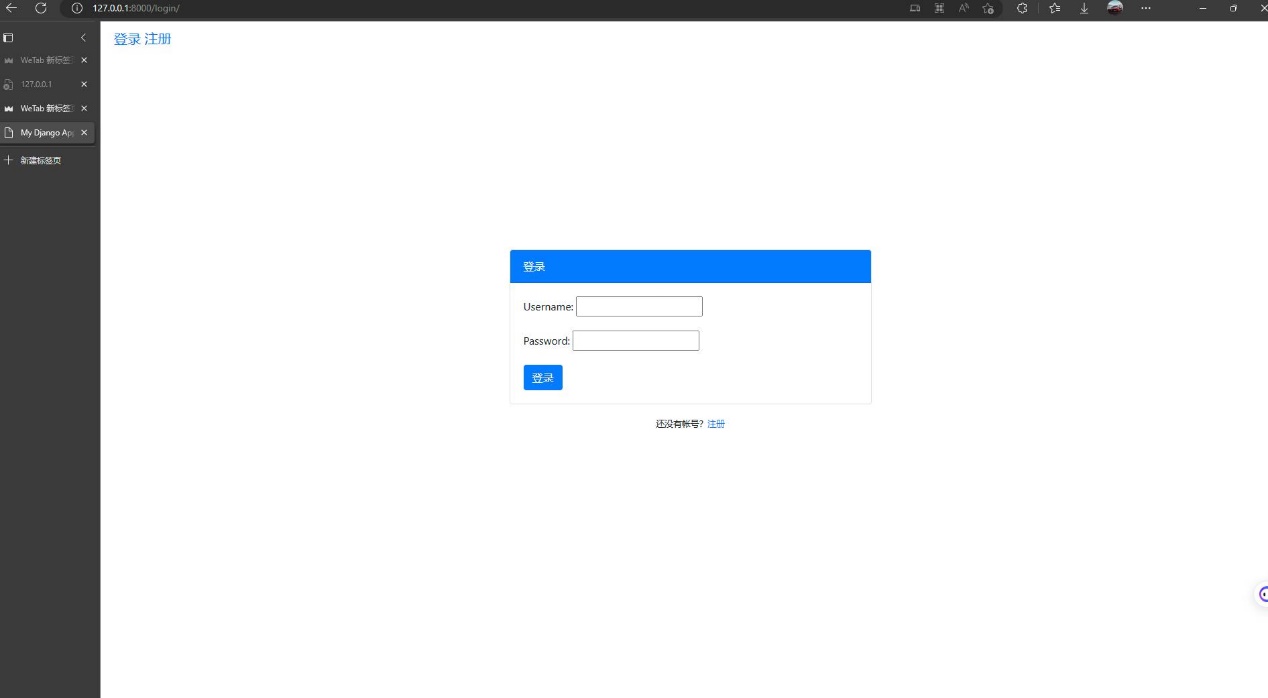


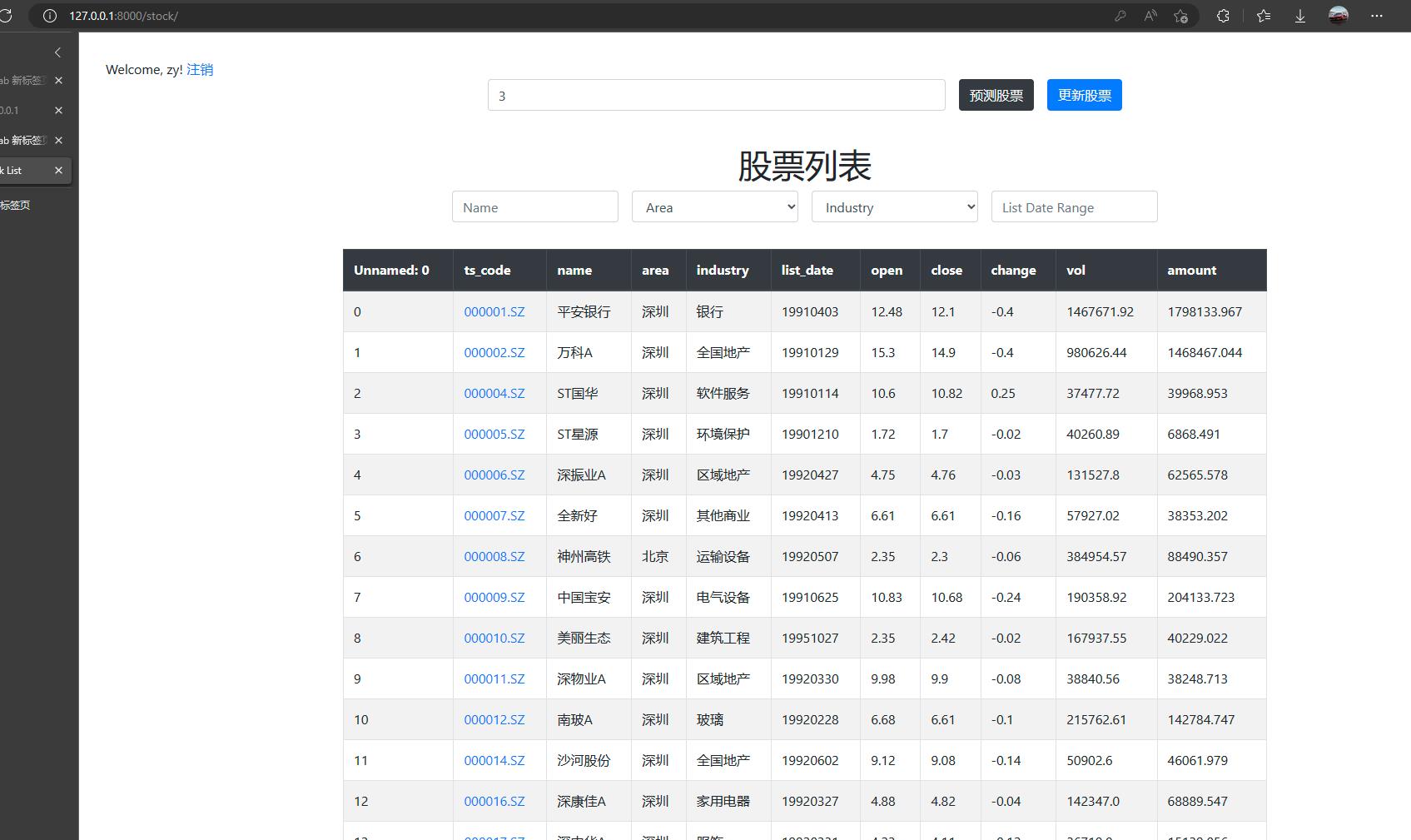
* settings.py：此文件包含Django项目的所有设置。您可以在这里指定数据库配置，静态文件路径，安装的应用程序列表等等。
* forms.py：此文件包含Django应用程序的表单类。表单类用于验证用户提交的数据，并将其转换为模型实例。
* manage.py：此文件是Django项目的命令行工具。您可以使用它来运行服务器，创建数据库迁移，运行测试等等。

运行方式：

python manage.py runserver

## 项目效果展示











## 使用注意事项

主页最上面的两个按钮是预测和更新股票的。因为数据量很大，所以这两个操作都非常费时间。左边的框是输入需要预测多少天后的股票价格，系统对10支股票进行了预测，并按照收益进行了排序。目前预测10支股票在我的电脑上需要**73s**。如果你想要预测更多的股票，就更慢了。之所以这么慢第一是因为数据量大，第二是因为我们做的是在线预测，用的实时的数据。

**在进行股票预测和更新，不要进行其他操作，否则会更新或者预测失败。**

其次就是更行股票了，就是从网上获取最新的股票数据更新到心里的data文件夹中，保持数据的实时性。同样地，因为有5000多个股票，所以更行股票操作需要**1个多小时**，比较慢。