新冠疫情数据分析及可视化系统的设计与实现

Design and implementation of COVID-19 data analysis and visualization system

专　　业：　计算机科学与技术（信息处理）

姓　　名：　谢佩恒

指导教师姓名：　李伟

申请学位级别：　学士

论文提交日期：　2021 年 05 月 30 日

学位授予单位：　天津科技大学

摘　要

2020 年初，新型冠状病毒引发的肺炎疫情爆发，并在世界范围内造成了严重影响。为了易于理解的向公众展示目前的疫情状况，宣传防疫知识，提供对于未来疫情状况的预测。本文收集了国内外的疫情相关数据进行分析，对关于疫情的多个维度的复杂数据进行收集和处理工作，再使用可交互式动态图表表现出来。令公众可以通过动态的图表直观的看到国内外疫情的现状，做到对疫情“知己知彼”，增强对抗疫的信心。

本文选取了腾讯数据源，网易数据源提供的关于最新和以往疫情数据的 API 接口以及百度疫情大数据报告作为数据来源，使用 Python 作为后端开发语言，Flask 作为后端框架开发了一款新冠疫情数据分析及可视化系统。公众可以通过登录网页查看由动态图表所展示的疫情数据，国外部分国家的疫情发展趋势预测以及抗疫信息等内容。

在本篇论文当中，主要对此系统的开发意义，相关技术，系统设计、功能实现、测试用例以及本系统存在的不足之处以及对未来的开发和展望做了相关阐述。

关键词：疫情；可视化；线性回归；分析

ABSTRACT

In early 2020, the outbreak of pneumonia caused by New Coronavirus broke out and had a serious impact worldwide. In order to show the current epidemic situation to the public easily, publicize the epidemic prevention knowledge, and provide the forecast for the future epidemic situation. In this paper, we collected and analyzed the epidemic related data at home and abroad, collected and processed the multi-dimensional complex data about the epidemic, and then displayed them with interactive dynamic charts. So that the public can intuitively see the development trend of the epidemic situation at home and abroad through the dynamic chart, so as to "know yourself and know each other" of the epidemic situation, and enhance the confidence in the anti epidemic.

In this paper, we choose Tencent data source, NetEase data source, etc. to provide API interface for the latest and past epidemic data, as well as Baidu epidemic data report as data source, Python as the back-end development language, Flask as a back-end framework, and develop a COVID-19 data analysis and visualization system. The public can view the epidemic data displayed by the dynamic chart, the forecast of epidemic development trend in some foreign countries and the anti epidemic information through the login website.

In this paper, the main significance of the development of this system, related technologies, system design, function implementation, test cases and the shortcomings of the system, as well as the future development and prospects are described.

Key words: Epidemic situation; Visualization; Linear regression; analysis

目　　录

第一章 绪论

本文绪论部分首先阐述了本新冠疫情数据分析及可视化系统的研究背景以及研究意义，随后考察了此类系统在国内以及国外目前的发展状况，其次对说明了本系统应该具备的基本功能，最后对本论文的组织架构进行了说明。

第一节 研究背景与意义

自从新型冠状病毒引发的肺炎疫情在中国首次被报告发现，之后在全球范围内开始逐渐流行蔓延以来，与本次疫情相关的各个角度的数据，诸如各地区的每日新增病例，新增病例的发展趋势等重要信息就受到了公众的越来越密切的关注。一些大公司，组织以及机构等纷纷推出了自己的疫情数据信息发布平台。并且目前这些平台仍然在根据目前疫情的发展态势和公众对于疫情关注焦点的变化来积极更新维护自身的平台，目前，这些平台已经逐渐成为了公众了解和掌握疫情发展趋势的主要途径之一。

现如今，传统媒体正在逐渐衰落，根据 2020 年 4 月 28 日中国互联网络信息中心发布的第 45 次《中国互联网络发展状况统计报告》可知[1]，中国网民规模为 9.04 亿，互联网普及率高达 64.5%。通过这些数据可以知道，我国目前的网络普及率处于较高的水平。所以通过网络这种传播媒介来进行新冠疫情相关数据的发布可以起到更快，更广泛的传达信息的效果。

相对于传统的静态图表而言，动态图表可以传达更为精确，信息量更为丰富的内容。因此选择合适的可视化工具实现动态图表是传达疫情数据的较为有效的方法。基于目前较为成熟的可视化工具，本项目根据目前较为成熟的数据源，以易于理解和信息量丰富的可视化图表来展示目前国内外疫情的现状，防疫的进展和对未来发展的粗略预测。使得公众可以及时对全球疫情的发展趋势有更为直观，准确的了解。通过比较国内外疫情的发展趋势以及结合时事新闻中各个国家对于抗疫的政策措施，我们可以进一步深刻的理解我国践行“人民至上，生命至上”的伟大以及国外在资本主义制度的落后性的影响下导致防疫不力的现实状况。

第二节 国内外发展现状

在新冠疫情发生之后，疫情的发展状况和未来可能的趋势就成了公众关注的焦点。及时向公众传递疫情相关的精确数据，有助于增强公众的防疫信心，以及遏制疫情假消息和谣言的传播。自疫情开始蔓延以来，国家卫健委网站便开始每

日刊载全国疫情相关数据，并随着疫情发展状况的推移发布更多维度的内容信息，如在疫苗开始大规模接种之后开始发布疫苗接种剂次的信息[2]。为了使得这些疫情相关信息更容易被公众所理解，在疫情开始后从部分个人到组织机构都推出了疫情数据可视化平台。其中，国内外信息较为齐全，使用人数较多的有百度疫情大数据报告网站，腾讯健康微信小程序中的疫情版块等。这些由大型公司所推出的平台有数据更新及时，版面会随着目前公众对于疫情的关注焦点的改变而相应进行调整，确保始终传达公众最需要的信息的优点。因而现在已经成为了公众获取疫情信息的重要途径。为国内新冠疫情防控起到了巨大贡献。

由于国外疫情仍在肆虐[3]，所以国外也经历了从个人制作疫情图表到大型组织机构设立了较为完备的新冠疫情数据发布平台的情形。目前国外使用人数较多，较为著名的是约翰斯·霍普金斯大学的信息平台。该平台现已成为全球信息最为丰富，更新及时的新冠疫情信息平台[4]。除此之外，还有微软公司的新型冠状病毒肺炎追踪网站[5]等信息平台，这些平台都从多个维度提供了详尽的信息。除此之外还有世界卫生组织的新冠疫情网站[6]。

以上此类平台的建设和发展以及平台自身对自己的内容和图表设计的不断改进都证明了目前向公众传达准确，及时和容易被理解的疫情数据信息仍然是目前非常重要的工作。而通过对以上平台和网站的内容进行分析可知动态图表目前仍是展现疫情信息的较为有效的工具。

第三节 论文主要工作内容

本论文通过对爬虫技术的学习和研究，以及国内外在数据可视化方向上的研究和应用现状，再结合时下的疫情热点和大众可能关心的内容，设计了运行于云服务器上的新冠疫情数据分析及可视化系统。本文的主要工作如下。

研究以及分析了本新冠疫情数据分析及可视化系统的研究背景以及研究意义，调查了国内外的新冠疫情数据分析及可视化平台目前的现状以及未来可能的改进和发展的空间。通过总结自身的使用经验以及评估和分析公众对此类平台所需展示信息的需求，确定了本平台应具有的核心功能。

学习并掌握了多种爬虫和解析工具的使用方法，包括 requests，beautifulsoup 和 selenium 等功能强大的爬虫和解析工具。以及解析从数据接口返回的 JSON 文件的方法和手段，为新冠疫情相关数据收集模块的顺利编写奠定了坚实的基础。

通过分析本系统所应该具有的功能，大致划分了需要几个功能模块，确定了每个模块各需要具有何种功能，保障了整个开发工作可以保质保量的完成。

学习基础的云服务器运营维护技术，确保本新冠疫情数据分析及可视化系统在本地完成开发后可以顺利的迁移到云服务器上，以及在云服务器上部署成功后

确保系统的正常运行以及之后的运营维护和继续开发的顺利进行。

第四节 论文组织架构

本论文共由七个章节组成，在接下来的内容中将介绍每个章节的主要内容。

第一章：绪论，本章将对本论文所要实现的新冠疫情数据分析及可视化系统的研究背景和研究意义分别进行阐述，并较为全面的概括了目前国内外这类平台的发展状况以及今后的发展趋势，以及该系统完成后所应具备的功能，同时也对本论文的组织架构进行了详细的描述。

第二章：相关理论以及相关技术概述。本章节将简单的介绍开发该系统时使用的数个关键技术以及原因，包括使用 Python 语言作为爬虫的开发语言和本系统后端的开发语言的特点和优势，目前较为成熟的数据可视化工具以及特点等。并阐述实现新冠疫情数据分析及可视化系统时在哪里使用到了这些技术以及为什么要使用这些技术。

第三章：新冠疫情数据分析及可视化系统的需求分析和概要设计。需求分析这个概念包括功能性需求分析和非功能性需求分析，其中的功能性需求分析包括公众可以顺利通过网站查看由新冠肺炎疫情相关的数据制成的动态图表，以图片形式展示的疫情防护相关知识以及部分国家疫情数据的预测。非功能性需求包括信息系统中保证系统性能、系统可靠性以及可扩展性要求等方面的需求要素。

第四章：新冠疫情数据分析及可视化系统的总体设计。本章节结合了第三章的需求分析和概要设计部分的相关内容以及公众对本系统的要求的分析和总结，对新冠疫情数据分析及可视化系统进行具体功能模块的设计和重要组件的搭建。重点分析了本系统的数据库的设计的问题。

第五章：新冠疫情数据分析及可视化系统的详细设计与实现。本章节介绍了本系统有哪些主要的功能模块，以及这些模块的具体的实现方式。包括如下几个部分。如何从不同的网页中爬取数据，以及如何对爬虫加以改进以便节省时间，提高效率；如何爬取信息制作词云图片；如何使用公式拟合部分国家的疫情数据并预测之后可能的发展；如何对更新频繁的数据进行页面局部动态刷新等等。

第六章：新冠疫情数据分析及可视化系统的系统测试。本章对新冠疫情数据分析及可视化系统的各个功能模块进行了测试，在得到测试结果后根据测试结果对本系统加以改进，反复重复此步骤，最终使得本系统得到了基本的完善，并对本系统未来的改进和发展道路进行了一定的思考。

第七章：总结与展望。对围绕本论文进行的各项工作进行了总结，并对这一领域未来的发展进行了分析和预测。

第二章 相关理论及技术概述

本章主要介绍在开发新冠疫情数据分析及可视化系统的过程中使用到的相关技术。首先，本章首先介绍了 Python 这门语言，包括这门语言的发展历程以及这门语言的主要应用领域。其次介绍了爬虫相关技术的演进和结构。然后本章介绍了数据可视化领域的内容，重点介绍了 ECharts 这个数据可视化工具的基本使用方法以及使用它的理由。最后介绍了使用 Python 作为后端开发语言的相关情况，包括数个 Python 后端框架的简介和相互之间的对比。

第一节 Python 语言的特点

由于 Python 语言引入国内的时间点落后于 Java，C++等公众熟知的语言，普及程度相对目前的热门语言来说相对较低，因而在 Python 因为人工智能，爬虫等的大规模应用而变得广为人知之前，很多人并不十分清楚这门语言相较于其他语言的独特优势所在[7]。

与 JAVA 和 C++等历史较为悠久的编程语言相比，Python 的发展历史并不长，但有着较为丰富的背景。其诞生于二十世纪八十年代。作为众多计算机编程语言中的一种，Python 是一种解释型语言，但同时也具有一部分编译语言所具备的特性。

除了简洁，学习门槛低等优势以外，Python 还可以组合不同的模块，又被称为胶水语言。Python 还具有极为丰富的库，内容丰富且质量很高。因为存在这些库，使用 Python 进行编程来解决问题成为一件较为轻松的事情。这也是其学习难度低的体现[8]。

Python 语言编程的主要应用领域有如下：

1. Web 应用开发

Python 语言由于具有开源和跨平台等诸多特点而在 Web 开发的领域得到了较高的重视和较广泛的应用。涌现出了一批诸如 Flask 等的质量较高，使用人数多，适用范围广的 Web 框架。使用 Python 构建的主流网站在国外有 YouTube，在国内有豆瓣网等。

2. 网络爬虫

网络爬虫意即利用自动化运行的程序来收集和处理需要的网络资源。爬虫的类型总的来说大体可以分为两类，分别是主题爬虫和通用爬虫。主题爬虫是指使用者如果需要爬取一些类型和位置较为固定的数据的话，通过对网页进行分析，确定需要爬取的数据大致所在的位置，随后提取数据。而通用爬虫的典型代表就是一般所说的搜索引擎。

3. 数据分析

随着时代的发展，诞生了诸如 Matplotlib、NumPy 等功能强大，使用人数众多的 Python 库[9]。同时，也使得 Python 在数据分析领域以及计算领域得到了广泛的应用。使用目前成熟的计算类库可以轻易完成诸如矩阵运算等的操作，是数据分析的强有力工具。

第二节 基于 Python 的爬虫

现如今，爬虫技术已经成为了一项重要技术。在以前，需要从网站中获取某一类的全部信息时，只能靠人工手动保存，而爬虫的出现解决了这一问题。爬虫通过模拟人通过浏览器浏览和保存信息的行为，获取网站的源代码，或者从数据接口获取 JSON 等类型的数据以及其它各种网站资源，再对这些信息进行进一步的处理，最终得到需要的信息。

当前爬虫的种类从大体上可分为聚焦爬虫和通用爬虫。通用爬虫常见的是搜索引擎，无差别的收集数据，存储，提取关键字，构建索引库，给用户提供搜索链接。聚焦爬虫是指有针对性的编写特定领域数据的爬取程序，针对某些类别数据采集的爬虫，是面向主题的爬虫。

爬虫的架构大致可以分为三部分，首先是网址管理器，用来存放想要从中爬取信息的网站的网址。其次是网页内容下载器，用来保存获取到的网页的代码或者其它网站资源。最后是网页解析器，对保存好的网页代码进行解析，得到有价值的数据。

在内容下载器方面，常用的有 Requests 库。使用该库来模拟发起网络请求，获得网页的 HTML 代码。

在网页解析器方面，目前使用较为广泛的有 Beautiful Soup 等。Beautiful Soup 是一个可以从 HTML 或 XML 文件中提取数据的 Python 库。它能够通过 Python 内置的转换器或者使用者想要使用的转换器实现惯用的文档导航，查找，修改文档的方法。

第三节 数据可视化工具

目前，各行各业随着数据的不断增加，对数据的呈现和对数据的分析逐渐有了较为迫切的需求。因此，对于数据可视化工具的需求越发迫切。目前，虽然已经有了许多数据可视化工具，但是对于非计算机专业的数据分析人员或是界面设计人员来说，在较为快速的实现基于网络的交互式可视化方面仍然存在着不小的困难。目前已有的工具虽然已经提供了较以往更为方便的使用体验，允许用户将精力更多的放在交互式图表的设计上，但同时也要求用户必须对网页开发有一定

深度的了解。例如，D3.js 这款可视化工具要求用户在使用前应该对 HTML、CSS、SVG 和 DOM 有一定深度的了解[10]，而 Vega 这款可视化工具则要求用户在使用前需要掌握一套新的语法。这些要求都使得对计算机和前端了解不够深入的开发人员在面对交互式图表的开发时面临着较大的困难[11]。

ECharts 这款可视化工具则在一定程度上解决了以上问题。其包含丰富的图表案例和良好的图形互动界面，用户可以根据自己的需求对图表进行修改和创建[12]。使用者在使用 ECharts 时，不必对网络相关的编程有深入的了解，只需要在使用前花费一小段时间就能大体熟悉该工具提供的可视化组件。因此，目前已经有不少科研项目以及组织机构使用该款工具进行数据可视化工作，诸如在环境监测方向的水质检测领域[13]等。

总的来说，ECharts 相较于其余可视化工具，有三个主要优势。

一、易于使用。用户只需要专注于可视化设计，而不是花费过多的精力学习相关的语法。

二、丰富的内置交互。高效的数据探索和分析需要丰富的可配置交互。ECharts 设计并实现了针对每个图表类型的丰富的内置交互，最大限度的减少的了用户的定制需求。

三、高性能。通过引入流系统体系结构和增量渲染模式，ECharts 实现了高性能，即使是在处理数百万个数据点时也同样可以保持高性能。

在具体如何配置和使用上，ECharts 使用一个 JSON 格式文件来声明组件，样式，数据和交互。JSON 格式的主要优点在于存储，传输和执行非常安全，易于进行进一步的验证。并且 ECharts 官网针对每一种图表都提供了基础的代码模板并支持在线对代码进行改动并实时查看效果，使用者在官网上对图表代码稍加修改后复制进行使用，具体使用界面如图 2-1 所示。

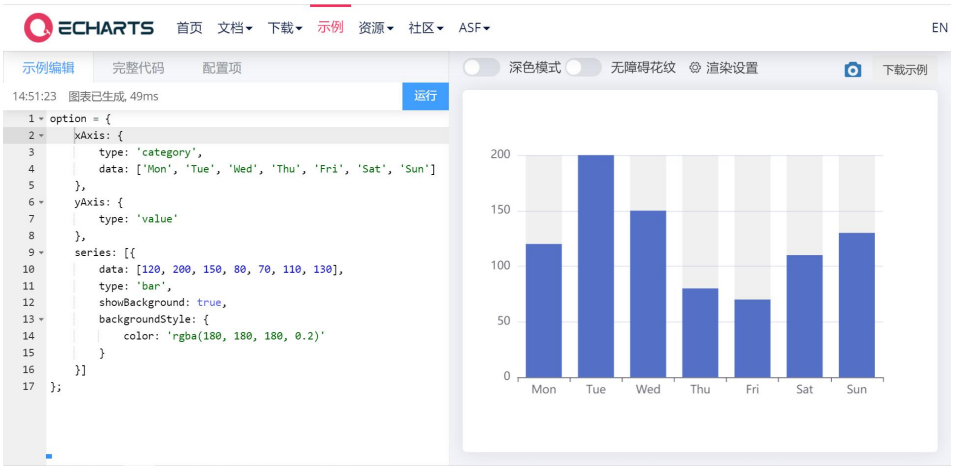

图 2-1 ECharts 操作界面

ECharts 使用“系列”来抽象一组从数据映射和编码的图形元素。类似的，

柱状图，折线图，饼状图等图表类型各可以抽象为一个系列。换句话说，系列是一种图表类型的实例。

ECharts 具有信息和资源都十分丰富的官网网站[14]，得益于其由百度开发的原因，其官网网站对于国内开发者非常友好，初学者可以轻松的通过官方网站从零开始学习 ECharts 的使用方法。通过官方网站提供的各类图形的丰富案例轻松制作自己的图片。在只需要简单的使用集中类型的图片的情况下，使用者甚至无需学习其语法和 API，只需要在官方网站提供的丰富的模板案例中选择最贴合自己使用要求的模板，参照官方文档加以修改即可。而齐全的官方 API 文档保证了在语法方面不必花费过多的时间进行学习。

第四节 基于 Python 的后端框架

基于 Python 强大的功能，其也可以用于 web 开发当中。Python 在 Web 开发应用方面具有 Flask、Django、CherryPy 与 Tornado 等五种主要框架。 在这五种框架之中，每一种 Web 开发框架都各自具有相较于其它框架较为独特的功能。

Flask 框架使用 Jinja2 作为模板引擎，使用体验非常便利。该框架本身为轻量级框架，如果需要复杂的功能可以自行安装扩展，比较适合于一些规模较小，访问量不是很大的 Web 网站开发。

Django 是 Python 下各种不同的 Web 框架的重量级选手中最有代表性的一位，适合大型 Web 网站的开发。许多成功的网站都基于 Django。Django 运所运用的框架模式是 MTV 模式，但本质上和 MVC（模型-视图-控制器）区别不大，也是为了各组件间保持松耦合关系，只是定义上有些许不同，Django 的 MTV 分别是指模型，模板和视图。

CherryPy 是一种比较简单的 Web 开发 框架，其突出特点是通过最少的步骤来实现 Python 代码与 Web server 的连接。CherryPy 还有非常灵活的插件功能、内置分析功能以及可以同时在多个 HTTP server 运行的优势[15]。

Tornado 框架是一种非阻塞 IO 形式的 Web server，所以运行速度特别快，是一种开源类型的、全栈式和异步网络库。

综合考量了本系统的规模以及各个框架的学习和使用成本，本系统最终选择了 Flask 框架进行使用

第三章 新冠疫情可视化平台的需求分析与概要设计

本章节首先从技术可行性和应用可行性两个方面对本系统进行了详细的分析。再从新冠疫情可视化系统的基本设计出发，结合软件工程的需求分析方法，对本系统进行了功能性需求分析以及非功能性需求分析。在这其中，首先对本系统进行了需求分析，明确了系统的主要功能，通过建立用例模型的方式对新冠疫情可视化平台的各个功能进行分析。除此之外，对系统的安全性、扩展性、可移植性、易用性等各个方面进行了非功能性需求分析。

第一节 新冠疫情可视化系统的可行性研究

系统的可行性分析是指以系统、全面的分析为主要方法，通过实际的业务场景着手，分析新冠疫情可视化系统的具体业务以及实际需要用到的，围绕影响新冠疫情可视化系统构建的多种因素，论证构建目前的新冠疫情可视化系统是否可行。

在技术方面，开发新冠疫情可视化系统所使用的平台主要是 pycharm 和 vscode 等，使用的技术也是 selenium，Flask 等的开源的技术和框架。尤其是 selenium 和 Flask 两者都十分轻便，好用且功能强大，都拥有活跃的社区以及大批的使用者，便于互相交流使用经验和解决实际开发中遇到的问题。

在应用可行性方面，本新冠疫情可视化系统运营在腾讯云的高性能轻量应用服务器上。采用业界主流方式进行部署，保证了用户的使用体验。

总的来说，无论是从技术可行性方面，还是应用可行性这两个方面来看，本新冠疫情可视化系统都有较为良好的可行性。

第二节 新冠疫情可视化系统的需求分析

软件的需求就是项目必须提供的功能和限制性条件。软件的需求大致分为功能性需求和非功能性需求。

软件需求分析是软件计划阶段的重要步骤。该步骤是分析系统在功能上需要实现何种功能，而不需要考虑如何去实现。软件需求分析的目的是把用户对于该系统提出的需求进行整理，确定该系统需要实现何种功能，完成何种任务。

本节将从功能性需求分析和非功能性需求分析这两种需求分析本新冠疫情可视化系统的需求。

第三节 功能性需求分析

功能性需求是本新冠疫情可视化平台必须要实现的功能，是直接为该系统的使用者提供的功能的需求。

就本系统而言，是想要设计并最终实现一个基于云服务的新冠疫情可视化平台。本系统的主要业务场景是使用可视化图表的方式向公众展示疫情数据，抗疫知识以及对部分国家疫情发展趋势的粗略预测。并且公众可以以静态图片的方式下载网页中的动态图片。

第四节 非功能性需求分析

平台的功能性需求分析主要分析了本新冠疫情可视化平台需要具备哪些功能的问题。而非功能性需求分析主要解决了本系统如何做得更好，更完善的问题。这也是一个非常重要的步骤。虽然非功能性分析看似距离用户较远，但是通过该分析可以提升平台的质量，提升用户的使用体验，提升系统长期运营的方便程度。下面将从系统的易用性、安全性、扩展性、可移植性等方面进行分析[21]。

一、易用性

本系统部署在云服务器上，在系统设计时特意为了可移植性以及为了方便在本地开发完成后迁移到云端以及迁移至云端后直接在云服务器上进行修改和开发而进行了优化设计，使得本系统的部署和后续运营都是极为方便的。除此之外，因为本系统使用 Python 语言编写以及在后端使用 Flask 框架，所以在出现问题之后在寻求帮助时是十分方便的。因此，本新冠疫情可视化系统具有较高的易用性。

二、安全性

本新冠疫情可视化系统仅具有数据的可视化展示功能，使得外界基本没有通过技术手段破坏系统运行的可能性。并且在云服务器端也采取了目前广泛采取的安全措施。这些举措都充分保证了本系统的安全性。

三、高可用性

本新冠疫情可视化平台运行在腾讯云的轻量应用服务器中，具备轻运维，开箱即用的特点。并且使用 supervisor 管理器守护进程，设置计划任务定时采集和更新数据。因此，本系统具有较高的可用性。

四、扩展性

目前本系统所提供的的服务较少，只能展示由爬虫定时爬取以及由数据接口定时获取的被可视化的信息。后续的继续开发和扩展充满了很多可能性。因为本网站显示的数据是从数据库直接取出，而被取出的数据是在获得数据后直接整理好的。因此具备极高的扩展性，可以部署更多的爬虫以及从更优质的数据接口定

时获取数据并加以整理后存入数据库，避免了每当收到请求就更新数据造成消耗性能的增长。因此本新冠疫情可视化系统具有较为良好的扩展性。

五、移植性

本新冠疫情可视化系统为了移植性进行了专门设计，在迁移到新的服务器后，只需要按照顺序依次启动数个文件并设置计划任务即可在短时间内在另一个服务器上运行本系统。因此，本系统具有较为良好的可移植性。

第四章 新冠疫情可视化系统的总体设计

在第三章已经完成的需求分析以及概要分析的基础上，本章节将详细讨论这些功能是如何实现的。

第一节 新冠疫情可视化系统的架构设计

本系统的功能相对来说较为单一，公众可以通过该系统查看基于疫情数据绘制的动态图表，也可以直接通过表格查看数据。通过图片查看基础的防疫知识以及查看基于部分国家往期数据所推算的未来的感染人数。

根据以上分析，可以得出本系统所应该具备的基本模块。首先是数据获取模块。本系统的全部数据来源于四个腾讯提供的数据接口，一个网易提供的数据接口和百度疫情大数据报告。从这几个来源所获取的数据需要经过处理后才能存储进数据库。所以其次就是数据处理模块，该系统所需的数据分散在从数据接口拿到的 JSON 文件中的不同地方，因此需要处理完成后再进行存储。同时还需要计算出预测结果，然后和历史数据一同存入数据库。最后是数据可视化模块。在包含数据的数组传输至网页后，已经编写好的数据可视化代码在读取数据后将直接生成动态图表。

第二节 数据库设计

根据第三章的新冠疫情可视化系统的需求分析的结果，确定了如何进行本系统的数据库的设计工作。这里最终选择了 SQLite 作为本系统的数据库。SQLite 是用 C 语言编写的开源嵌入式数据库引擎，包含在一个相对小的 C 库中，支持大多数的 SQL92 标准，并且可以在所有主要的操作系统上运行。相对于传统数据库，SQLite 具有更好的实时性、系统开销小、底层控制能力强。SQLite 能够高效地利用嵌入式系统的有限资源，提高数据的存取速度，增强系统的安全性[16]。以及使用 SQLite 可以为开发，测试以及最后的迁移至云服务器等等步骤提供极大的便利，因此最终决定采取 SQLite 作为数据库。在项目较为成熟之后再使用 Mysql 作为数据库。除此之外，本小节对数据库中所应存储的数据的数据类型进行了详尽的分析，完成了表的设计。

通过分析可知，本系统的数据库中应当存储经过处理的疫情数据，以便在网页被打开时动态图表所需数据直接由数据库被取出后能直接传送给网页而无需在后端服务器中经过处理后再发送以便节省时间。以及疫情相关数据本身就较为

碎片化，整体关联度不高。而且实际上也受制于数据接口的提供方式。除此之外，受制于不同数据接口的数据更新时间不一致，甚至更新内容有时也会有所缺少。因此在数据的存储上采取在前端页面上使用同一类数据的动态图片的数据就存放在一个数据库中的方法进行数据库的设计。

通过分析可知，为了满足疫情预测的功能，那么就需要存储数个国家的历史数据以便使用函数拟合后进行预测。因为数据接口提供的多个国家的疫情数据的开始日期各有不同，为了保险和方便起见就每个国家的数据各存储在一张表中。经过综合考虑，我们应该预测世界上经济总量较大的国家，疫情初期情况较为严重的国家的数据，再综合考虑我们可以从数据接口中得到哪些国家的数据，最后决定了这部分一共存储八个国家的数据，分别是巴西、德国、俄罗斯、法国、美国、西班牙、意大利、英国。每张表除了存储历史数据以外，为了保证网页请求时就可以直接传输包括历史数据和预测数据在内的全部数据，所以各个国家的表格应该同时包含这两种数据。在这其中，历史数据使用日期和感染人数相对应的形式，预测数据则使用距离有数据记录以来的天数对应预测的感染人数的形式，在表中的字段名为序号。以德国来举例，具体情况如表 4-1。

表 4-1 德国的预测表

字段名	数据类型	说明
time	TEXT	时间
confirm	INTEGER	累计确诊
number	INTEGER	序号
forecast	INTEGER	预测

现如今，疫苗的注射情况已经成为了公众所关心的问题。所以本系统中将展示疫苗相关的动态图片，因此将设置一张表用来存储这方面的信息。具体情况如表 4-2。

表 4-2 疫苗接种情况表

字段名	数据类型	说明
vaccination_country	INTEGER	国内累计接种
vaccination_country_new	INTEGER	国内新增接种
country_rate	INTEGER	国内接种率
vaccination_global	INTEGER	全球累计接种
vaccination_global_new	INTEGER	全球新增接种
global_rate	INTEGER	全球接种率

另一大类型的表将存储作为一个新冠疫情可视化系统所应该展示的基本的信息，比如最新一日的全球层面，全国层面以及各省市层面的诸如新增感染，累

计感染，累计治愈，累计死亡等信息，以记录省市层面当日最新数据的表为例，具体情况如表 4-3。

表 4-3 province 表

字段名	数据类型	说明
province	TEXT	省份名称
city	TEXT	城市名称
confirm_new	INTEGER	新增确诊
confirm_now	INTEGER	现有确诊
confirm	INTEGER	累计确诊
treat	INTEGER	累计治愈
dead	INTEGER	累计死亡

因为从数据源得到的历史数据往往截止前日。因而最新数据与历史数据中间相隔一天。为了存储和提取数据方便起见，设置一张表单独用来存储最新数据。具体情况如图 4-4 所示。

表 4-4 country_now 表

字段名	数据类型	空否
confirm	INTEGER	累计确诊
treat	INTEGER	累计治愈
dead	INTEGER	累计死亡
confirm_now	INTEGER	现有确诊
confirm_new	INTEGER	新增确诊
foreign_now	INTEGER	当日新增境外
foreign	INTEGER	累计境外

第五章 新冠疫情可视化系统的详细设计与实现

本章对基于 Python 的新冠疫情可视化系统的每个功能模块进行设计，并细致的阐述了各个功能模块的工作流程，具体开发步骤以及较为关键部分的代码。以及在开发时遇到的困难以及针对性的解决办法。

第一节 从数据接口获取每日信息

本文的疫情数据来源是腾讯新闻新型冠状病毒肺炎疫情实时追踪网站[17]和网易肺炎疫情实时动态播报网站[18]。

以爬取腾讯疫情新闻网站为例。如果考虑直接从页面进行信息爬取的话，可能只会拿到部分数据，而动态图表背后的累计数据等重要资源就难以拿到，因此在网站调试界面记录网站刚打开时的网络日志，可以发现网站上的数据是以 JSON 文件的格式传输至网站上，调试方法如下图 5-1 所示。

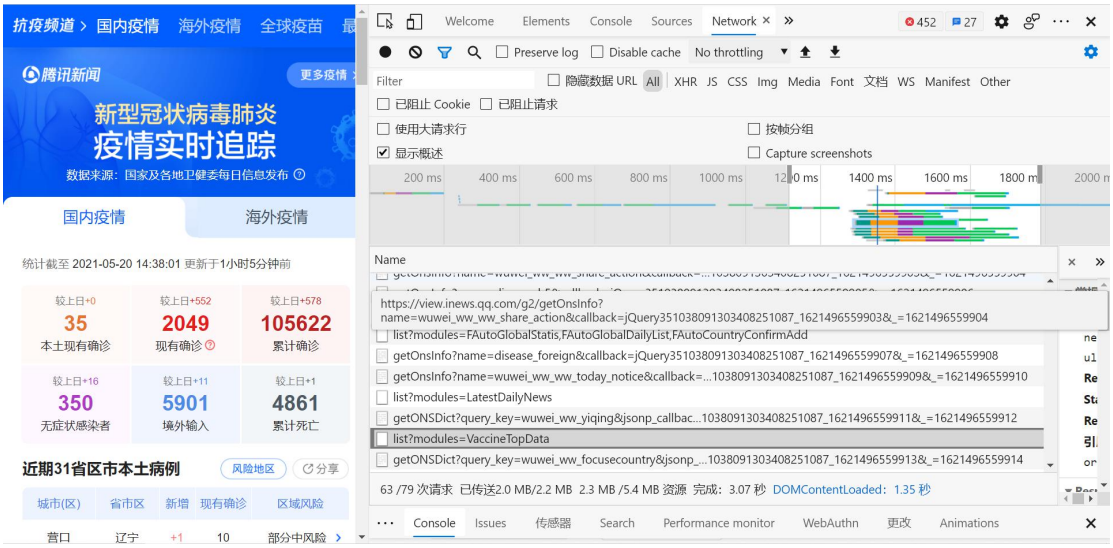

图 5-1 对网页进行调试

经过对本网站所加载文件的仔细梳理，一些包含疫情数据的 JSON 文件如下所示：

疫苗接种相关信息：list?modules=VaccineTopData

当日疫情最新数据：getOnsInfo?name=disease_h5

累计疫情数据：getOnsInfo?name=disease_other

按照国家名提取最近两个月的各国累计数据：list?countrys=%E5%8D%B0%E5%BA%A6

多个国家的长期累计数据：list?modules=FAutoCountryMerge

重复以上方法在网易肺炎疫情实时动态播报网站进行查找后，得到了数个 JSON 文件，对这些文件进行分析后可以得到其中的数据。下面将展示如何从 getOnsInfo?name=disease_h5 的接口中得到具体的数据。

首先使用 Python 字典中的 keys()方法返回字典中的所有的键，到得到的效果如下 dict_keys(['lastUpdateTime', 'chinaTotal', 'chinaAdd', 'isShowAdd', 'showAddSwitch', 'areaTree'])。在使用 type()判断这些键对应的值是数组还是字典，然后使用对应的方法查看对应的值[19]。通过这种方法掌握这个 JSON 文件所包含的内容。随后从中提取想要的数据。从该 JSON 文件中提取全国所有城市的疫情相关数据的方法如下所示。

```
def get_city_data(data):
    cmy = []
    for i in range(len(data['areaTree'][0]['children']) - 1):
        a1 = data['areaTree'][0]['children'][i]['name']
        for j in range(len(data['areaTree'][0]['children'][i]['children']) - 1):
            a2 = data['areaTree'][0]['children'][i]['children'][j]['name']
            b2 = data['areaTree'][0]['children'][i]['children'][j]['today']['confirm']
            c2 = data['areaTree'][0]['children'][i]['children'][j]['total']['nowConfirm']
            d2 = data['areaTree'][0]['children'][i]['children'][j]['total']['confirm']
            e2 = data['areaTree'][0]['children'][i]['children'][j]['total']['heal']
            f2 = data['areaTree'][0]['children'][i]['children'][j]['total']['dead']
            cmy.append([a1, a2, b2, c2, d2, e2, f2])
    return cmy
```

在从 JSON 文件中解析到数据后，不能直接存入数据库，首先而是要根据设计动态图表时的要求在对数据进行处理后再进行存储，以便加快网站被访问时从数据库中提取数据的效率。这部分详细内容在之后的章节会进行阐述。

第二节 从百度疫情大数据报告爬取词云所需数据

词云已经日渐成为了一种突出重点的常用方式，在各种媒体中都被广泛使用。在互联网上也有很多可以快速简便制作词云的网站。由于我们的系统中的词云图需要定时刷新，因此需要用到 Python 的 wordcloud 库来进行词云的制作。

在制作词云之前，首先需要从百度疫情大数据报告网站[20]爬取相关数据，所需爬取的数据如图 5-2 所示。

图 5-2 所需爬取的热搜

由于热搜列表的大部分内容处于折叠状态，直接使用爬虫爬取只能得到一部分热搜词，因此使用 Selenium 工具进行爬取。Selenium 是一个 WEB 自动化工具。可以用于自动化测试和爬虫工作。

网络爬虫，又称网络蜘蛛，是一种可以从网页上根据设置自动下载数据的代码程序。爬虫程序可以自动解析和传输网页数据，并将这些数据进行分析下载。Python 具有种类丰富的官方和非官方制作的网络爬虫库。其中 selenium 库是一个以 Google 浏览器为驱动的网页自动化测试工具，它能完全模拟用户手动操作网页的过程，并展现在浏览器界面上，操作十分方便[21]。

首先需要下载版本正确的 chormedriver 来驱动谷歌浏览器。并且 chormedriver 需要和谷歌浏览器的版本号应该按照一定的对应关系相对应。因此下载 window 和 linux 版本的驱动便于本地使用和迁移至云服务器后使用。本系统所使用的的 chromedriver 版本号以及谷歌浏览器的版本号分别为 91.0.4472.19 与 90.0.4430.212。为了便于使用 chromedriver 避免因为路径等问题无法启动，所以将 chromedriver 放置于本项目的项目目录中即可直接使用。

因为本项目要最终迁移至云服务器，所以应该使得爬虫程序在运行中以静默模式启动浏览器。

在具体如何爬取上，使用 css 选择器定位页面中信息流的按钮位置模拟手动展开更多信息。随后用 Xpath 定位文档中想要获取的信息的位置，随后对 Xpath 进行调整，使得能够对文档中对应的元素和属性进行遍历，进而爬取全部的标题。爬虫的具体设置代码如下所示。

```
def get_baidu():
    url = 'https://voice.baidu.com/act/newpneumonia/newpneumonia/?from=osari_pc_1'

    option = webdriver.ChromeOptions()
    option.add_argument('headless')
    option.add_argument('no-sandbox')
    option.add_argument('disable-dev-shm-usage')
    browser = webdriver.Chrome(executable_path='/bs/chromedriver',chrome_options=option)

    browser.get(url)
    but = browser.find_element_by_css_selector('#ptab-1 > div.Virus_1-1-303_2SKAfr > div.Common_1-1-303_3lDRV2')
    but.click()
    time.sleep(3)
    c = browser.find_elements_by_xpath('//*[@id="ptab-1"]/div[3]/div/div[2]/a/div')
    cmy = []
    for i in c:
        cmy.append(i.text)
    browser.quit()
    return cmy
```

在以数组形式获得全部热搜数据后，使用 Python 的 jieba 库进行分词工作。Jieba 是目前针对中文的分词效果较为优秀的 Python 中文分词组件[22]。主要支持三种分词模式，包括精确模式，搜索引擎模式和全模式。精确模式适合进行文本分析，将句子精确的切开，只输出最大概率组合。搜索引擎模式适合搜索引擎分词，为了提高召回率，主要在精确模式的基础上对长词再次加工。全模式是指句子存在冗余，将句子中所有可组词的词语的都扫描处理。本系统使用可以返回可迭代的列表类型函数进行处理。三种模式的具体信息如表 5-1 所示。

表 5-1 jieba 分词的三种模式

模式	函数
精确模式	lcut(s)：返回列表类型
	cut(s)：返回可迭代的列表类型
搜索引擎模式	lcut_for_search(s)：返回列表类型
	cut_for_search(s)：返回分词结果
全模式	lcut(s，cut_all = True)：返回列表类型
	cut(s，cut_all = True)：返回有概率的单词

在得到了包含由全部热搜标题切割而成的词语组成的数组后，即可开始制作词云。词云是以词语为最基本的单位，是一种可以更加直观和易于理解的展示文本的方式之一。wordcloud 库是 Python 中较为知名且易于使用的第三方用于制作词云的库。规定特定文本词在文本数据源中出现的次数绘制词云的形状、尺寸和颜色。本文为了使用户能够理解目前抗疫热点的关注点在哪里，利用越高频越突出的视觉效果，将分析出来的高频词汇用词云图进行展示。在数组中出现频率越高的词语，在词云图片中其尺寸就越大，反之就越小。

随后使用 wordcloud 对生成词云。在此之前需要准备词云的背景图片。根据

wordcloud 的使用规则，图片中的黑色部分是生成词云的位置。因此在静态资源文件夹中放入正方形黑色图片。由于词云图片需要每隔一段时间就根据新采集到的信息进行重新制作，因此设置在新的图片制作出来后删除固定路径下的旧图片随后将新图片保存在相同的路径下。除此之外，必须要注意的是需要指定词云图片显示文字的字体。否则文字会出现乱码。

在网页被打开后，因为词云部分使用了网页局部动态加载技术，因此使用传统的在图片标签中写出静态资源相对路径的形式来加载图片无法满足这一要求。因此使用 Base64 库对图片信息进行处理，将图片信息通过 Flask 发送至浏览器并显示图片。

第三节 搭建 Flask 框架

Flask 框架是一个轻量级 Python 框架。Flask 本身仅具有基本功能，但可以根据需求自行下载需要的模块。因此非常适合用于本系统的开发工作。

本项目目录即包含了支持本项目运行的全部文件，包括爬虫文件，数据库文件，路由，网页以及静态资源。本框架的项目目录如下图 5-3 所示。

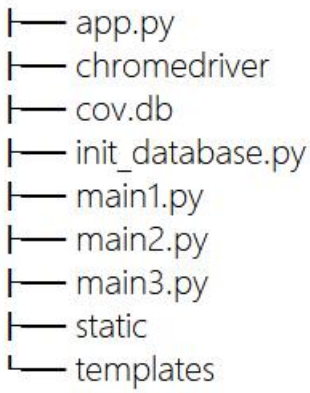

图 5-3 项目目录

在这其中 app.py 文件为 Flask 框架的启动文件，内容包括路由部分，在启动 Flask 框架时直接运行该文件即可。init_database.py 文件为数据库建立文件，运行后创建数据库并创建所有用到的表。main1.py，main2.py 以及 main3.py 为爬虫文件，通过设置定时启动来及时爬取数据。

项目目录中有两个特殊的文件夹分别用来存放网页以及静态文件，这两个文件夹分别是 templates 和 static。static 文件中存储了网页所需的 CSS 文件，词云所需的字体文件以及黑色的背景图。以及 ECharts 和 jQuery 以来的 JavaScript 文件。以上两个文件夹的目录如下图 5-4 所示。

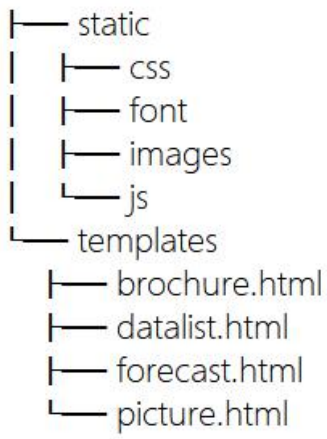

图 5-4 static 文件夹和 templates 文件夹的结构

在 app.py 文件中的路由的设置上采取以下设置方法，本系统分别设置四个网页来展示四个不同维度的内容。首先是/picture 页面展示动态图表，其次是/datalist 页面以表格形式展示疫情数据，然后是/brochure 页面是抗疫信息的展示页面。最后是/forecast 页面用来展示部分国家的疫情的预测数据。为了及时的提供最新数据，本系统具有可以在不重新加载网页的情况下对网页的某部分进行更新的功能。因为数据接口的数据不会实时更新而是在一段时间内进行集中更新。因此对于动态图表页面的的所有数据都使用本功能是没有必要的。因此设置/ajax1 对本网页的开头部分的以数字形式显示的内容进行更新。设置/ajax2 对本网页的词云部分进行更新。整个系统的路由设置如图 5-5 所示。

```
@app.route('/ajax1')
def ajax1():…

@app.route('/ajax2')
def ajax2():…

@app.route('/forecast')
def forecast():…

@app.route('/brochure')
def brochure():…

@app.route('/datalist')
def datalist():…

@app.route('/picture')
def picture():…
```

图 5-5 路由设置

第四节 可视化图表以及网页设计

本系统使用 ECharts 作为制作动态可视化图表的工具。目前已经有了形形色色的新冠疫情数据可视化平台，并且这些平台都提供了丰富的信息。所以本系统在图表的展现上力图展现差异化的信息。

首先是关于疫情最基本的信息，包括新增确诊，现有确诊，累计确诊，累计治愈，累计死亡五种数据。目前当日境外输入病例数在当日新增病例数中所占的比例较大，所以新增境外输入病例数也是一个重要的数据。

除此之外，目前的抗疫时事新闻中，疫苗的接种情况是公众非常关心的信息。所以通过展示国内累计接种、国内新增接种、国内接种率这三个数据可以直观的展现我国疫苗的接种态势。同时全球累计接种、全球新增接种、全球接种率也是公众所关心的重要数据，因为这些数据的增长展现了全球疫苗接种情况的变化。而只有接种率超过一定数值后各国重新开放边境才能成为可供考虑的选项。除了展示数据之外，在数字的颜色的选择上也应有一定的考量。疫苗接种部分的数据设定为有层次感的绿色，象征着早日战胜疫情的希望。累计死亡部分的数据使用黑色以示严肃。其余数据则使用暖色调。该部分的具体表现如图 5-6 所示。

图 5-6 数据陈列部分

在该部分之后则是可视化图表部分。首先是以中国地图的方式来展示目前全国范围内的累计感染情况，通过不同层次的颜色以及触碰后显示该省份看数据可以直观的看到数据，如图 5-7 所示。

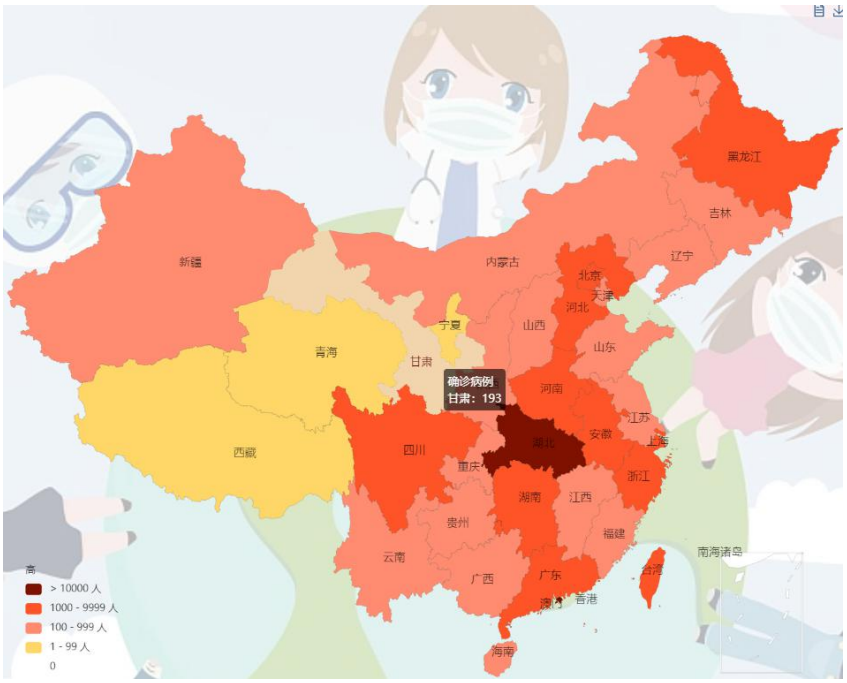

图 5-7 地图形式的全国累计确诊图

其次是以矩形树图的形式展示的各省区市和特别行政区以及其下市县一级的累计确诊情况。通过矩形树图可以清楚的看到各地区的累计感染数占总数的占比以及具体人数。在点击之后可以看到该省区市内的行政划分的具体情况。具体如图 5-8 所示。

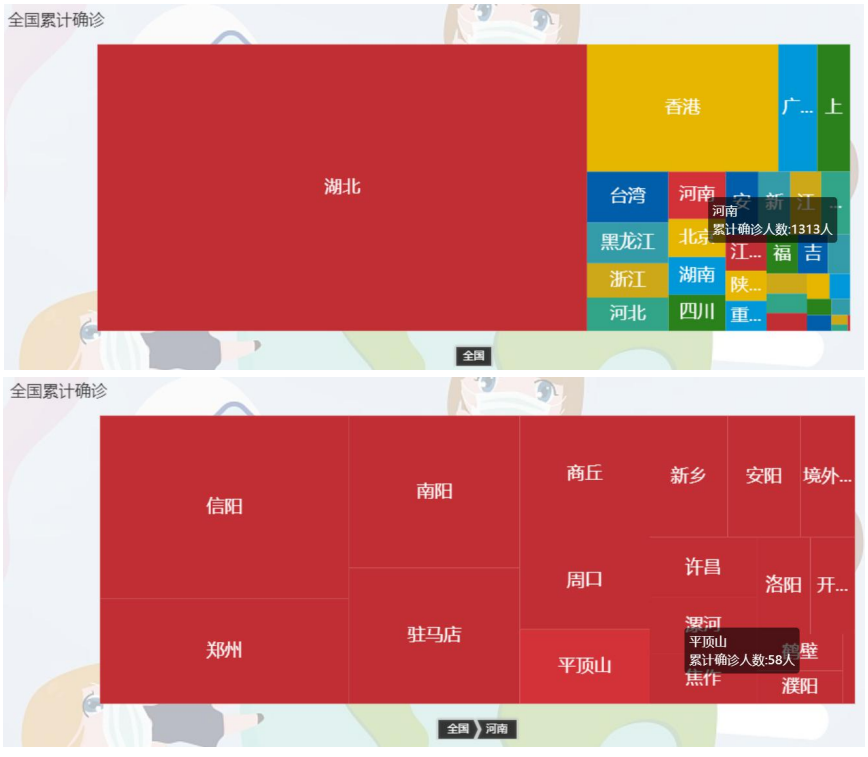

图 5-8 矩形树图演示

矩形树图的核心代码如下所示。

```
<div id="main1" style="width: 1050px;height:400px;margin: 50px"></div>
<script type="text/javascript">
    var myChart = echarts.init(document.getElementById('main1'), 'shine');
    option = {
        title: {
            text: '全国累计确诊',
        },
        tooltip: {
            trigger: 'item',
            formatter: '{b}</br>累计确诊人数:{c}人'
        },
        series: [{
            name: '全国',
            type: 'treemap',
            leafDepth: 1,
            drillDownIcon: '',
            roam: 'false',
            label: {
                fontSize: 20
            },
            data: [
                {% for i in p1 %}
                    {
                        name: '{{ i[0] }}',            // First tree
                        value: {{ i[1] }},
                        children: [
                            {% for j in i[2] %}
                                {
                                    name: '{{ j[0] }}',       // First leaf of first tree
                                    value: {{ j[1] }}
                                },
                            {% endfor %}
                        ]
                    },
                {% endfor %}
            ]
        }]
    };
    myChart.setOption(option);
</script>
```

在传统的图表部分，对于存在变化趋势的数据使用折线图进行展示。全国新增感染，全国累计数据，新增境外输入，累计境外输入使用折线图进行展示。下面以全国累计数据图为例，如图 5-9 为例。

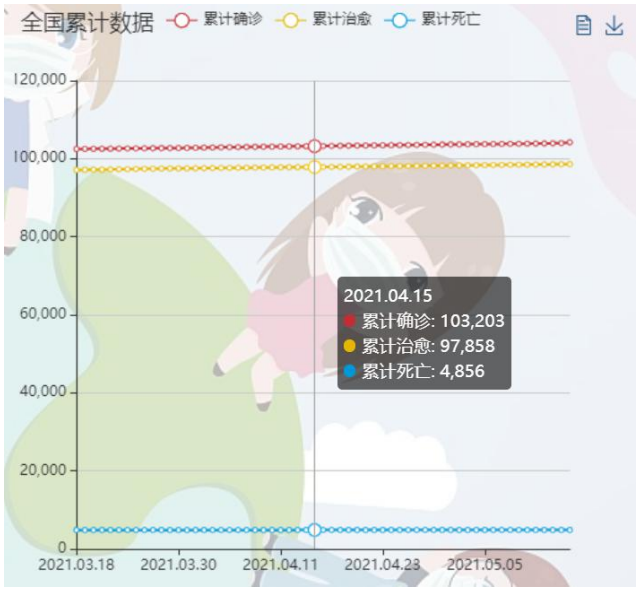

图 5-9 全国累计数据图

因为扇形图可以清晰的表现部分与部分，整体与整体之间的关系，所以为了清晰的展现国内疫苗的接种情况以及国内接种情况在全球来看处于何种水平，使用饼状图来展示国内新增接种占国外新增接种的百分比以及国内累计接种占国外累计接种的百分比。如图 5-10 所示。

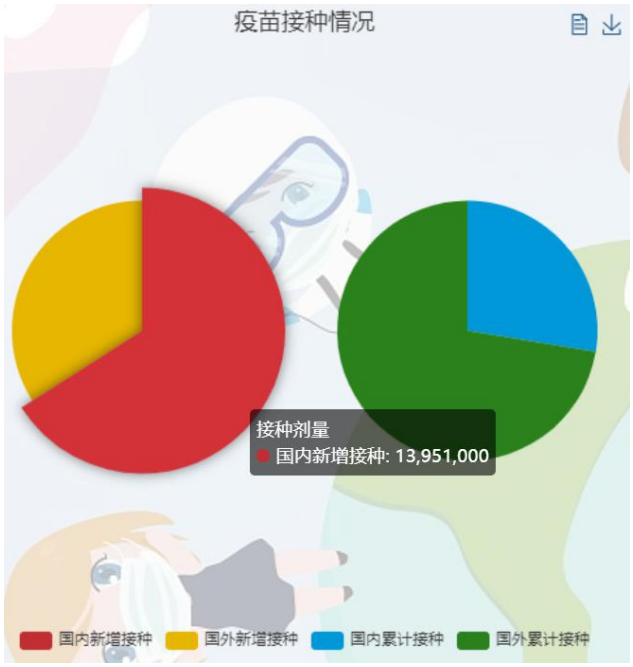

图 5-10 疫情接种情况图

为了清晰的展示目前公众舆论对于疫情的关注点在哪里，使用词云图片是一种很好的方式。相对于其他形式来说词云更能吸引公众的注意力。本文使用 wordcloud 这个 Python 库来生成词云，词云的效果如图 5-11 所示。

图 5-11 词云图

词云图的核心代码如下所示。

```
con = sqlite3.connect('cov.db')
cur = con.cursor()
sql = '''
    select 热搜词条 from baidu
'''
data = cur.execute(sql)
cmy = []
for i in data:
    cmy.append(i[0])
cur.close()
con.close()
text = ''
for i in cmy:
    text = text + i

# 通过jieba将长语句拆开变成词语
cut = jieba.cut(text)
string = ' '.join(cut)

# 将背景图片变成二进制数组
img = Image.open(r'/bs/static/images/词云背景图.png')
img_array = np.array(img)  # 将图片转换为数组

# 使用词云库进行处理，生成图片
wc = WordCloud(
    background_color=None,
    mask=img_array,
    font_path=r'/bs/static/font/msyh.ttc',
    mode='RGBA',
)
wc.generate_from_text(string)
```

为了能更好的展示中国国内有确诊病例的地区以及确诊病例数，使用南丁格尔玫瑰图来展现这项数据效果较好。南丁格尔玫瑰图是一种圆形的直方图，由弗罗伦斯·南丁格尔发明，最初用于表示军医院的季节性死亡率。图片效果如图 5-12 所示。

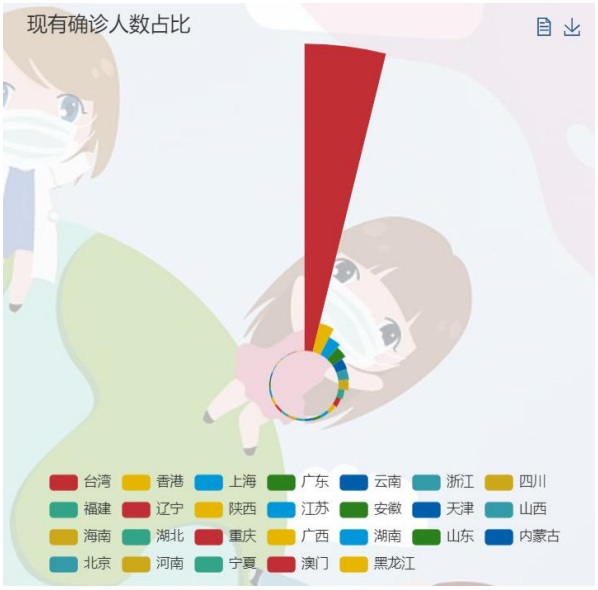

图 5-12 南丁格尔玫瑰图

其余数据则使用柱状图来表示，包括全球新增确诊 top10，全球累计确诊 top10 的国家和地区。

由于疫情相关的六个基本数据其来源数据接口的更新较为频繁，以及词云要展示目前的热搜词，所以更新应当及时。而对于本页的其余部分数据和图表的来源数据来说，因为诸如历史数据等的内容在一天之内并不频繁更新，所以设置页面局部动态刷新的必要性不大。因此对于应当设置局部动态刷新的部分使用 jQuery 进行局部动态刷新[23]。

使用图表有助于公众对数据有直观的印象，但也导致部分维度的数据不能清晰的展示出来。为了展示详细的国内疫情的最新数据和历史数据，在第二个页面使用表格的形式来展示国内疫情的详细数据。并在其中使用下拉表格的方式展示个省市区直辖市辖区内的县市的疫情数据。具体如图 5-13 所示。

地区	新增确诊	现有确诊	累计确诊	累计治愈	累计死亡	详细信息
台湾	542	882	2017	1123	12	暂缺
浙江	8	31	1355	1323	1	展开/收起
辽宁	7	14	422	406	2	展开/收起
上海	6	56	2037	1974	7	展开/收起

地区	新增确诊	现有确诊	累计确诊	累计治愈	累计死亡	详细信息
台湾	542	882	2017	1123	12	暂缺
浙江	8	31	1355	1323	1	展开/收起
杭州	0	0	181	181	0	
境外输入	8	31	136	105	0	
衢州	0	0	14	14	0	

图 5-13 国内详细疫情数据表格

除了展示数据以外，本系统还将展示防疫要点等信息。因此设置轮播图来展示抗疫信息。所展示图片来源于人民日报微信公众号。之所以选取这些图片是因为这些图片的设计简明大方，切中要点，具有良好的宣传效果。为了方便开发，使用了 jQuery 这个成熟的 JavaScript 库。

第五节 根据公式预测部分国家的疫情发展趋势

本部分使用 Logistic 模型拟合疫情的历史数据，并给出对于部分国家未来一段时间内的预测。

逻辑斯蒂方程(Logistic function)由比利时数学家兼生物学家皮埃尔·弗朗索瓦·韦吕勒(Pierre Francois Verhulst)在研究人口增长模型时提出，是对马尔萨斯人口模型(Malthus, 1798)的改进。目前该方程运用于医学和工程等多个领域，诸如风力发电机功率等[24]。

马尔萨斯人口模型假定人口增长率保持不变[25]。在公式中，人口增长率用 r 表示，人口数用 P 表示。是时间 t 的函数。通过对微分方程进行求解，可以得到人口数量随着时间进行变化的函数。其中 P0 是初期的人口数量。公式如下图 5-14 所示。

$$\frac{\mathrm{d}P}{\mathrm{d}t} = rP$$

$$P(t) = P_0 e^{rt}$$

图 5-14 马尔萨斯人口模型

根据该公式可以知道，人口增长的类型为指数增长，即一般来说的 J 型曲线。然而，在现实生活中受到自然环境的限制，以及疫病等的影响，人口增长率不可能长期保持一个数字不变[26]。

Logistic 模型则是在马尔萨斯人口模型的基础上进行了一定程度的改进[27]。在该模型中，人口增长率不是 r，而是$r(1-\frac{P}{K})$。在这个改进版公式中，K 可以被理解为目前该环境允许的人口数量的最大值。在这个条件下，当人口数量 P 越来越接近最大人口数量 K 时，人口增长率就越来越低，意即人口增长率随着人口数量的持续不断的增加而线性的逐渐减少。通过对经过修改的微分方程进行求解，可以得到人口数量随着时间进行变化的函数。其中 P0 为初始人口数量。公式如下图 5-15 所示。

$$\frac{\mathrm{d}P}{\mathrm{d}t}=r(1-\frac{P}{K})P$$

$$P(t)=\frac{K}{1+(\frac{K}{P_0}-1)e^{-rt}}$$

图 5-15 Logistic 模型

由图 5-16 可以看出这两种模型之间的差异。

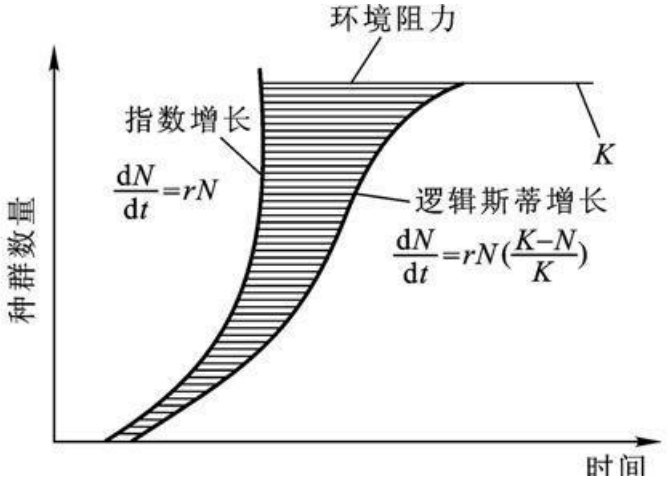

图 5-16 两种模型之间的差异

将 Logistic 模型与马尔萨斯人口模型相比较，可以发现 Logistic 模型更加切合实际生活。因此这种模型经常被用于描述生物种群和传染病感染人数的增长以及商品的销售等领域。

在该公式的应用中，还存在一个重要问题，那就是对初始感染人数 P0 的估计。因为在疫情初期部分西方国家的感染人数报告存在严重的低估。根据爬取的数据显示美国在 2020 年 1 月 8 日只显示累计有 5 人被感染，但实际数据应该远高于此数据。所以首先大致估计出在有数据统计的最早的日期的已被感染人数代入公式中进行运算，查看拟合出的图像与实际情况之间的差异。然后不断进行调整，最终确定应该对应国家的最早有记录的日期的感染人数。在程序中使用数组 c_p0 来将初始感染人数这一参数传入函数中进行运算，核心部分代码如下图所示所示。

```
def main():
    savepath = 'cov.db'
    c_name = ['俄罗斯', '巴西', '德国', '意大利', '法国', '美国', '英国', '西班牙']
    url = 'https://api.inews.qq.com/newsqa/v1/automation/modules/list?modules=FAutoCountryMerge'
    c_r = [0,0,0,0,0,0,0,0]
    c_p0 = [20000, 500000, 40000, 10000, 50000, 150000, 1000, 1000]
    country_data = []
    country_d_data = []
    duoyu = []
```

在对数据进行拟合上，本系统选择了 Scipy.optimezi 库的 curve_fit 函数。Scipy 是一个用于数学、科学、工程领域的常用软件包，可以处理插值、积分、优化、图像处理、常微分方程数值解的求解、信号处理等问题。

第六节 云服务器部署

本部分将分别介绍云服务器的选择，调试以及部署工作。

在云服务器的选择上由于本系统计划在未来增加微信小程序版本，因此决定选用腾讯云。经过慎重考虑，最终选择了腾讯云的轻量应用服务器。

轻量应用服务器（Lighthouse）是新一代面向中小企业和开发者的云服务器产品，具备轻运维、开箱即用的特点，适用于小型网站、博客、论坛、电商以及云端开发测试和学习环境等轻量级业务场景，相比传统云服务器更加简单易用，并通过一站式融合常用基础云服务帮助用户便捷高效的构建应用。使用腾讯云轻量应用服务器有助于开发人员将精力更多的集中在业务的开发上而不是配置服务器本身。因此最终选择了该款产品。在服务器操作系统的选择上而言，与 Windows 操作系统相比，Linux 操作系统的性价比、安全性以及开放性，还有版权方面的都更加具备优势[28]。而 ubuntu 则是一个成熟的发行版，有丰富的社区资源。并且由于本项目所使用的的服务器资源并不丰富，使用 Linux 系统是这类资源有限的服务器的最佳选择。Linux 目前有众多发行版，在这之中最终选择了 ubuntu。ubuntu 有着活跃的社区，可以帮助解决开发中遇到的各种问题。

并且在自身 linux 知识的掌握上较为有限的情况下选择使用宝塔面板来提供

服务器部分功能的可视化界面，降低使用 linux 的门槛，便利开发流程。宝塔面板界面如图 5-17 所示。

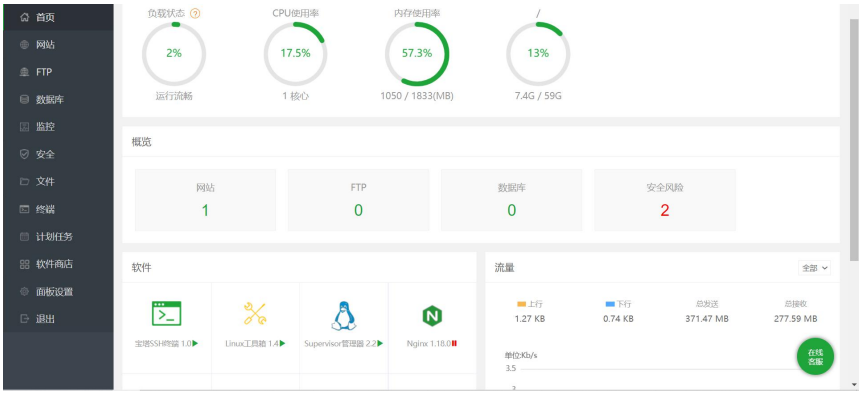

图 5-17 宝塔面板示意图

在本系统迁移至云服务器上之后，首先应该在开发环境中运行并进行调试。设置 debug 模式为开启状态并将地址设置为 0.0.0.0。这样可以使用本机地址加端口号直接打开该网页以便一边修改一边查看效果。

在调试工作完成之后，关闭 debug 模式。使用面板中的计划任务功能设置爬虫定时运行。本系统共有三个爬虫文件负责爬取三种内容。main3.py 文件负责爬取用于疫情预测部分的数据，由于这部分数据以天数为单位，每天会更新前一天的数据，因此设置一天只爬取两次。main2.py 文件负责爬取百度疫情大数据报告中关于疫情相关新闻部分的内容，以便制作热搜词云图，因此设置需要以小时为单位进行多次爬取。main1.py 负责爬取除了以上两部分内容之外的全部数据，这部分数据的更新频率适中，因此设置以小时为单位进行多次爬取。

为了能够在云服务器中后台运行 Flask，避免关掉终端后 Flask 停止运行，以及进程因为某种原因意外掉线后可以自动重启，使用 supervisor 进程管理程序添加对 Flask 的进程守护以便解决以上问题。

第六章 新冠疫情可视化系统的系统测试

系统测试是系统开发流程中的重要环节之一。系统测试是指用一些列方法来对软件的质量进行检测。判断该软件是否符合设计之初的基本要求，总结软件目前的技术状态与预期目标之间的差别。以便在之后加以改进。

系统测试目前可以使用人工测试和自动化测试的方法进行测试。本系统目前的状态使用人工测试就可以获得很好的效果。

为了检验目前已经开发完成的新冠疫情数据分析及可视化系统，本章节将对本系统进行系统测试。为了对本系统进行快速有效的测试，首先，根据本论文之前章节所分析得出的本系统应该具有的功能来编写测试用例。之后根据已完成的测试用例逐条来对系统进行测试。得到测试结果之后再根据反馈对系统进行改进和再开发工作。并在修改完成后反复进行此流程。在系统的缺陷基本修改完成后，将整个系统测试流程中的有价值内容记录至本章内容中。本章节通过系统测试用例以及在测试过程中的说明来阐述本新冠疫情数据分析及可视化系统的全部测试流程。

第一节 进行系统测试的目的

对新冠疫情数据分析及可视化系统进行系统测试的目的是通过对整个系统的测试，检验它是否有不符合最初设计要求的问题。通过系统测试可以可以发现本系统在设计之初十分存在一些设计不当的地方，比如该系统的安全性是否足够高，是否能够抵御各种类型的攻击，保证系统的正常稳定运行。再例如该系统能否在有大量用户访问的情况下还能正常工作。总的来说，进行系统测试的主要目的还是尽可能的检查出系统在按照既定要求进行编写时可能出现的诸多问题。

第二节 系统系统测试环境

为了保证系统测试的结果是完全真实可靠的，本次针对新冠疫情数据分析及可视化系统的测试是在用户普遍使用的环境和设备上进行的，这样做的目的是为了贴近用户的实际使用情况。具体的测试环境如下表 6-1 所示：

表 6-1 进行测试的环境和设备表

测试软件和硬件配置	软件硬件的版本及型号
处理器	Intel(R) Core(TM) i7-7700HQ
内存	16.0 GB (15.9 GB 可用)
操作系统	Windows 10 家庭中文版
浏览器	Edge 浏览器（90.0.818.56 版本） Chrome 浏览器(90.0.4430.212 版本)

第三节 系统功能测试

对于本新冠疫情数据分析及可视化系统，本次系统测试的主要是用来检测该平台的各个功能在一般使用条件下能否正常运作。再结合设计阶段对于本系统的功能的规划，本次测试将对划分好的每个功能设计测试用例。并且按照设定好的测试用例进行充分的测试。

一、网站被正常打开的测试用例

网站被正常打开的测试用例如表 6-2 所示。

表 6-2 网站被正常打开的测试用例表

用例描述	测试网站是否能被正常打开
测试方法	输入网站网址并进入
预期结果	网站可以打开，所有数据正常显示
测试结果	网站可以打开，所有数据正常显示

二、动态图表功能测试用例

动态图表功能测试用例如表 6-3 所示。

表 6-3 动态图表功能测试用例表

用例描述	测试动态图表是否正常
测试方法	在图片上移动鼠标观察
预期结果	数据均正常显示
测试结果	数据均正常显示

第四节 测试总结

本章的主要任务是对新冠疫情数据分析及可视化系统进行系统测试工作。以

上小节中分别对测试的设备及其环境、测试用例、测试的结果进行了介绍。结果证明本次系统测试的结果符合之前分析的平台的需求分析部分的内容。对于用户来说本平台也能够满足其使用需求。

就本次系统测试的过程和结果而言，本次开发的新冠疫情数据分析及可视化系统基本符合预期，但是就本系统在未来的持续性维护来说，仍存在一些问题。因为爬虫所爬取的网站可能使用新的反爬虫措施使得爬虫失效，数据接口可能会发生变化导致无法读取数据。所以本项目在本次测试之后的运营维护中将继续进行改进和进一步开发，并且定期进行维护以便保证系统的高可用性。

第七章 总结与展望

通过回顾在完成毕业设计的数个月中的学习和开发经历，本次毕业设计主要完成了以下几个部分的工作。

首先是阅读了数据可视化领域的相关资料，研究了当下数据可视化领域的发展趋势和现如今的应用情况，并实际动手进行了练习。对目前成熟的可视化框架的优劣进行了比较，对数据可视化领域有了基础的认识。

其次，通过文档和博客对 Python 的各种框架进行了解，为使用 Python 作为后端语言，使用 Flask 作为后端框架打下了坚实的理论和实践基础。

然后，对目前流行的爬虫工具和解析工具进行了学习，再综合之前学习的多种技术在完成毕设之前首先完成了一个爬取豆瓣电影数据的爬虫小项目进行训练，有了如何搭建一个可视化网站的基本概念，较为熟练的掌握了搭建此类平台的各种技术，充分保障了接下来毕业设计的代码编写工作的顺利进行。

随后，对毕业设计的业务逻辑和使用场景进行了分析，确定了本系统所应该具有的功能以及各功能需要用到的技术。据此完成了数据库和爬虫以及解析部分内容的设计工作，为完成项目搭好了基本架构，为之后的开发工作做好了准备。

接下来根据既定的开发任务和开发计划在计划时间内完成了全部开发工作。一步一步的实现当初设计中的大部分功能。部分功能受限于学习的深度和时间限制进行了重新设计，最终较为圆满的完成了软件的编写工作。

最后，对完成全部代码编写工作的毕业设计进行了充分的测试，保障了该系统运行的稳定性和可靠性。

从对本系统所需技术的陌生到熟悉，从模仿现有的此类系统的设计到根据自身的需求进行开发，在完成毕设的这段时间里学习到了诸多知识。但是尽管进行了大量的工作，但本系统还是有一些不完善之处。因为部分技术的学习不是足够深入，以及时间的限制。所以也留下了一些遗憾，一些计划中的功能没能实装，一些功能的设计和实现本可以变得更好。

随着国外疫情的持续肆虐和国内疫情零星散发的现象仍在持续，在可预见的未来也仍将会持续不短的时间，再加上公众对于疫情中想要获取的信息的重点的不断地变化，继续后续开发工作并长期维护新冠疫情数据分析及可视化系统是十分有必要的。因此在以后的学习和实践过程中将会对本平台进行进一步的设计和开发以及维护。在爬虫方面会继续优化网页解析部分的内容，减轻服务器负荷。在动态图表方面会继续增加更多的动态图表以便及时传递疫情相关信息。

最后，衷心祈愿世界能够及早从因为新冠疫情造成的严重破坏中复苏。这场疫情来势汹汹，不仅仅造成了物质上的极大损失，更造成了公众精神上的极大创

伤。在疫情严重时，假新闻横飞，民众人心惶惶。信息的透明和及时的传递从未变得如此重要。因此，各大疫情信息发布平台创立，对于破除假新闻的恶劣影响，稳定公众的抗疫信心起到了极大的作用。希望有朝一日此类平台因为疫情的结束完成了自己的历史使命而纷纷关闭。

参考文献

[1] 中国互联网络信息中心.第 45 次《中国互联网络发展状况统计报告》[EB/OL]. http://www.cnnic.net.cn/hlwfzyj/hlwxzbg/hlwtjbg/202004/t20200428_70974.htm, 2020-4-28.

[2] 国家卫健委疫情防控动态[EB/OL].[2021-05-20].http://www.nhc.gov.cn/xcs/yqfkdt/gzbd_index.shtml

[3] 赵青,武洁雯,房元圣,杨昕娉,梁作如,纪瀚然,张荣娜,庞明樊,戚晓鹏.2021 年 2 月全球新型冠状病毒肺炎疫情风险评估[J].疾病监测,2021,36(03):204-208.

[4] 李崇寒.美国现代医院、医学院的发源地　发布疫情图的约翰斯·霍普金斯大学[J].国家人文历史,2020(11):18-23.

[5] 新型冠状病毒肺炎追踪[EB/OL].[2021-05-20].https://www.bing.com/covid/local/unitedstates

[6] WHO Coronavirus (COVID-19) Dashboard | WHO Coronavirus (COVID-19) Dashboard With Vaccination Data[EB/OL].[2021-05-20].https://covid19.who.int

[7] 王照.Python 语言编程特点及应用[J].电脑编程技巧与维护,2021(03):19-20+44.

[8] Kuldeep Singh Kaswan,Jagjit Singh Dhatterwal,B. Balamurugan. Python for Beginners[M].CRC Press:2021-05-25.

[9] José Unpingco. Python Programming for Data Analysis[M].:2021-05-06.

[10] Tarek Amr,Rayna Stamboliyska. Practical D3.js[M].Apress, Berkeley, CA:2016-01-01.

[11] Deqing Li,Honghui Mei,Yi Shen,Shuang Su,Wenli Zhang,Junting Wang,Ming Zu,Wei Chen. ECharts: A declarative framework for rAPId construction of web-based visualization[J]. Visual Informatics,2018,2(2).

[12] 杨明欣.基于 ECharts 的可视化通用视觉语言分析[J].设计,2020,33(22):114-117.

[13] Yifu Sheng,Weida Chen,Huan Wen,Haijun Lin,Jianjun Zhang. Visualization Research and Application of Water Quality Monitoring Data Based on ECharts[J]. 1 College of Engineering and Design, Hunan Normal University, Changsha, China.;2 Zhaoyin Network Technology (Shenzhen) Co., Ltd., Shenzhen, China.; Corresponding Author: Jianjun Zhang.,2020,2(1).

[14] Apache ECharts[EB/OL].[2021-05-20].https://ECharts.apache.org/zh/index.html

[15] 李宗杰.Python 脚本语言在 Web 开发中的应用探究[J].电子元器件与信息技术,2020,4(12):136-137.

[16] 申纯洁.嵌入式数据库 SQLite 的编程及 DD-WRT 路由器程序的仿真[J].湖北第二师范学院学报,2016,33(08):55-59.

[17] 实时更新：新冠肺炎疫情最新动态[EB/OL].[2021-05-20].https://news.qq.com/

zt2020/page/feiyan.htm#/.
[18]实时更新|新冠肺炎疫情动态地图[EB/OL].[2021-05-20].https://wp.m.163.com/163/page/news/virus_report/index.html?_nw_=1&_anw_=1.
[19]刘玉玲,郑力新.新冠肺炎疫情数据的抓取及可视化研究[J].电子设计工程,2021,29(07):40-44.
[20]新型冠状病毒肺炎疫情实时大数据报告[EB/OL].[2021-05-20].https://voice.baidu.com/act/newpneumonia/newpneumonia/?from=osari_pc_1.
[21]杨登,袁芳.基于 Python 爬虫的数据分析[J].中国新通信,2020,22(18):76-77.
[22]庄礼金,戴泽鑫.网络爬虫的设计与实现[J].信息技术与信息化,2020(12):47-49.
[23]Kyoko Namikawa. An Introduction To Jquery[M].Tritech Digital Media:2018-08-23.
[24]Jing Bo,Qian Zheng,Zareipour Hamidreza,Pei Yan,Wang Anqi. Wind Turbine Power C urve Modelling with Logistic Functions Based on Quantile Regression[J]. Applied Scie nces,2021,11(7).
[25]赵晓晶.结合现实评述马尔萨斯人口理论[J].时代金融,2018(03):321-322.
[26]Secord James A. Revolutions in the head: Darwin, Malthus and Robert M. Young.[J]. British journal for the history of science,2021,54(1).
[27]徐荣辉.逻辑斯蒂方程及其应用[J].山西财经大学学报,2010,32(S2):311-312.
[28]魏秀卓.Linux 操作系统的应用及发展[J].信息记录材料,2021,22(01):188-190.

致　谢

大学这四年不知不觉就过去了，到了写论文的致谢部分的时候才意识到大学期间的工作已经接近尾声，很快就要毕业了。不得不感慨大学这四年过得真是太快了，感觉昨天才离开高考考场，结束高中生活，今天就又要告别大学了。

来到天津科技大学就读本专业实在是我人生中的一大幸运，在这里，我的宿舍的室友关系融洽，班上的同学都积极好学，所参与社团中的同学也给了我很大的鼓励和支持。在这样的环境中完成大学的学业我的心中只有感恩。

感谢教育我的老师们，他们在教学上悉心指导，态度上无比的和蔼可亲，现在真是后悔漫长的四年中没有与老师们多多交流和探讨。我会永远铭记我的老师们。在这其中，特别感谢我的毕业设计的指导老师李伟，在他的指导下我顺利的完成了此次毕业设计项目。

在此还要感谢大学四年里帮助到我的诸多朋友。

感谢我的室友们，我们在一起度过了快乐的四年时光，他们的包容和善意功不可没。

感谢柴梦妍，何心月，张樊昊，景石宽。他们是我完成各种任务的支柱。

感谢齐鹏，是他邀请我加入大创团队，在这个团队中我们一起学习，一起进步。除了圆满完成大创任务以外，我们每个人都从中学到了很多知识。

以及感谢我的家人，在他们的大力支持下我得以心无旁骛的完成学业。让我面临任何选择时都能从容应对。