

****

**信 息 学 院**

**程序设计实践A课程项目开发报告**

**课程项目 城市天气数据采集与分析展示系统**

**——基于Scrapy采集分析日本各城市天气**

**姓名学号 潘宇科 37320222204170**

**实习时间 2024.6.25 — 2024.7.18**

**2024年 7 月 25 日**

1. **项目目标与要求**

* 项目目标：

通过该综合项目， 全面掌握Scrapy数据爬取、 数据处理与可视化、 Web开发等技术， 并具备整合多项技术解决实际问题的能力。

* 项目描述

设计并实现一个综合项目， 要求使用Scrapy爬取数据， 对数据进行处理与可视化， 并通过一个Web应用展示采集到的信息和分析结果。

* 项目要求

1. 数据采集

• 使用Scrapy爬取指定网站的数据。

• 需要爬取的信息根据项目具体要求而定， 如： 标题、 日期、 内容、 评分等。

• 数据存储到数据库（如SQLite或MongoDB） 中。

2. 数据处理与分析

• 使用Pandas或其他数据处理工具对采集到的数据进行清洗和整理。

• 对数据进行统计分析或情感分析等， 生成相应的分析结果。

• 使用Matplotlib或其他可视化工具对分析结果进行可视化展示。

3. Web应用开发

• 使用Flask或Django等Web框架开发一个Web应用。

• 实现一个Web界面， 展示数据和分析结果。

• 提供搜索和过滤功能， 例如根据特定字段进行搜索和筛选。

• 将数据可视化图表嵌入到Web页面中， 提供分析结果的展示。

4. 文档与演示

• 撰写项目文档， 详细描述项目的设计、 实现过程和使用方法。

• 录制项目演示视频， 讲解项目的功能和特点， 并制作演示PPT。

• 完整材料打包（文件命名格式： 学号\_姓名\_项目名称） 上传至lnt.xmu.edu.cn的期末大作业提交处

评估标准

• 数据采集的完整性和准确性。• 数据处理与分析的合理性和准确性。• 数据可视化的合理性和美观性。• Web应用的功能性、 用户体验和界面设计。• 项目文档的完整性和规范性。• 项目演示视频的清晰度和表现力， 包括PPT内容的详细讲解和实际项目功能的展示。

1. **项目内容**

**（一）、功能需求**

1、城市天气数据采集与分析展示系统

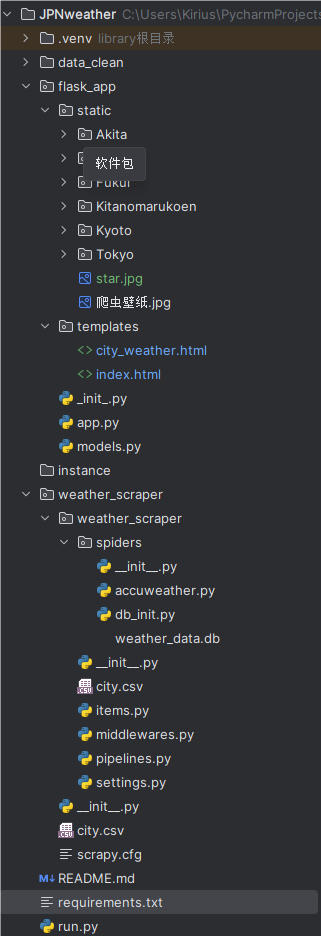
（1）、数据采集： 使用Scrapy爬取多个城市的天气数据。

（2）、数据处理与可视化： 清洗整理数据， 并生成温度变化图、 湿度变化图等。

（3）、Web应用开发： 使用Flask或Django开发Web应用， 展示天气信息和数据分析结果。

（二**）、系统设计**

**1.项目结构**



图一 项目结构图

Dataclean包含转换过的csv类型的城市天气的文件与clean.py（将数据库中数据导出并进行数据清洗）。

Falskapp中的static存放生成的图标以及网页布局所用到的图片，templates存放创建的html模板，包括一个首页与一个城市天气的详情页面模板。

Weatherscraper则是所写的爬虫对accuweather网站中的天气信息进行抓取并且插入数据库weather\_db中。

根目录下有一个city.csv是我记录的各个城市对应的url网址，方便爬虫进行相应的爬取。

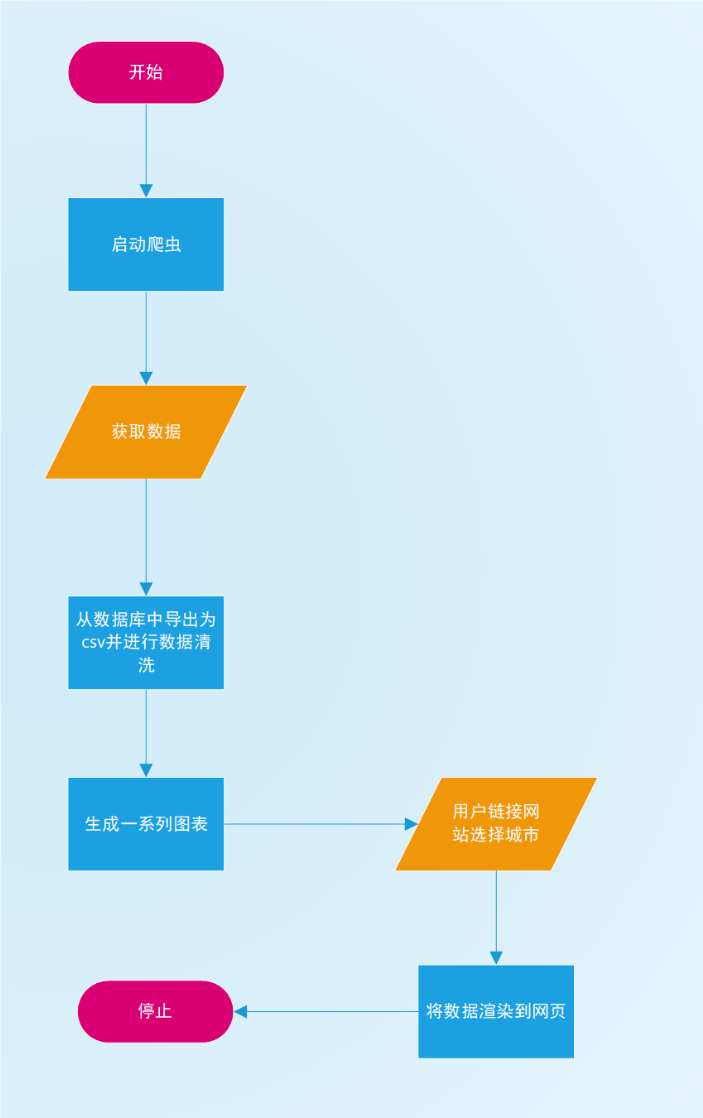


图1 系统流程图

**2.系统流程**

先运行weather\_scraper/spiders中的db\_init.py，这里会创建一个数据库：weather\_data.db

然后在weather\_scraper中运行脚本

.\.venv\Scripts\activate//激活虚拟环境

scrapy crawl accuweather//启动爬虫

这里会将网站上的信息导入数据库中

接下来运行data\_clean中的clean.py，数据库中的数据会被转化成csv文件并清洗

最后运行run.py,进入http://127.0.0.1:5000/ 就行了

**3.数据库的构建：**

共建立了四张表，分别是city、daily\_forecast、hourly\_forecast、today\_weather，外键为city，通过城市名联系四张表，后三张表分别记录未来几天的天气预报、未来几小时的预报以及当前天气状况。

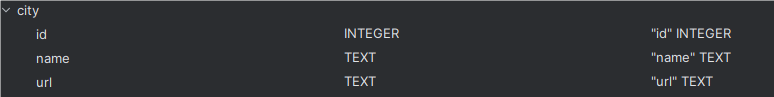


表1 City表

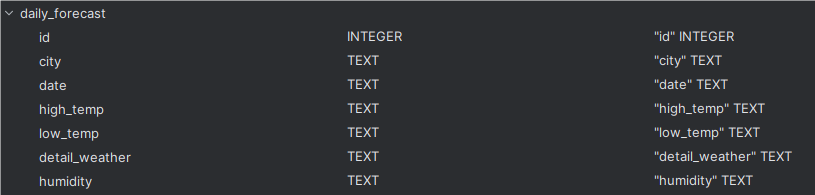


表2 daily\_forecast

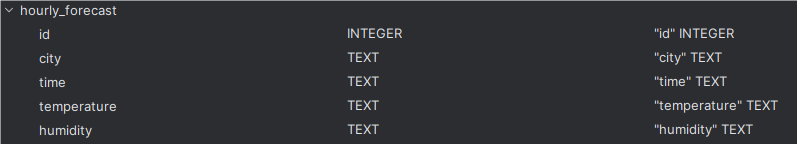


表3 hourly\_forecast

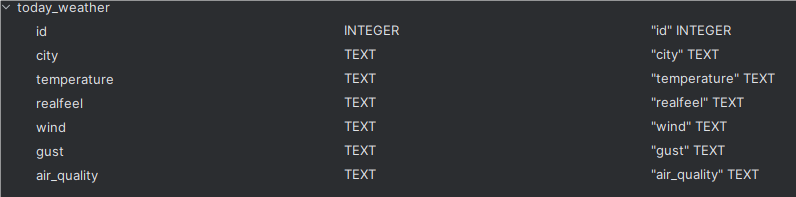
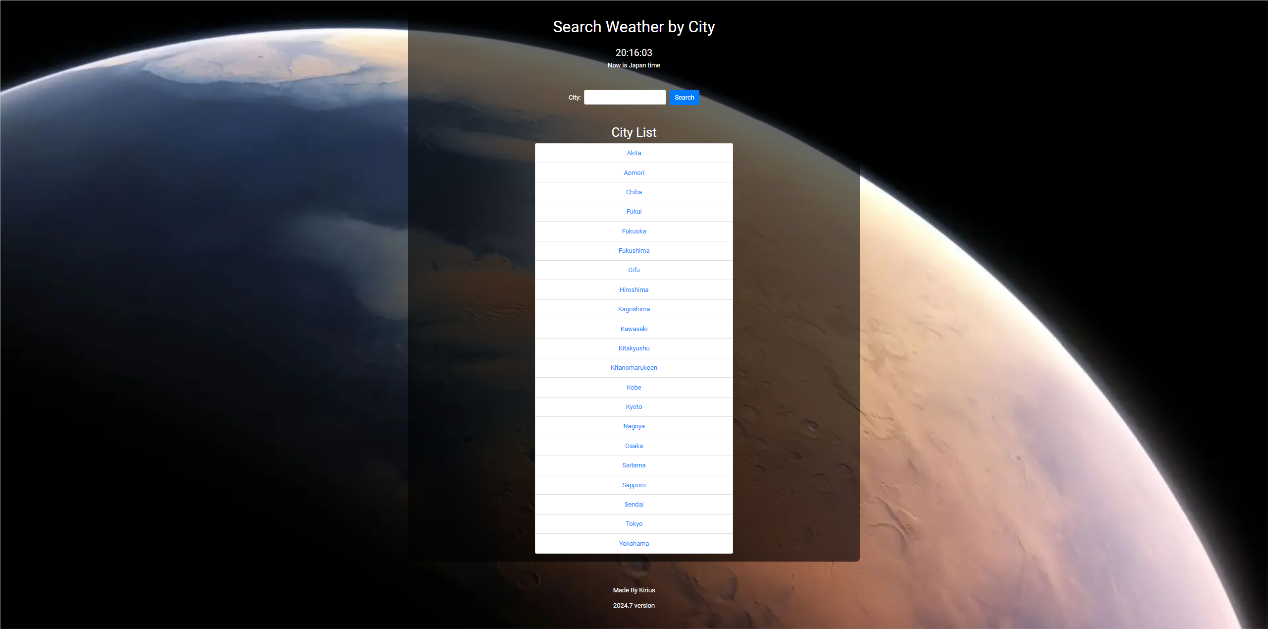


表4 today\_weather

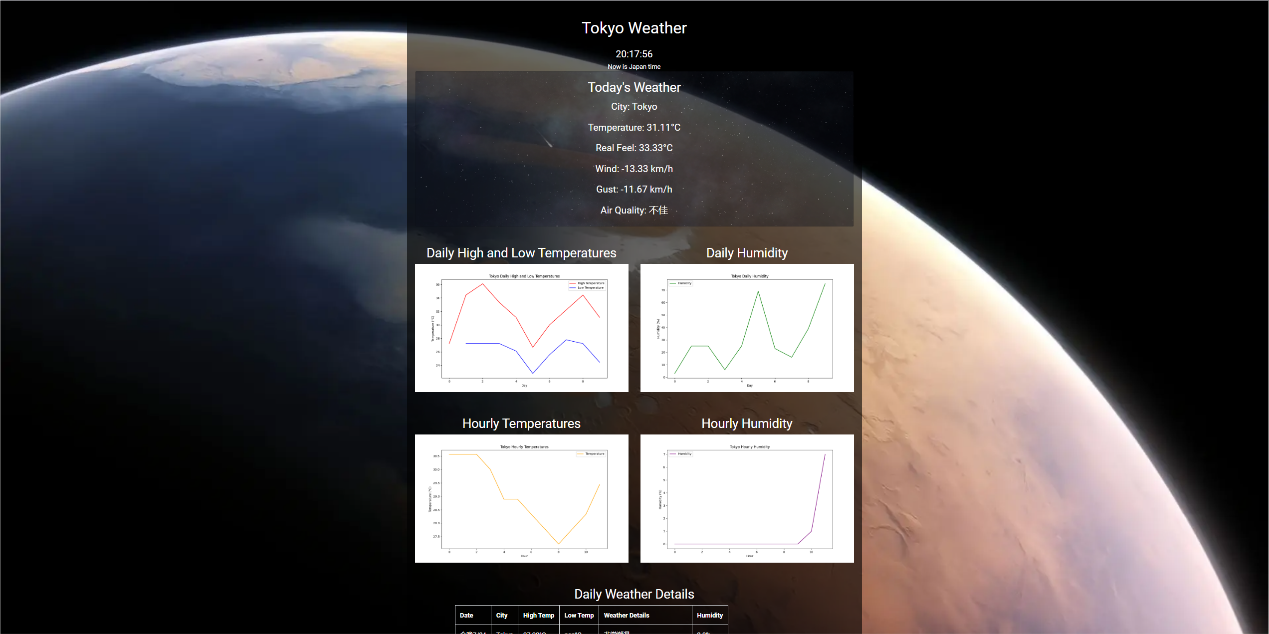
4.界面展示

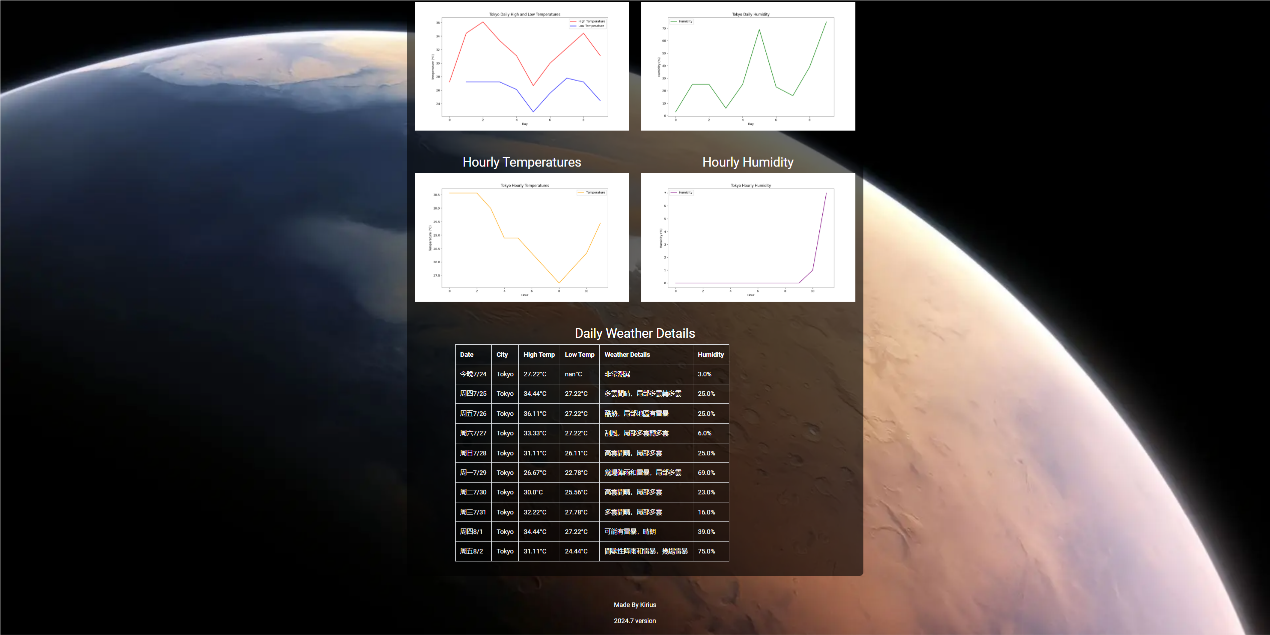


首页

搜索框可以搜索并进入城市

也有列表展示所有城市并且可以点击跳转





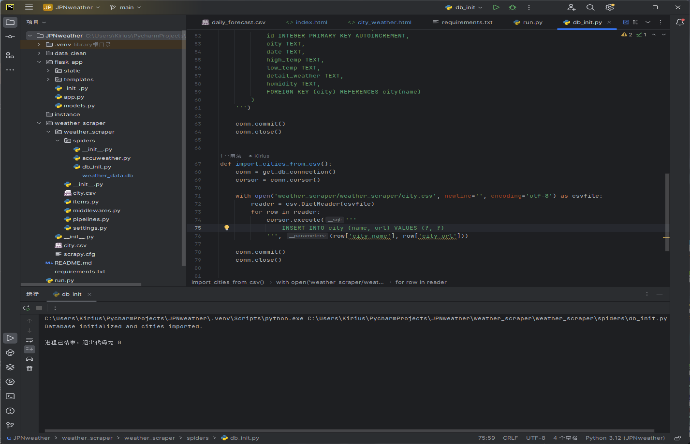
详情页

可以看到当日天气的详情以及未来几天以及几小时的预测图表，主要参数是温度以及湿度

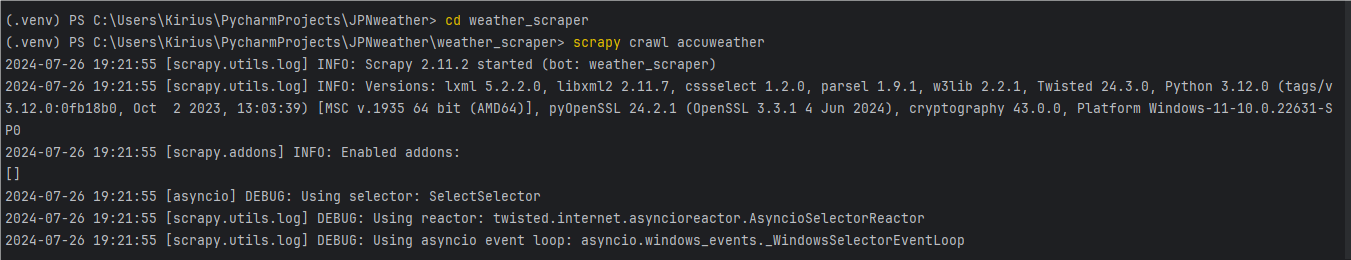
后有一张表格展示未来几天更加详细的预测信息

（三）、系统测试

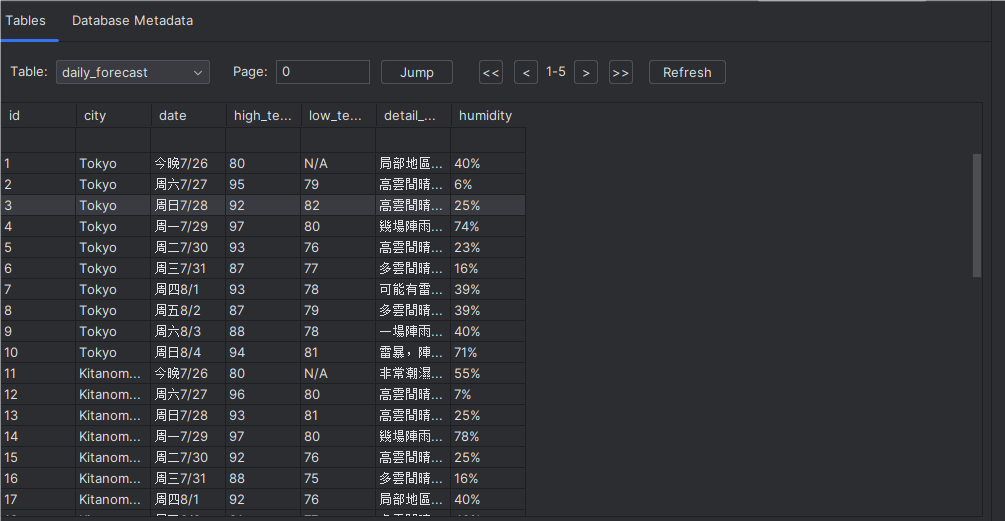
首先，启动db\_init初始化数据库



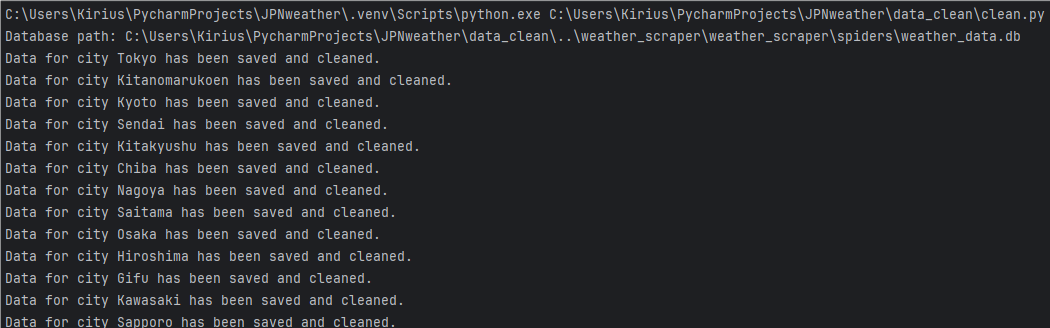
然后进入虚拟环境运行爬虫抓取数据



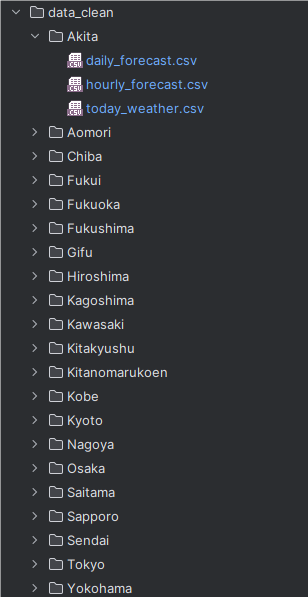
检查数据库，确认抓取成功



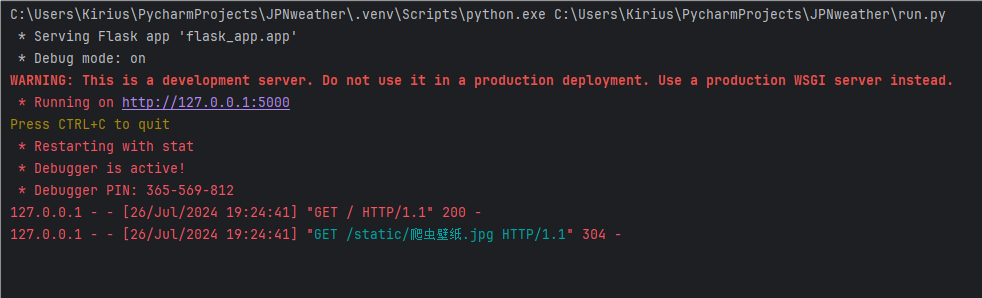
运行clean.py对数据库中的数据进行处理



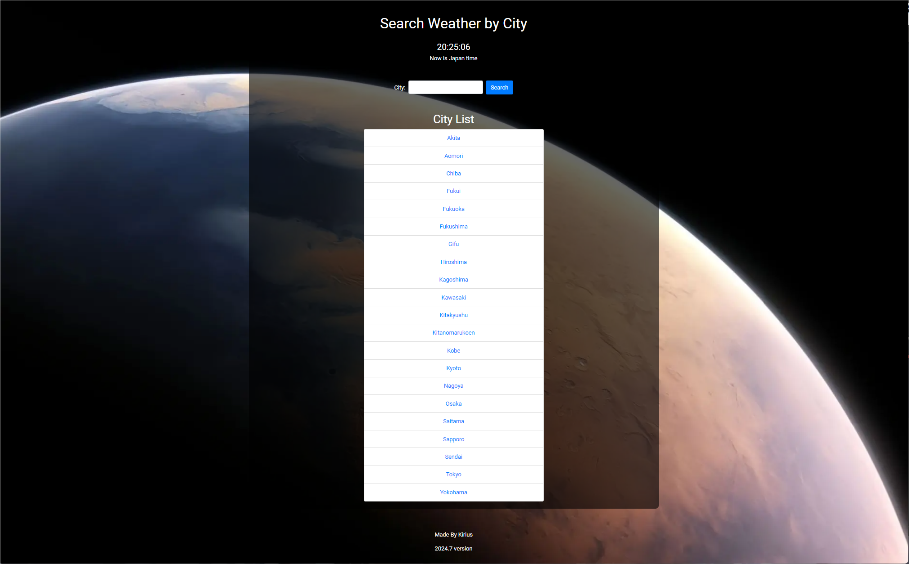
可以看到各城市的数据均处理完成



运行run.py启动网页

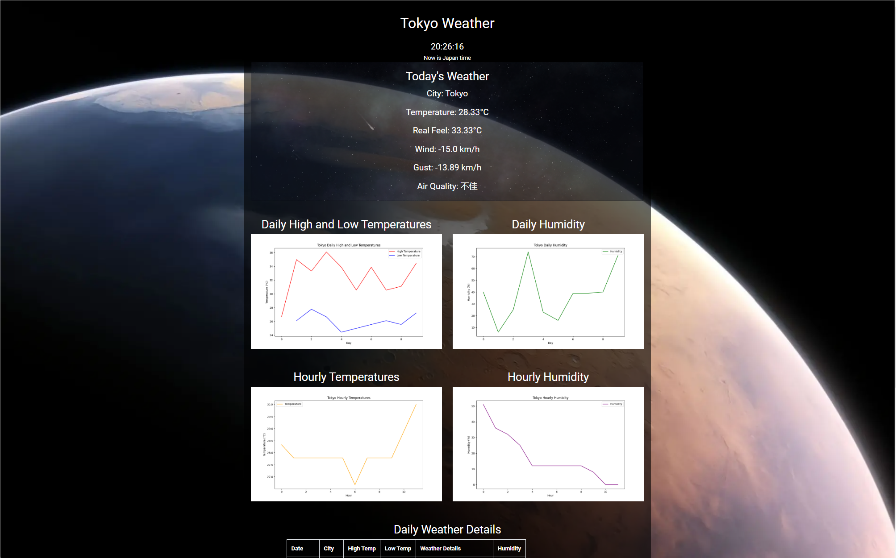


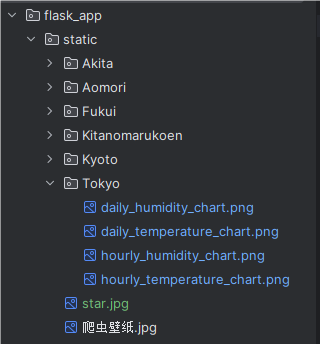
进入<http://127.0.0.1:5000>



输入Tokyo进行测试

相应的图表会在进入城市天气详情页时生成





如上，展示完毕

1. **课程总结**

在写这个项目的时候还是遇到了挺多困难的，因为不熟悉python语言，所以很多具体的工作都是借助ai进行编程的，在抓取数据的时候，由于我选择的是accuweather网站中日本的天气进行抓取的，chrome的开发者工具将html中的相关字段进行了转译，比如一个源网页显示的是华氏度，我们这边看到的是摄氏度，以及涉及到文字的一些数据，我这边显示的都是转换过的中文，导致前期爬取数据是字符一直匹配不上，后面在IDE中将html重新打印进行debug才解决了问题。

后续对数据进行处理以及建立flask网站就顺利很多，主要是简单设计了一下功能界面以及对网站进行了美化，包括加了两张背景图，以及对界面中的框图做了一个半透明的处理，看起来顺眼多了。

在测试的过程中我发现网页的运行并不稳定，但我在本地找不出相关的问题，只能理解为flask本身不稳定。

以下感想为ai生成

技术挑战与解决

在项目开发过程中，遇到了诸多技术挑战，如数据源网站的反爬机制、数据清洗中的异常值处理等。通过团队的共同努力和技术研讨，我们成功克服了这些困难。针对反爬机制，我们采用了合理的请求频率和模拟浏览器行为的方法；对于数据清洗中的异常值，我们使用了Pandas的高级数据处理功能，确保数据的准确性和完整性。

技术提升

本项目不仅提升了我们对Scrapy、Pandas和Matplotlib等工具的使用熟练度，还增强了我们在Web开发方面的能力。通过实际操作，我们深入理解了数据采集、处理、分析和展示的全流程。这些技术经验将为我们未来的工作和学习提供宝贵的参考。

作为本项目的开发者，我感到非常自豪。我们不仅完成了项目的预期目标，还在过程中积累了宝贵的经验。通过对天气数据的采集、处理、分析和展示，我们展示了扎实的技术能力和项目管理能力。

未来，我将继续提升自己的技术水平，不断学习新的知识和技能，力争在更多的项目中展现专业风范。我坚信，通过不断的努力和积累，我将能够应对更多更复杂的技术挑战，并在职业生涯中取得更加辉煌的成绩。