



电子科技大学中山学院

University of Electronic Science and Technology of China, Zhongshan Institute

毕业设计(论文)

基于 Vue 的疫情实时展示 设计与实现

教学单位: 计算机学院

专业名称: 计算机科学与技术

学 号: 2017030103089

学生姓名: 袁鑫

指导教师: 宋喜佳(讲师)

指导单位: 计算机学院

完成时间: 2021 年 5 月 30 日

电子科技大学中山学院教务处制发

独创性声明

本人声明：所呈交的学位论文，是本人在指导老师的指导下独立进行研究工作所取得的研究成果。除文中特别加以标注、所列参考文献和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学中山学院或其它教育机构的学历、学位或课程、培训等非获奖类证书而使用过的成果材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。本人完全意识到本声明的法律后果由本人承担。

作者签名：

日 期：

论文使用授权声明

本学位论文作者完全了解电子科技大学中山学院有关保留、使用学位论文的规定，本人同意学校有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权电子科技大学中山学院可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

(保密的学位论文在解密后应遵守此规定)

论文级别：本科论文

学科专业：

论文题目：

作者签名：

日 期：

导师签名：

日 期：

基于 Vue 的疫情实时展示 设计与实现

[摘要] 近两年来,新型冠状病毒引发的肺炎疫情肆虐,第一时间获取清晰、直观、可靠的数据是帮助民众“理清”疫情发展形势、加强对疫情的心理建设和构筑正确认识的重要手段。相较于通过电视新闻及网络获取的零碎信息,本疫情实时展示系统整合国内外更全面的数据,结合“央视频”发布的疫情新闻与辟谣信息,为用户打造了一个数据可靠、查询功能完备的疫情信息获取平台。

本系统基于 Vue.js 前端开发框架,使用 Vue-Cli 搭建系统。利用 JetBrains 的 IDE 开发工具 IntelliJ Idea 解决因协议、端口不同而无法请求到域名内资源的问题后,使用 axios 创建 http 请求,通过 API 接口从网络获取疫情数据。借助 ECharts 图形组件,通过世界地图、折线图、柱状图、数据集等多种形式,完成对国内外新冠肺炎疫情数据的展示与交互。为了让用户有更好的体验,使用 vue-aplayer 音频插件制作了简易的音频播放器,实现音乐的获取、播放、音量调节等功能。本系统经过测试,基本实现了上述全部功能,能高效稳定的运行,用户体验良好。

[关键词] 新冠肺炎; Vue.js 框架; 统计图; 显示

Design and Implementation of L^AT_EX Thesis of Zhongshan Institute

[Abstract] In the past two years, the pneumonia epidemic caused by the new type of coronavirus is raging. Obtaining clear, intuitive, and reliable data at the first time is an important means to help the people "clarify" the development situation of the epidemic and build a psychological construction and correct understanding of the epidemic. Compared with the fragmentary information obtained through TV news and the Internet, this real-time epidemic display system integrates more comprehensive data at home and abroad, combined with epidemic news and rumors released by "Central Video", to provide the public with a more convenient way to query epidemic information.

This epidemic real-time display system is based on the Vue.js front-end development framework, using Axios, Vue-Cli and other technologies, using JetBrains' IDE development tool IntelliJ Idea to solve the network cross-domain problem, and obtain epidemic data from the network through the API interface, with the help of ECharts graphics components, Through the world map, line chart, histogram, data set and other forms, to complete the display and interaction of domestic and foreign new crown pneumonia epidemic data, using the vue-aplayer audio plug-in, to achieve simple music acquisition, playback, volume adjustment, etc. Music player function.

[Keywords] 2019-NCov; Vue.js framework; statistical graph; display

目录

第 1 章 绪论	1
1.1 课题背景	1
1.2 目的意义	1
1.3 论文主要工作	1
1.4 论文组织结构	1
第 2 章 相关技术和理论基础	3
2.1 Vue.js	3
2.2 Node.js	3
2.3 跨域设置	3
2.3.1 同源策略	3
2.3.2 跨域限制	3
2.3.3 实现跨域的方法	3
2.4 请求数据	3
2.5 ECharts	4
第 3 章 系统分析(需求分析)	5
3.1 功能需求分析	5
3.1.1 省市信息搜索	5
3.1.2 音乐播放器	5
3.2 系统分页设计	5
3.3 系统数据分析	6
3.3.1 数据可靠性分析	6
3.3.2 数据时效性分析	6
3.3.3 数据展示方式分析	6
3.4 非功能需求分析	10
3.4.1 可行性需求	10
3.4.2 技术性需求	10
3.4.3 用户交互与体验	10
第 4 章 系统设计	11
4.1 总体设计	11
4.2 详细设计	12
4.2.1 数据获取	12
4.2.2 数据展示	14

4.2.3	模块设计	16
第 5 章	系统实现与测试	18
5.1	系统实现	18
5.1.1	解决跨域问题	18
5.1.2	全球疫情数据页面	18
5.1.3	全国疫情数据页面	21
5.1.4	疫情新闻页面	27
5.1.5	更多资讯页面	30
5.2	系统测试	34
5.2.1	全球数据页面	34
5.2.2	全国数据页面	35
5.2.3	疫情新闻页面	37
5.2.4	更多资讯页面	38
第 6 章	总结和展望	39
6.1	本文总结	39
6.2	未来展望	40
附录 A	源码	41
	参考文献	42
	致谢	43

图目录

3-1	分页设计	5
3-2	地图示例	7
3-3	折线下方是否着色对比	7
3-4	标签图例样式	8
3-5	图 3-5 折线图示例	8
3-6	数字居中与右对齐效果对比	9
3-7	水平与垂直显示多系列数据效果对比	9
4-1	系统设计	11
4-2	北京疫情统计	12
4-3	世界疫情返回参数	12
4-4	新闻信息返回参数	13
4-5	辟谣信息返回参数	13
4-6	世界地图展示样式	14
4-7	全国地图展示样式	14
4-8	全勾选	14
4-9	部分勾选	15
4-10	每日新增人数趋势示例图	16
4-11	搜索流程图	17
5-1	现有确诊数据和累计确诊数据	19
5-2	累计确诊人数显示	20
5-3	全国疫情地图	23
5-4	省份显示多条数据	24
5-5	国内疫情数据省市两级列表展示	26
5-6	轮播图	29
5-7	跳转至央视新闻	30
5-8	音乐播放器	32
5-9	不同地区现有确诊信息	34
5-10	不同地区累计确诊信息	34
5-11	列表展示各国家疫情数据	35
5-12	国内不同省份疫情数据	35
5-13	国内不同省份疫情数据	36
5-14	省市二级列表疫情信息	36
5-15	轮播图	37
5-16	新闻	37
5-17	实时辟谣信息	38

5-18 音乐播放器	38
----------------------	----

表目录

4-1	中国病例列表样式	15
4-2	中国病例二级列表样式	15
4-3	国外病例列表样式	16

第1章 绪论

1.1 课题背景

2020年以来,新型冠状病毒肺炎疫情的爆发影响了全球几乎所有人的生活。为了使公众了解疫情发展态势、实现人们掌握传染病发展趋势的愿望,信息可视化作为一种设计手段和呈现方式,能够有效地将原本单调的统计数字转化为可读性强的信息数据。与新冠疫情爆发初期一样,流行病学专家曾光说,“不怕病例就怕不报”。人们对一种传染性疾​​病及其病原体的致病性、传播途径、病例数量了解得越多,对它的恐慌、恐惧就会越少。

大多数省份出现新的感染病例。以 Vue.js 作为前端框架,利用 ECharts 组件,用多种图表显示疫情统计数据的变化趋势,通过中国地图、世界地图显示疫情在国内外的总体分布状况。本疫情实时展示系统正是在此背景下应运而生。

1.2 目的意义

开发一款疫情实时展示系统,能有效的将统计数字转化成具有可读性的信息数据,让民众明晰新冠疫情的发展方向,实现人们掌握传染病发展趋势的愿望。

1.3 论文主要工作

介绍本研究课题的来源及主要研究内容。

论文的背景见1.1,论文的目的意义见1.2。

本作品主要工作如下:

- 整体系统架构设计;
- 功能模块的设计与实现;
- 前端的编程与实现;
- 数据的采集整理与展示。

1.4 论文组织结构

第1章说明疫情实时展示系统的课题诞生背景,存在的目的意义以及主要工作,全文的组织结构。

第2章说明了系统开发的相关技术。包括 Vue.js 前端开发框架。Vue-Cli、Echarts 组件等技术。

第3章对疫情实时展示系统做了需求分析。详细介绍了系统在实际应用中的功能需求,系统的业务分析,系统的用例分析,系统的功能分析和非功能性需求。

第4章是总体设计。说明了整体结构、各个模块和数据交互方式。

第5章介绍如何实现该系统，通过源码说明关键技术。

第6章对系统的发展进行了总结和展望，分析了系统存在的问题和需要进一步改进的地方。

第 2 章 相关技术和理论基础

2.1 Vue.js

Vue 是一套用于构建用户界面的渐进式框架,与其它大型框架不同的是,Vue 被设计为可以自底向上逐层应用,Vue 的核心库只关注视图层,不仅易于上手,还便于与第三方库或既有项目整合;另一方面,当与现代化的工具链以及各种支持类库结合使用时,Vue 也完全能够为复杂的单页应用提供驱动。^[1]

2.2 Node.js

Node.js 发布于 2009 年 5 月,是一个基于 Chrome V8 引擎的 JavaScript 运行环境,即依赖于 Chrome V8 引擎解释代码,能够让 JavaScript 运行在服务端的开发平台,高并发是其突出优势,可用以搭建响应速度快、易扩展的网络应用。^[2]

2.3 跨域设置

2.3.1 同源策略

疫情数据每日都会变化,为保正疫情数据的时效性,多来源于新浪、百度、天行数据等站点提供的疫情数据 API,使用 Axios 远程访问并获取到 json 数据。

2.3.2 跨域限制

跨域限制主要体现在以下几个方面:

1. 在跨域的网址上无法得到 Cookie 和 LocalStorage 数据。
2. 无法访问网站的 DOM。
3. 无法向跨域的服务器送出 Ajax 请求。

2.3.3 实现跨域的方法

配置文件设置。在项目根目录中打开 config.js,配置网络疫情数据接口及代理接口地址,并设置为允许跨域。

2.4 请求数据

Vue 中导入 EChart 组件,显示国内外疫情数据;在 Vue 对象数据中定义多个数组,分别储存确诊、疑似确诊、已治愈、已死亡等多项数据,并作为 ECharts 图表组件的数

据源。

2.5 ECharts

ECharts 是开源可视化库，涵盖多种图表。^[3] 兼容当前绝大部分浏览器及兼容多种设备，可随时随地任性展示。

第3章 系统分析(需求分析)

3.1 功能需求分析

本系统主要分为数据获取和展示两部分。数据来源为百度、新浪和天行数据。向用户展示的信息分为四大模块,其中包含全球疫情、国内疫情、新闻数据及汇总数据。用户可以通过跳转分区及搜索功能获取想要的信息。

3.1.1 省市信息搜索

各省市列表信息基于确诊人数进行降序排序,存在用户无法第一时间查看所在省市信息的情况。用户可以通过搜索功能定位信息,使信息的获取更高效。

3.1.2 音乐播放器

歌曲鼓舞人心,音乐播放器能使用户在获取疫情信息时保持良好的心态。设计的音乐播放器通过jsDeliver加速来优化从仓库获取歌曲的速度,降低网络拥塞,提高访问响应速度,增强用户体验。

3.2 系统分页设计

系统分为全球数据、全国数据、疫情新闻及更多资讯四个分页。疫情信息以块形式显示,以防止用户由于数据的复杂性而在第一时间获取所需信息。

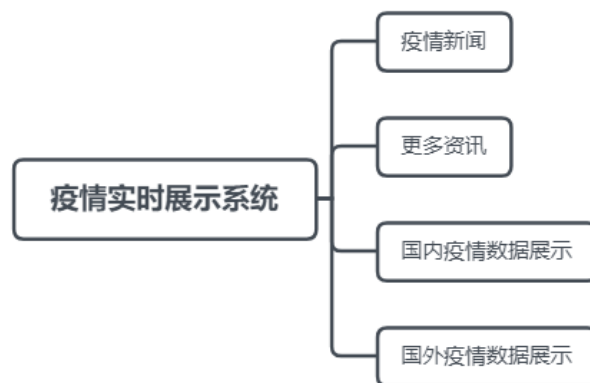


图 3-1 分页设计

3.3 系统数据分析

3.3.1 数据可靠性分析

从数据接口和接口数据源进行可靠性分析

1. 数据接口

疫情实时显示系统的数据接口来自新浪、百度、天行数据等。

2. 数据来源

国外数据源于 WHO 及各国的正式报道。国内数据源于各省市级报告和国家卫健委。

3.3.2 数据时效性分析

从统计口径和数据更新时间两方面进行分析

1. 统计口径

由于有时差，故使用北京时间；新增的数据和走势图数据是将昨天的数据从前一天的数据中减去后的结果，每天更新一次。

2. 更新时间

由于需要查实国外疫情数据或等待相关数据，会存在较短延迟。

3.3.3 数据展示方式分析

图表是一种常见的数据可视化方式。但其种类繁多，故在此分析适合疫情数据的展现方式。

1. 地图

地图作为一种主流的显示疫情方法，是统计疫情数据的不二之选。地图适用于和地理位置相关的数据分析。如不同城市的人口统计信息。单击地图上的某个区域以冒泡当前区域的特定数字信息。具体分析使用地图的策略如下：

(a) 使用色块来强调，尽量避免使用簇或剖面线来强调一些区域。

(b) 地图轮廓避免复杂，保证简洁标准。

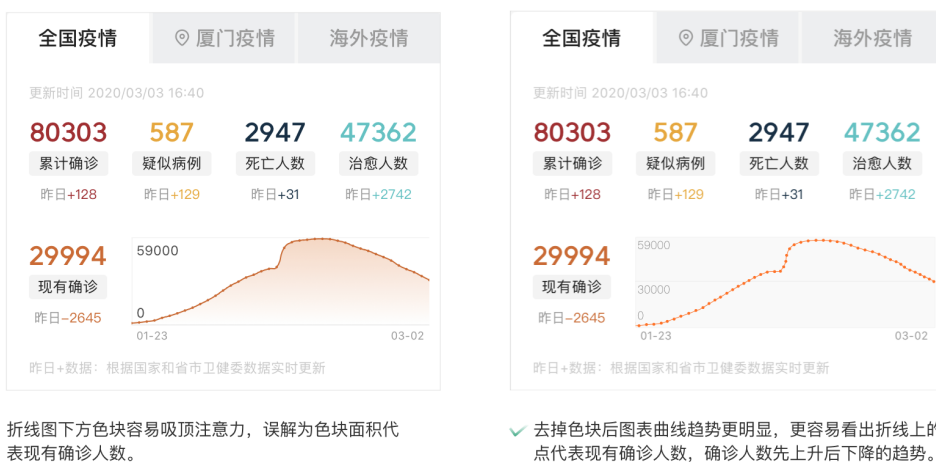


图 3-2 地图示例

2. 折线图

折线图有棱有角非常直观。疫情地图中用折线图描述全国疫情新增趋势。折线图还支持单击图表上的点以显示与当前横坐标相对应的纵坐标值，以帮助更详细地查看信息。具体分析使用折线图的策略如下：

(a) 折线图能很明显的表现出数据的走向，但注意避免给折线与横轴连成的区块上色。区县着色后看起来更像区域地图，容易造成数据的误解。如图 3-3 左边，很难区分是折线上的点指示 29994 的确诊病例数还是色块区域指示 29994 的确诊病例数；3-3 右图里，图表曲线的趋势就变得很明显，现有的确诊病例数量呈先上山后下山的趋势。



折线图下方色块容易吸顶注意力，误解为色块面积代表现有确诊人数。

✓ 去掉色块后图表曲线趋势更明显，更容易看出折线上的点代表现有确诊人数，确诊人数先上升后下降的趋势。

图 3-3 折线下方是否着色对比

(b) 折线图标不应过高或过低。适当处于二分之三的位置，参照点使用户能更直观的理解数据内容。

(c) 折线图不应过粗或过细。太粗会减少折线变化细节；太细容易与辅助线和背景

混淆。

(d) 图例和标签文字要简洁，易于——对应图表。如图 3-4 所示

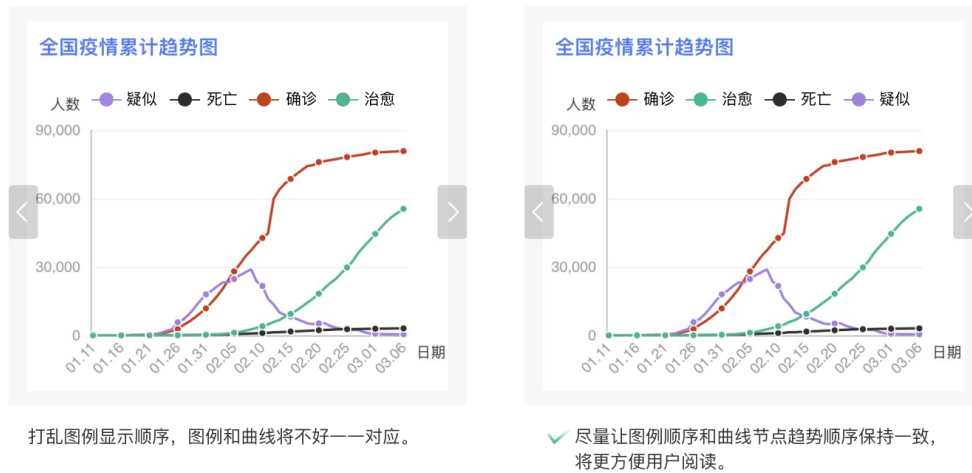


图 3-4 标签图例样式

(e) 折线坐标轴标注应当直观合理。常见为 x 轴按时间推移标注，y 轴按数据增量标注。其中 y 轴尽量使用常用增量，使数据分布状态更能体现真实趋势。



图 3-5 图 3-5 折线图示例

3. 表格

表格可以容纳大量且多种类的信息。具体分析使用表格的策略如下：

(a) 表格编号尽量右对齐，能更直观的对比数据。考虑到移动端界面只能容纳有限信息，排版上有时仍会使用居中排列。如图 3-6 所示



图 3-6 数字居中与右对齐效果对比

4. 单位仅在表格第一行显示一次。提高用户阅读效率。
5. 减少网线的使用，用鲜亮的颜色标注某列数据。弱化网格线突出数据，同时使用特殊颜色突出例如每日新增人数等重要信息。
6. 使用垂直显示有多个数据系列的信息。垂直显示，方便用户比较诊断和治愈人数等信息，方便用户阅读。如图 3-7 所示



图 3-7 水平与垂直显示多系列数据效果对比

3.4 非功能需求分析

本系统的非功能需求主要分为可行性、技术性、用户体验三个方面。

3.4.1 可行性需求

疫情爆发后各国家及世卫组织积极统计相关数据，相关企业带头整理信息，为本系统细化并展示疫情信息扎实了根基，使本系统对新冠肺炎疫情信息的分类及展示成为可能。

3.4.2 技术性需求

Vue.js 是当下流行的前端开发 MVVM 渐进式框架，将业务与视图代码分离，让各自职责更清晰，相比传统的调用 JQuery 库方式，Vue.js 侧重于视图层，同时支持 PC 及移动端，因此适合开发各种单页面应用，大幅度的提高了前端开发效率。^[1] 本系统以 Vue.js 作为前端框架，同时应用 Echarts 组件，使用多种图表形式展示数据的分布及变化趋势，以实现不同数据的多样化显示。

3.4.3 用户交互与体验

1. 该系统整合了大部分可获取的疫情信息，为用户获取疫情相关数据提供了更大的可选性，对疫情信息有更全面的掌握。
2. 该系统将相同数据通过不同形式表达，使用户观察数据的角度更多元化。
3. 本系统有快速搜索、地图区块勾选等功能。用户能提高查询效率，节约时间。

第4章 系统设计

4.1 总体设计

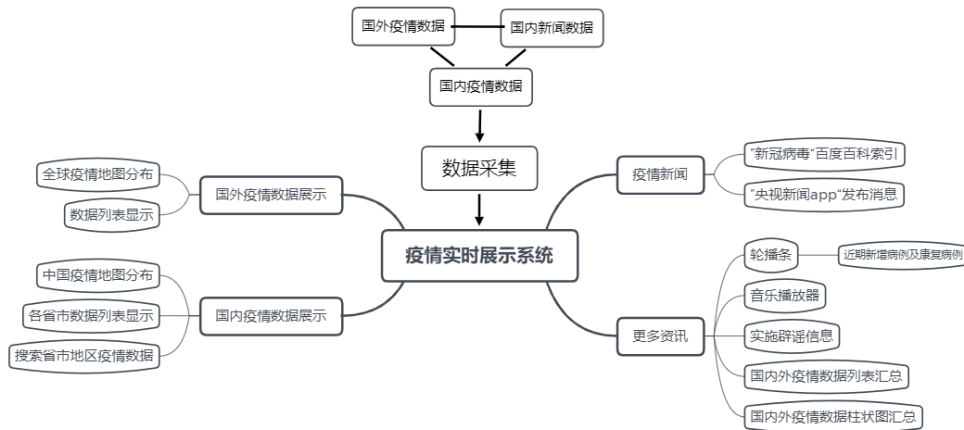


图 4-1 系统设计

经过用户需求分析，本系统设计了四大模块：国外疫情数据展示、国内疫情数据展示、疫情新闻、更多资讯。

1. 国外疫情数据展示

该模块向用户展示了全球疫情信息。其中包含可拖动的新冠肺炎疫情世界分布地图、根据感染人数对各个国家用不同颜色进行标注、现有确诊和累计确诊统计数据的切换、各国疫情信息的侧边栏图表显示等功能。

2. 国内疫情数据展示

该模块向用户展示国内疫情信息。将国内疫情分别用地图和二级省市列表来展示，还为用户提供简单的搜索功能。

3. 疫情新闻

该模块包含疫情实时新闻及资讯。通过轮播，新型冠状病毒预防信息，“央视新闻app”新闻信息三部分组成。

4. 更多资讯

该模块向用户展示了汇总信息。包括实时辟谣信息、音乐播放器等功能。

4.2 详细设计

4.2.1 数据获取

4.2.1.1 国内数据获取

国内各省市新冠肺炎感染及治愈信息主要爬取自新浪,图 4-2 展示部分数据

virus_city:

province	name	conNum	susNum	cureNum	deathNum
北京	海淀区	41	0	0	0
北京	怀柔区	1	0	0	0
北京	丰台区	16	0	0	0
北京	大兴区	22	0	0	0
北京	东城区	3	0	0	0
北京	昌平区	12	0	0	0
北京	西城区	22	0	0	0
北京	朝阳区	35	0	0	0
北京	石景山区	5	0	0	0

图 4-2 北京疫情统计

4.2.1.2 国外数据获取

国外新冠肺炎感染信息调用天行数据提供的接口,接口地址为 <http://api.tianapi.com/txapi/ncovabroad> 图 4-3 展示部分返回参数

名称	类型	示例值	说明
modifyTime	int	1584159933000	数据修改时间
continents	string	欧洲	大洲
provinceName	string	意大利	地区名
currentConfirmedCount	int	14955	现存确诊人数
confirmedCount	int	17660	累计确诊人数
suspectedCount	int	1439	治愈人数
deadCount	int	1266	死亡人数
locationId	int	965008	地理位置编号
countryShortCode	string	ITA	国家代码

图 4-3 世界疫情返回参数

4.2.1.3 新闻数据获取

新闻信息调用天行数据提供的接口,接口地址为 <http://api.tianapi.com/txapi/ncov/index>,图 4-4 为部分返回参数

名称	类型	示例值	说明
news	object	新闻资讯数组	新闻动态列表
desc	object	疫情概括数组	全国疫情数据
currentConfirmedCount	int	55881	现存确诊人数
confirmedCount	int	74679	累计确诊人数
suspectedCount	int	2053	累计境外输入人数
curedCount	int	16676	累计治愈人数
deadCount	int	2122	累计死亡人数
seriousCount	int	306	现存无症状人数
suspectedIncr	int	8	新增境外输入人数
currentConfirmedIncr	int	-2002	相比昨天现存确诊人数
confirmedIncr	int	403	相比昨天累计确诊人数
curedIncr	int	2289	相比昨天新增治愈人数
deadIncr	int	116	相比昨天新增死亡人数
seriousIncr	int	4	相比昨天现存无症状人数

图 4-4 新闻信息返回参数

4.2.1.4 辟谣数据获取

辟谣信息调用天行数据提供的接口,实际返回数据源于各大权威报道,接口地址为 <http://api.tianapi.com/txapi/rumour/index>,图 4-5 展示部分返回参数

名称	类型	示例值	说明
id	string	e633815442b986f8220183e6c42d2166	谣言详情ID
date	string	2020-01-23	谣言鉴别时间
title	string	武汉三口之家迪士尼游玩确诊新型冠状病毒肺炎	谣言标题
explain	string	谣言	谣言解释,没有解释时判断类别
imgsrc	string	https://p.qpic.cn/jiaozhen/0/1176d1479aaf4d05b148db998f2e9734/0	谣言图片
markstyle	string	fake	谣言类型: fake虚假、genuine确实
desc	string	上海迪士尼度假区方面对记者表示,“该消息是谣传信息...”	澄清概况
author	string	《经济观察报》社全新商业资讯平台	权威报道来源

图 4-5 辟谣信息返回参数

4.2.2 数据展示

4.2.2.1 地图展示形式

使用 ECharts 展示数据。使用 ECharts 组件里的地图组件，按确诊人数进行着色，多条数据显示则是使用了 tooltip。地图展示分为世界地图和全国地图两方面，世界地图中点击各国会显示该国当前确诊人数，全国地图中点击各省则显示该省确诊总数、现存确诊数、死亡数、治愈数。图 4-5 和图 4-6 分别为世界地图及全国地图展示样式

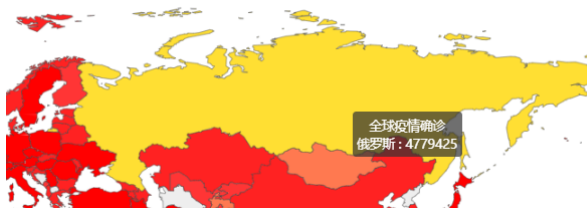


图 4-6 世界地图展示样式



图 4-7 全国地图展示样式

各个颜色区块放置在地图旁便于用户进行对比，用户可以自主勾选展示部分信息。图 4-7 和图 4-8 分别为全勾选及勾选部分的状态

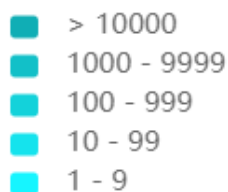


图 4-8 全勾选



图 4-9 部分勾选

4.2.2.2 列表展示形式

列表展示分为中国病例和国外病例两个模块。主要数据分为省份、现存确诊人数、累计确诊人数、死亡人数和治愈人数这 5 类。如表4-1所示

表 4-1 中国病例列表样式

省份	现存确诊	累计确诊	死亡	治愈
湖北	-	-	-	-
香港	-	-	-	-
广东	-	-	-	-
上海	-	-	-	-
黑龙江	-	-	-	-
浙江	-	-	-	-
...	-	-	-	-

中国病例列表中设计了二级列表，点击省份即可显示对应市区感染信息，二级列表如表4-2所示

表 4-2 中国病例二级列表样式

省份	现存确诊	累计确诊	死亡	治愈
湖北	-	-	-	-
武汉市	-	-	-	-
孝感市	-	-	-	-
黄冈市	-	-	-	-
鄂州市	-	-	-	-
...	-	-	-	-
香港	-	-	-	-
广东	-	-	-	-
...	-	-	-	-

国外病例显示的主要信息是国家，增加诊断，累积诊断，死亡和治愈。如表4-3所示

表 4-3 国外病例列表样式

国家	增加确诊	累计确诊	死亡	治愈
美国	-	-	-	-
印度	-	-	-	-
巴西	-	-	-	-
法国	-	-	-	-
俄罗斯	-	-	-	-
土耳其	-	-	-	-
...	-	-	-	-

4.2.2.3 数据集展示形式

该表将每日新增人数作为纵轴，近三个月的时间作为横轴，以每日为单位显示新增确诊人数的起伏。每日新增人数的确定方法是由今日确诊人数减去昨日确诊人数所得。如图 4-9 所示

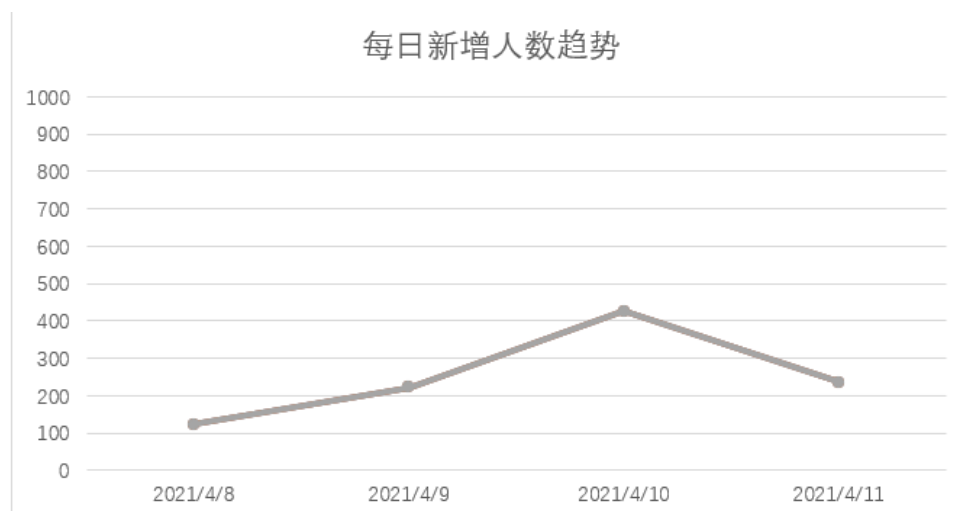


图 4-10 每日新增人数趋势示例图

4.2.3 模块设计

4.2.3.1 省市疫情信息搜索模块

执行简单搜索功能。流程描述如下：

1. 用户点击搜索框，输入内容；
2. 检测到搜索框值变化，取用户输入值；
3. 判断用户输入值。如果为空，则显示原始列表。如果不为空，则过滤原始列表；
4. 将用户输入值与所有列表数据进行遍历匹配，若匹配，则展示匹配条目，完成搜索。

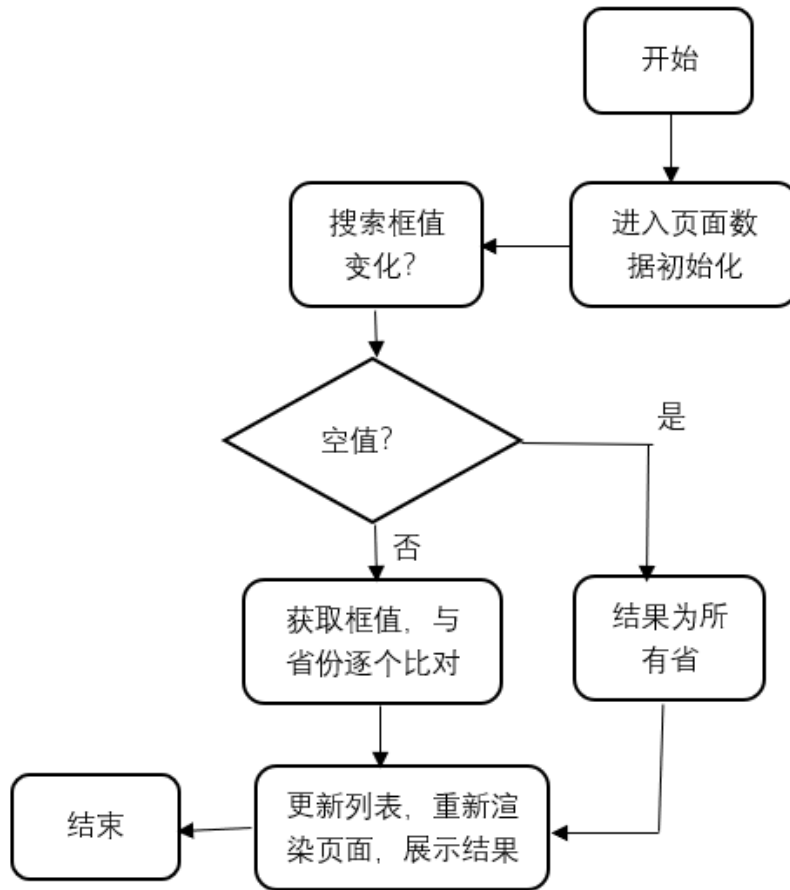


图 4-11 搜索流程图

第5章 系统实现与测试

5.1 系统实现

5.1.1 解决跨域问题

配置文件设置。在项目根目录中打开 config 索引后配置相应接口地址。

```
proxyTable: {  
  '/api': { // 代理的接口名  
    // 此处即为网络疫情数据接口  
    target: 'https://www***.***', // 代理的接口  
    // 地址  
    secure: true,  
    changeOrigin: true, // 允许跨域  
    pathRewrite: {'^/api': ''} // 接口名重写  
  }  
}
```

5.1.2 全球疫情数据页面

5.1.2.1 获取国外疫情数据

使用 axios 将请求信息发送到天星数据 api 并成功返回数据。代码如下:

```
getWorldData(info) {  
  this.axios({  
    url: 'http://api.tianapi.com/txapi/ncovabroad/index',  
    method: 'GET',  
    params: {  
      key: '869941cd56fe09e14b255d12467651bd'  
    }  
  })  
}
```

5.1.2.2 全球地图显示国外疫情数据

引入 ECharts 的 map.js 组件。代码如下:

```
import echarts from "echarts";  
import 'echarts/map/js/world';  
import "echarts-gl"  
import mapdata from '@/untlis/map-WorldOption.js';
```

遍历地图数据。代码如下：

```

then(res => {
  if (res.data.code == 200) {
    this.show=false;
    let WorldData=[];
    let WorldTableData=[];
    for( let i in res.data.newslst){
      let obj={}
      obj.name=res.data.newslst[i].provinceName
      obj.value=res.data.newslst[i][info]
      WorldData.push(obj)
    }
  }
}
    
```

切换现有感染者分布和累计感染者分布，通过点击按钮隐藏前一个状态，重新遍历数据并显示。如图 5-1 所示

```

<b-button v-b- tooltip .hover v-for='(item,index) in switchworlddata' : title ='全球"+item.name+"疫情信息"' :key='index' @click="onswitch(item.info,index)">{{item.name}}</b-button>
    
```

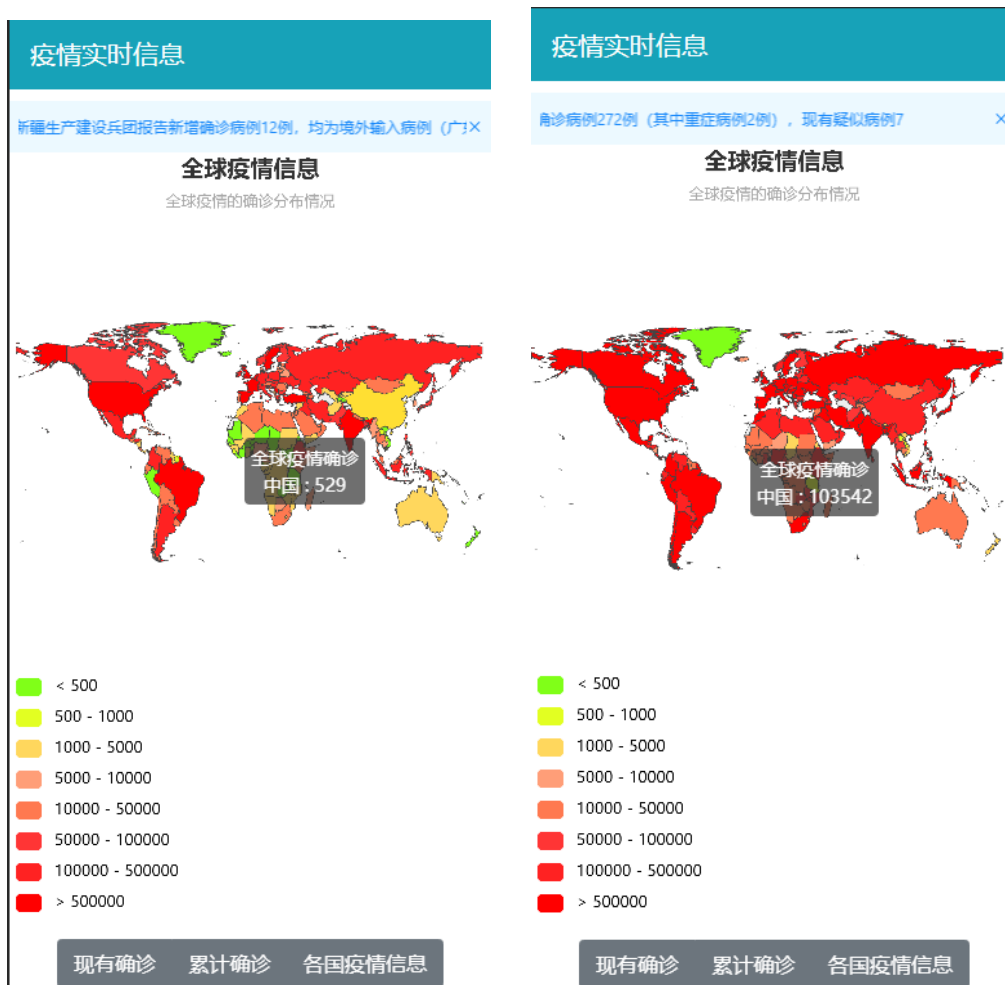


图 5-1 现有确诊数据和累计确诊数据

5.1.2.3 列表显示国外疫情数据

遍历侧边栏。代码如下:

```
let Tableobj={};
Tableobj.国家=res.data.newslst[i].provinceName;
Tableobj.感染数=res.data.newslst[i][info]
Tableobj.死亡数=res.data.newslst[i].deadCount;
WorldTableData.push(Tableobj)
```

点击“各国疫情信息”，显示侧边栏并加载各国疫情信息数据。如图 5-2 所示

```
<b-button v-b-toggle . sidebar - right >各国疫情信息</b-button>
</b-button - group >
<b-sidebar id="sidebar - right" title ="各国疫情信息" right shadow backdrop width="16.5rem"
  >
  >
  <div class ="px-3 py-2">
  <p>
  <b-table striped hover :items="worlditems"></b-table>
  </p>
  <b-img src="https://cdn.jsdelivr.net/gh/devil-trigger/sdn/ncov/6.jpg" fluid thumbnail
    style ="border-radius: 22px;"></b-img>
```

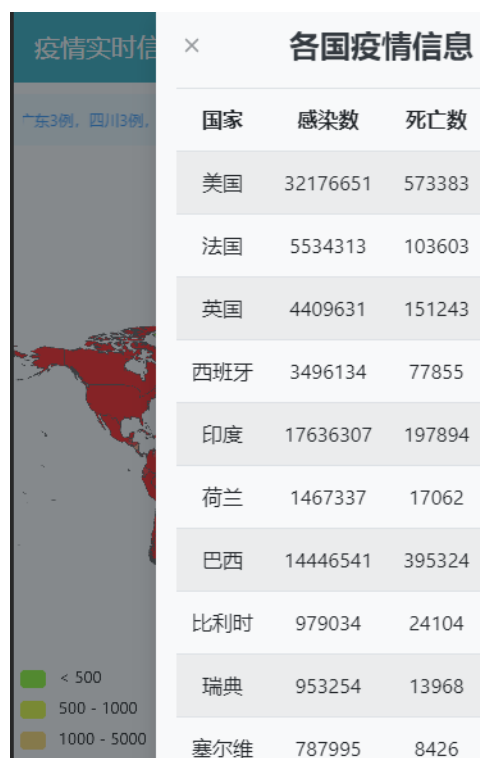


图 5-2 累计确诊人数显示

5.1.3 全国疫情数据页面

5.1.3.1 获取全国疫情数据

通过 async 获取 jsonp 封装的新浪疫情信息地图数据。代码如下：

```

async getData() {
  let that = this;
  // 浏览人次
  let {data: {epidemic}} = await
  jsonp('https://sop-pae.baidu.com/sop/api/epidemic', 'cb', {type: 0}); // 访问人数
  that.epidemicCount = epidemic;
  // 整体数据
  let {data: mapData} = await
  jsonp('https://gwpre.sina.cn/interface/fymap2020_data.json'); // 地图数据
  mapData.list.forEach(item => item.city.sort((a, b) => b.conNum - a.conNum));
  mapData.list.sort((a, b) => b.value - a.value);
  mapData.otherlist.sort((a, b) => b.value - a.value);
  that.mapData = mapData;
}

```

5.1.3.2 地图展示全国疫情数据

引入中国地图。代码如下：

```

import echarts from 'echarts'; // echarts
import 'echarts/map/js/china'; // 中国地图

```

绘出地图标注。代码如下：

```

let that = this;
if (!that.chinaChart) {
  that.chinaChart = echarts.init(that.$refs.chart);
  window.addEventListener('resize', function () {
    that.chinaChart.resize();
  });
}
that.chinaChart.setOption({
  title: {
    text: '全国疫情地图',
    subtext: '数据来源: 国家及各省市地区卫健委',
    textStyle: {color: '#10aeb5', fontSize: 18},
    subtextStyle: {color: '#999', fontSize: 12},
    top: 10,
    left: 10
  },

```

规定地图样式。如图 5-3 所示

```
series : [{
  type: 'map',
  map: 'china',
  left : 10,
  right : 10,
  top: 30,
  bottom: 10,
  itemStyle : {borderColor: '#eee'},
  label: {show: true, fontSize: 8, color: '#333'},
  data: that.mapData.list ? that.mapData.list.map(item => {
    return {name: item.name, value: item.value, obj: item}
  }) : []
}],
visualMap: [{
  type: 'piecewise',
  show: true,
  hoverLink: false,
  pieces: [
    {min: 10000},
    {min: 1000, max: 9999},
    {min: 100, max: 999},
    {min: 10, max: 99},
    {min: 1, max: 9}
  ],
  inRange: {
    color: ['#16f5ff', '#10aeb5']
  },
  itemWidth: 10,
  itemHeight: 8,
  itemGap: 4,
  left : 10,
  bottom: 10,
  padding: 0,
  textStyle : {color: '#666', fontSize: 10}
}]
})
```

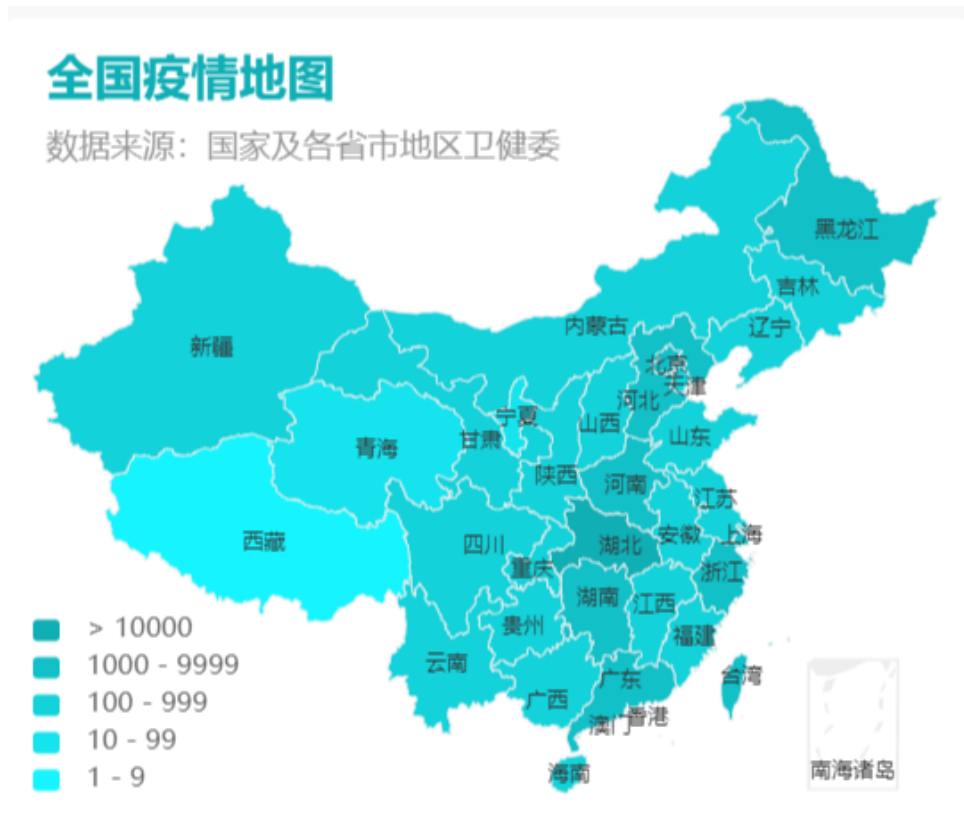


图 5-3 全国疫情地图

数据输入是在数据序列中，只需处理数据格式放入数据中即可。如图 5-5 所示

```

tooltip : {
  show: true,
  formatter : function (obj) {
    if (!obj.data) return;
    return ` <p style="font-size:8px">总数: ${obj.data.value}</p>
    <p style="font-size:8px">现存: ${obj.data.obj.econNum}</p>
    <p style="font-size:8px">死亡: ${obj.data.obj.deathNum}</p>
    <p style="font-size:8px">治愈: ${obj.data.obj.cureNum}</p>`
  }
},
    
```

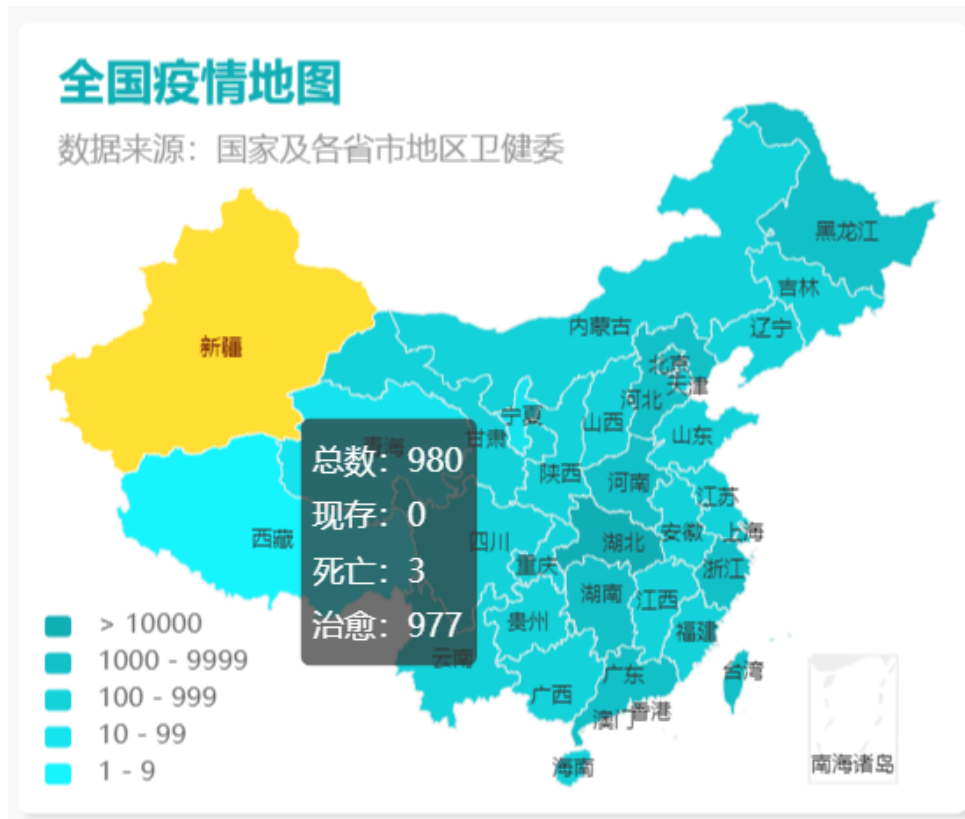



图 5-4 省份显示多条数据

5.1.3.3 列表分级显示全国省市疫情数据

引入城市列表数据到省级列表页面。代码如下：

```
import City from './data-table-city'

export default {
  components: {City},
  props: {
    title: {type: String, default: ''},
    list: {type: Array, default: []}
  }
}
```

城市二级列表 table 样式。代码如下：

```
<div class="table-li">
<div class="table-tr" @click="isCheck=!isCheck">
<div class="table-td a">{{itemData.name}}</div>
<div class="table-td b">{{ title . includes ('中国')?itemData.econNum:itemData.conadd}}</div>
<div class="table-td c">{{itemData.value}}</div>
<div class="table-td d">{{itemData.deathNum}}</div>
```

```

<div class="table-td e">{{itemData.cureNum}}</div>
</div>
< transition name="fade">
<div class="children-table" v-show="isCheck">
<div class="table-tr" v-for="item in itemData.city">
<div class="table-td a">{{item.name}}</div>
<div class="table-td b">{{item.econNum}}</div>
<div class="table-td c">{{item.conNum}}</div>
<div class="table-td d">{{item.deathNum}}</div>
<div class="table-td e">{{item.cureNum}}</div>
</div>
</div>
</ transition >
</div>

```

省级一级列表 table 列表。代码如下：

```

<div class="table-warp">
<div class="table-box">
<div class="title-box">{{ title }}</div>
<div class="table-tr">
<div class="table-td">省份</div>
<div class="table-td">{{ title .includes (' 中国')?' 现存 ':' 增加'}}确诊</div>
<div class="table-td">累计确诊</div>
<div class="table-td">死亡</div>
<div class="table-td">治愈</div>
</div>
<city v-for="item in list" :itemData="item" :key="item.name" : title =" title "></city>
</div>
</div>

```

中国病例					中国病例				
省份	现存确诊	累计确诊	死亡	治愈	省份	现存确诊	累计确诊	死亡	治愈
湖北	5	68157	4512	63640	湖北	5	68157	4512	63640
香港	160	11748	209	11379	武汉	5	50362	3869	46488
广东	40	2334	8	2286	孝感	0	3518	129	3389
上海	62	1972	7	1903	黄冈	0	2907	125	2782
黑龙江	0	1610	13	1597	荆州	0	1580	52	1528
浙江	11	1332	1	1320	鄂州	0	1394	59	1335
河北	0	1317	7	1310	随州	0	1307	45	1262
河南	4	1312	22	1286	襄阳	0	1175	40	1135
台湾	48	1110	12	1050	黄石	0	1015	39	976
北京	9	1057	9	1039	宜昌	0	931	37	894
湖南	3	1044	4	1037	荆门	0	928	41	887
安徽	0	994	6	988	咸宁	0	836	15	821
新疆	0	980	3	977	十堰	0	672	8	664
四川	24	978	3	951	仙桃	0	575	22	553
江西	0	937	1	936	天门	0	496	15	481
山东	5	876	7	864	恩施州	0	252	7	245
江苏	9	718	0	709	潜江	0	198	9	189
重庆	5	596	6	585	神农架林区	0	11	0	11
福建	26	590	1	563	香港	160	11748	209	11379
					广东	40	2334	8	2286
					上海	62	1972	7	1903
					黑龙江	0	1610	13	1597

图 5-5 国内疫情数据省市两级列表展示

5.1.3.4 搜索省份疫情信息

使用 watch 监视属性方法。监听 keyword 值的变化。代码如下：

```

onsearch() {
  this . active = this . searchindex
  // console . log ( this . searchindex )
  if ( this . search == '' || this . search . length < 2 ) {
    return
  }
  if ( this . searchindex == 1 ) {
    let data = this . search
    if ( this . search . charAt ( this . search . length - 1 ) == '省' ) {
      data = data . substr ( 0 , data . length - 1 )
    }
    let ping = py . chineseToPinYin ( data )
    this . active = this . searchindex = 2
    this . seacon = this . navdata [ 2 ] . search
    this . $router . push ( {
      name : 'Province' ,

```

```
        query:{
            name:ping,
            cname:data
        }
    })
    this . search=''
    return
}
if ( this . active ==2){

    let data=window.citydatas . cities
    // console . log( data )
    let se=this . search
    if ( this . search . charAt( this . search . length -1) == '市'){
        se=this . search . substr (0, this . search . length -1)
    }
    // console . log( se )
    for ( let i=0;i<data . length ;i++){
        // console . log( data [ i ] )
        if ( data [ i ] . cityName == se){
            // console . log( data [ i ] )
            this . citydata [ 0] . 累计确诊 = data [ i] . confirmedCount
            this . citydata [ 0] . 已治愈 == data [ i] . curedCount
            this . citydata [ 0] . 当前确诊计数 = data [ i] . currentConfirmedCount
            this . citydata [ 0] . 死亡 = data [ i] . deadCount
        }
    }
    // console . log( yhis . citydata )
    this . modalShow = true
    this . search = ''
}
}
```

5.1.4 疫情新闻页面

5.1.4.1 新闻轮播图

使用 axios 请求新闻信息数据。代码如下：

```
async getNewsdata(){
    this . axios({
        url : 'http :// api . tianapi . com / txapi / ncov / index ' ,
        method : 'GET' ,
        params : {
```

```

        key: '869941cd56fe09e14b255d12467651bd'
      }
    }).then(res => {
      if (res.data.code == 200) {
        // console.log(res.data.newslst [0].desc)
        this.show=false
        this.nCovInfo.info1=res.data.newslst [0].desc.note1;
        this.nCovInfo.info2=res.data.newslst [0].desc.note2;
        this.nCovInfo.info3=res.data.newslst [0].desc.note3;
        this.nCovInfo.info4=res.data.newslst [0].desc.remark1;
        this.nCovInfo.info5=res.data.newslst [0].desc.remark2;
        this.nCovInfo.info6=res.data.newslst [0].desc.remark3;
        this.newsDatalist=res.data.newslst [0].news
        console.log( this.newsDatalist )
      }
    });
  }
}

```

封装新闻数据 newsDatalist。代码如下：

```

data() {
  return {
    slide : 0,
    show:true,
    sliding : null,
    nCovInfo:{},
    newsDatalist :[],
    srcdata :[3,4,5]
  }
},

```

轮播图左右按钮。代码如下：

```

onSlideStart ( slide ) { // 轮播图左右按钮
  this.sliding = true
},
onSlideEnd(slide) { // 轮播图左右按钮
  this.sliding = false
},

```

封装数据引入轮播图。代码如下：

```

<b-carousel
id="carousel-1"
v-model="slide"
: interval ="4000"
controls

```

```

no-hover-pause
indicators
background="#ababab"
img-width="1024"
img-height="480"
style="text-shadow: 1px 1px 2px #333;"
@sliding-start="onSlideStart"
@sliding-end="onSlideEnd"
>
<b-carousel-slide
v-for="(item, index) in newsDatalist"
:key="index"
:caption="item.infoSource"
:text="item.title"
:img-src="SlideImg(index)"
></b-carousel-slide>
</b-carousel>

```



图 5-6 轮播图

5.1.4.2 新闻信息

axios 请求到数据后渲染到前端页面。代码如下：

```

<b-card-group deck>
<b-card :title="item.title" v-for="(item, index) in newsDatalist" :key="index" img-alt="
Image" :img-src="ImgSrcData(index)" img-top>
<b-overlay :show="show" rounded="sm" style="height: 100%;" v-if="show"></b-overlay>
<b-card-text>
<p class="mb-0">{{item.summary}}</p>
</b-card-text>
<template v-slot: footer>
<small class="text-muted">{{item.infoSource}} 发布于 {{item.pubDateStr}}</small>
</template>

```

```

<div style="text-align: center;">
<b-button :href="item.sourceUrl" size="lg" variant="outline-secondary">了解详情</b-
  button>
</div>
</b-card>

</b-card-group>
    
```

点击了解详情跳转进入央视新闻客户端新闻主页。如图 5-8 所示



图 5-7 跳转至央视新闻

5.1.5 更多资讯页面

5.1.5.1 音乐播放器

使用 vue-oplayer 音乐插件，创建歌单。代码如下：

```

getmusiclist () {
  this .axios ({
    url : 'http ://139.9.163.138:3000/ song/ url?id=33894312',
    
```

```

        method: 'GET',
        params: {

        }
    }).then(res => {
        // console.log(res.data.data[0].url)
        // this.audio.url=res.data.data[0].url
        // console.log(this.audio.url)
        // this.audio.url=require(res.data.data[0].url)

    })
}
},
created() {
    this.getmusiclist()
}

```

使用 jsDelivr 加速优化从仓库获取歌曲的速度。操作如下：

```
https://cdn.jsdelivr.net/gh/你的用户名/你的仓库名@发布的版本号/文件路径
```

经由 jsDelivr 加速请求获得歌曲文件的部分代码如下：

```

data() {
    return {
        audio: {
            src: 'https://cdn.jsdelivr.net/gh/devil-trigger/sdn@master/ncov/music/G.E.M
                .%20%E9%82%93%E7%B4%AB%E6%A3%8B%20-%20%E5%B9%B3%E5%87%
                A1%E5%A4%A9%E4%BD%BF.mp3',
            cover: 'https://p1.music.126.net/5zs7IvmLv7KahY3BFzUmrq
                ==/109951163635241613.jpg?param=300y300', // prettier-ignore
            // lrc: 'https://cdn.jsdelivr.net/gh/devil-trigger/sdn@master/ncov/music/G.E
                .M.%20%E9%82%93%E7%B4%AB%E6%A3%8B%20-%20%E5%B9%B3%E5
                %87%A1%E5%A4%A9%E4%BD%BF.lrc',
        },
        list: [
            {
                src: 'https://cdn.jsdelivr.net/gh/devil-trigger/sdn@master/ncov/music/华晨宇
                    -你要相信这不是最后一天(Live).mp3',
            },
            {
                src: 'https://cdn.jsdelivr.net/gh/devil-trigger/sdn@master/ncov/music/.mp3',
            },
            {
                src: 'https://cdn.jsdelivr.net/gh/devil-trigger/sdn@master/ncov/music (With
                    hope,with me).mp3',
            }
        ]
    }
}

```



```

        },
        ]
    };
},

```



图 5-8 音乐播放器

5.1.5.2 辟谣信息

用 axios 请求天行 api 数据。代码如下：

```

getrumourdata() {
    this . axios({
        url : "http :// api . tianapi . com / txapi / rumour / index",
        method : 'GET',
        params : {
            key : '869941cd56fe09e14b255d12467651bd'
        }
    }).then(res=>{
        if (res . data . code == 200){
            this . rumourlist = res . data . newslst
        }
    })
}

```

```
        console.log(this.rumourlist)
    }
})
```

通过“实时辟谣”按钮打开侧边栏，显示辟谣信息。代码如下：

```
data(){
    return {
        rumourlist :[]
    }
},
```

通过“实时辟谣”按钮打开侧边栏显示辟谣信息。代码如下：

```
<b-button v-b-toggle.sidebar-right size="lg" v-b-tooltip.hover title="点击打开侧边栏">
    实时辟谣</b-button>

<b-sidebar id="sidebar-right" title="不传谣,不信谣" right shadow width="15rem">
<div class="px-3 py-2">
<b-card class="rumourstyle" :title="item.title" :img-src="item.imgsrc" img-alt="Image"
    img-top v-for="(item,index) in rumourlist" :key="index">
<b-card-text>
{{item.desc}}
<h2>{{item.explain}}!</h2>
</b-card-text>
<template v-slot:footer>
<small class="text-muted">{{item.author}}发布于{{item.date}}</small>
</template>
</b-card>
<b-img src="https://cdn.jsdelivr.net/gh/devil-trigger/sdn/ncov/7.jpg" fluid thumbnail
    style="border-radius: 22px;"></b-img>
</div>
</b-sidebar>
```

5.2 系统测试

5.2.1 全球数据页面

5.2.1.1 查看地图不同地区现有确诊信息

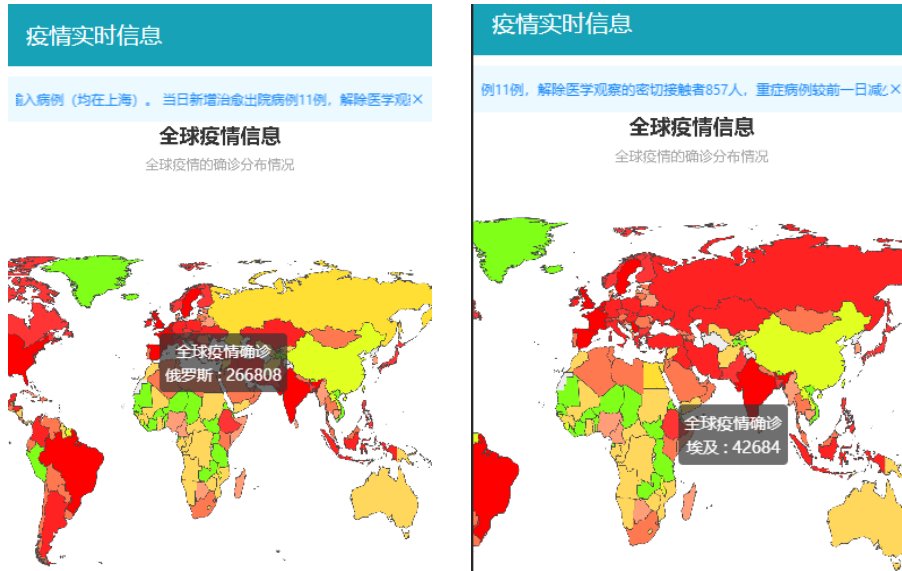


图 5-9 不同地区现有确诊信息

5.2.1.2 查看地图不同地区累计确诊信息

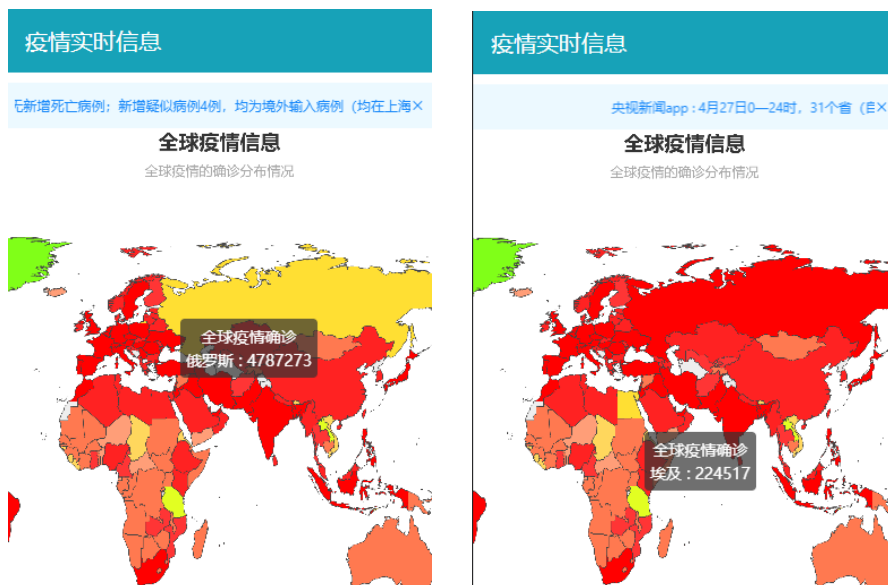


图 5-10 不同地区累计确诊信息

5.2.1.3 列表查看各国家感染数

全球病例				
省份	增加确诊	累计确诊	死亡	治愈
美国	52046	32927091	587384	25521913
印度	360806	17997113	201187	14817240
巴西	72140	14441563	395022	12815119
法国	30322	5595403	103762	341406
俄罗斯	0	4725252	107167	4352649
土耳其	43301	4710582	39057	4167263
英国	2697	4425259	127705	14567
意大利	10398	3981512	119912	3413451
西班牙	7665	3496134	77855	3186967
德国	21625	3338564	82326	2960900
阿根廷	25495	2905172	62599	2563223
哥伦比亚	17578	2804881	72235	2616821
波兰	0	2768034	65897	2467036
伊朗	0	2438193	70532	1907190
墨西哥	3592	2333126	215547	1856543
乌克兰	9590	2102011	45204	1666920
秘鲁	0	1768186	60013	1714995
印度尼西亚	0	1651794	44939	1506599
捷克	3256	1626033	29141	1542244
安提瓜和巴布达	0	1232	32	1014
毛里求斯	0	1206	17	1055
不丹	0	1053	1	951
法属波利尼西亚	0	773	0	528
钻石公主号	0	712	13	699
特克斯和凯科斯群岛	0	668	5	572
红宝石公主号	0	620	10	0
老挝	0	511	0	49
坦桑尼亚	0	509	21	183
直布罗陀	0	315	0	273
文莱	1	224	3	215
多米尼克	0	173	0	171
格林纳达	0	159	1	155
斐济	0	109	2	65
马提尼克岛	0	66	1	0
圣基茨和尼维斯	0	44	0	44
梵蒂冈	0	27	0	15

图 5-11 列表展示各国家疫情数据

5.2.2 全国数据页面

5.2.2.1 国内不同省份疫情信息展示

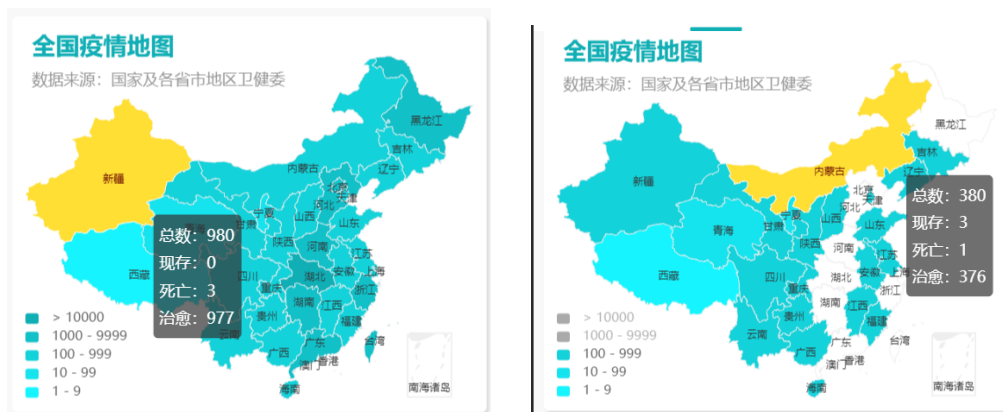


图 5-12 国内不同省份疫情数据

5.2.2.2 勾选指定感染人数区块不显示

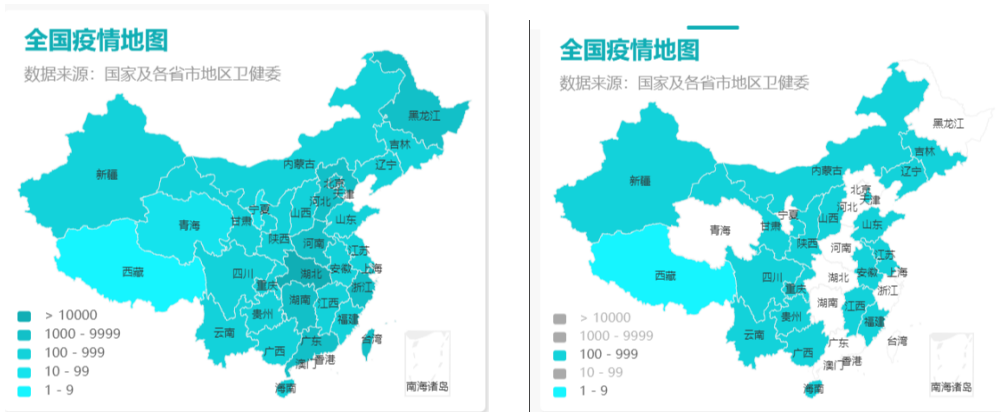


图 5-13 国内不同省份疫情数据

5.2.2.3 国内省市二级列表疫情信息展示

中国病例				
省份	现存确诊	累计确诊	死亡	治愈
湖北	5	68157	4512	63640
武汉	5	50362	3869	46488
孝感	0	3518	129	3389
黄冈	0	2907	125	2782
荆州	0	1580	52	1528
鄂州	0	1394	59	1335
随州	0	1307	45	1262
襄阳	0	1175	40	1135
黄石	0	1015	39	976
宜昌	0	931	37	894
荆门	0	928	41	887
咸宁	0	836	15	821
十堰	0	672	8	664
仙桃	0	575	22	553
天门	0	496	15	481
恩施州	0	252	7	245
潜江	0	198	9	189
神农架林区	0	11	0	11
香港	160	11748	209	11379
广东	40	2334	8	2286
上海	62	1972	7	1903
黑龙江	0	1610	13	1597
浙江	11	1332	1	1320
温州	0	504	1	503
杭州	0	181	0	181
宁波	0	157	0	157
台州	0	147	0	147
境外输入	11	113	0	102
金华	0	55	0	55
嘉兴	0	46	0	46
绍兴	0	42	0	42
省十里丰监狱	0	36	0	36
丽水	0	17	0	17
衢州	0	14	0	14
湖州	0	10	0	10
舟山	0	10	0	10
河北	0	1317	7	1310

图 5-14 省市二级列表疫情信息

5.2.3 疫情新闻页面

5.2.3.1 轮播图滚动并显示新闻摘要

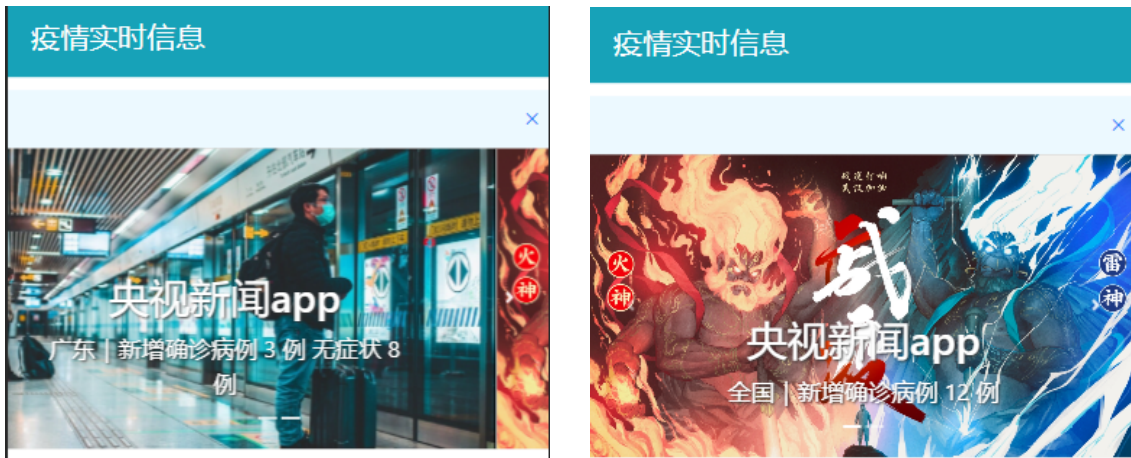


图 5-15 轮播图

5.2.3.2 新闻信息获取与显示



图 5-16 新闻

5.2.4 更多资讯页面

5.2.4.1 实时辟谣信息



图 5-17 实时辟谣信息

5.2.4.2 歌曲播放器



图 5-18 音乐播放器

第6章 总结和展望

6.1 本文总结

从毕业设计开题到现在即将完成毕业论文的编写,标志着大学生活的最后一次课程接近尾声。倾注到最后一课的心血现在回首仍是历历在目。确定研究方向和实现功能是第一道拦路虎。从最初的多人组队完成任务到之后要求各自为题单人完成,两次选题我都会心悸高估自己的水平,又担心工作量不足无法顺利完成答辩。万事开头难,之后的开发实现过程虽缓慢但却也是踏踏实实的在向前迈进,这些点滴的成就给了我信心,经过努力终于在最后完成了大部分模块的实现和展示。下面对本设计及实现过程进行总结。

该项目的选取是基于近两年来新冠肺炎疫情仍在肆虐,民众需要第一时间能获取清晰、直观、可靠的数据这一需求应运而生。我完成的项目主要依托于 Vue.js 前端开发框架将包括权威机构的公开报告,世卫组织,各国官方通知,国家卫生健康委员会,省市卫生健康委员会,省市政府,香港获得的国内外疫情数据相结合,澳门和台湾官员以及 ECharts 可视化库确保了项目显示数据的可靠性和及时性。我又通过研读《最简单的图形与最复杂的信息:如何有效建立你的视觉思维》^[1]学习了如何通过多种数据形式表达来更清晰高效的传递信息,这也是本项目的难点与重点:面对纷杂的数据,我究竟要如何将它们以简洁美观,全面高效的方式展现给用户。最后得出的结论是结合地图、表格、折线图等多种形式,将大部分功能的开发体现在国内省市疫情信息的展现上;国外疫情数据简化,作为国内疫情信息的补充。这个决定也让我的开发周期变得更短,有更多时间可以投入对系统的优化当中。再接下来我会具体列出开发中遇到的问题及我注意到的值得优化的地方。

首先需要解决的是跨域问题。我的解决方式是修改 config.js 中的配置文件,将跨域设置为 true。再者是开发过程中我曾调用过源于 github 库的音频文件,发现数据获取速度出奇的慢,最后我通过使用 jsDelivr 加 github 搭建的 CDN 来进行加速,优化资源的获取速度、降低网络堵塞、提高用户访问响应速度。最后遇到的是一个小插曲,却引起了我对从多方面采集数据的重视。论文完成快过半的时候,我请求的某个 api 接口地址突然从数据网站消失,导致“国内疫情信息”模块大面积无法获取到数据。迫不得已只好通过新浪、百度等渠道重新请求或爬取数据,所幸最后完成了原定功能。通过这次设计和开发的实时疫情显示系统,我对 Vue.js 框架下的诸如 vue-aplayer、跨域实现等此前一知半解的知识点得到了充分的学习,在对数据的采集、整理、展示方面也累计了不少经验。

6.2 未来展望

本疫情实时展示系统的开发虽已接近尾声,但在后期优化中仍发现了不少可以改进甚至颠覆的地方:

1. 在全球数据页面中可以将侧边栏的列表式信息安置在地图下面,由此可以加大页面空间的利用面积,使数据表能容纳更多数据。

2. 本次实现该系统脱离了后端,仅依靠前端对网络上的信息进行搜集与展示。这虽然使本项目变得轻量且简洁,但是无疑降低了用户的交互体验,减少了项目的寿命。日后应该搭配后端与数据库将疫情信息作为本地信息存储后再输出。

3. 可以添加更多综合性数据。例如给用户查询城市来往政策的渠道、用户在地图上选择省市时提供当地气温和空气污染指数等。

4. 可以结合 SpringBoot 加 Vue.js 实现一个数据标注系统。让用户能在地图上自由标注信息,加强用户与系统间的交互。

大学生活转瞬即逝,日后再看这份毕业设计应该会感触良多,但却始终不会忘记力争上游,不断学习的激情。

附录 A 源码

参考文献

- [1] 刘博文. 深入浅出 Vue.js[M]. [出版地不详]: 清华大学出版社, 2019.
- [2] 朴灵. 深入浅出 Node.js[M]. [出版地不详]: 人民邮电出版社, 2013.
- [3] 王大伟. ECharts 数据可视化: 入门实战与进阶[M]. [出版地不详]: 机械工业出版社, 2020.

致谢

四年之期已到。愕然回首，容颜虽变，我却依旧是那个朝着理想飞奔的读书人；我的举止依旧会留有恩师谆谆教导的痕迹；我的身旁依旧会有年龄相仿的“老伙计”一起开怀大笑共叙人生。感谢这四年你们的关照与付出。

在做毕业设计的这几个月里，首先我要感谢宋喜佳老师对我的关照与帮助。没有您的付出，我可能会在做毕设写论文上走不少弯路。您在日常的教学严谨却不乏活泼，深得同学们的喜欢和尊敬。

其次，我要感谢辅导员陈洁老师和张展骞老师。有幸能成为两位辅导员老师一段时间的班干部或好帮手我感到十分开心，相较于平日里两位老师对我学习和生活的关照，这些也只是鸡毛蒜皮般的事，我希望能通过未来的成长与成功来回应两位老师无私的付出。

同样，我也要感谢我的同学。这四年我们共同成长进步，感谢有你们的支持与鼓励，谢谢你们和我一起度过人生中的四年时光，即便以后各奔东西，也不会忘记当初一起奋斗的模样。

最后，我还要感谢我的父母。你们对我默默的支持是我人生前进路上最大的动力，养育之恩何以回报。