

<DKU REPORT>

2조 [Corona Website] 최종 보고서



과목 :	실무중심 산학협력 프로젝트 1
담당 :	송인식 교수님
제출 :	2021-06-15
조원 :	김형욱 32161269 윤시준 32177457 최원석 32144792

목차

1. 프로젝트 개요

1.1 프로젝트 제안 배경	4
1.2 기능 기요	4

2. 사용자 가이드

2. 메뉴 분류와 사이트 이용 가이드	5
2.1 Main Page.....	6
2.2 이용안내	9
2.3 통계 정보	10
2.3.1 국내 누적 확진자 통계.....	10
2.3.2 지역별 확진자 증가 추세.....	10
2.3.3 지역별 확진자 비교 통계.....	11
2.3.4 일정 확진자 초과일 조회.....	12
2.3.5 지역별 확진자 지도.....	12
2.4 방역 정보	13
2.5 정부 정책	14

3. 개발자 가이드

3. 개발자 가이드 개요	15
3.2 기본 플랫폼 구성 안내	16
3.2.1 [01] - Github Pages	16
3.2.2 [02] - Github Jekyll	17
3.2.3 [03] - Footer Loading & SNS share 버튼.....	18
3.2.4 [04] - Domain	19
3.3 통계 기능 안내	20
3.1 [05] - Data Crawling	20
3.2 [06] - Danfo.JS	21
3.3 [07] - heroku	25
3.4 기타 기능 안내.....	30
3.4.1 [08] – 드롭다운식 상단메뉴	30
3.4.1 [09] - Detail / Summary	31
3.4.2 [10] - Image Scale	32
3.4.3 [11] - Image Slider	33
3.4.4 [12] – 댓글 기능	34

4. 설계와 디자인

4.1 모듈 간 인터페이스	35
4.2 Data Source	36
4.3 Diagram	37
4.4 Risk Assessment	38
4.5 Coding Style	38
4.6 Team Structure	39

5. TESTING

5.1 testing	40
-------------------	----

6. Reference

6.0 Corona Core	41
6.1 Github link	41
6.2 Data Source	41
6.3 Corona Reference	41
6.4 domainin & API	41

7. 분석 및 소감

7 분석 및 소감.....	42
----------------	----

1. 프로젝트 개요

1.1 제안 배경

2020년 1월 20일 국내 첫 코로나 확진자 발생 이후 확진자 수가 계속해서 늘어남에 따라 관련 정보 제공 서비스들이 점차 생겨났습니다. 현재 수많은 국내외 민·관 사이트들이 각기 자리잡고 있습니다.

이러한 웹사이트들은 누적 확진자 수, 당일 확진자 수 등을 알려주며 사용자는 그 외에도 각종 지역별 피해 상황과 방역 정보 등을 확인할 수 있습니다. 이를 통해 현재 코로나 정보 사이트 접속자들은 각자의 목적에 따라 방역 지침 확인이나 사회현상 분석 등의 용도로 이런 서비스들을 다양하게 사용하고 있습니다.



1.2 기능 개요

그리고 이에 대한 저희 플랫폼이 제공할 기능을 간단히 살펴보겠습니다. 기존의 코로나 사이트들은 기본적인 국내 누적 확진자, 시도별 확진자 등에 대한 통계와 그래프를 제공합니다. 저희는 이들 사이트와 같이 저희 사이트가 데이터 제공 플랫폼의 필수적인 기능을 갖추도록 기본적인 사항들을 구현할 예정입니다.

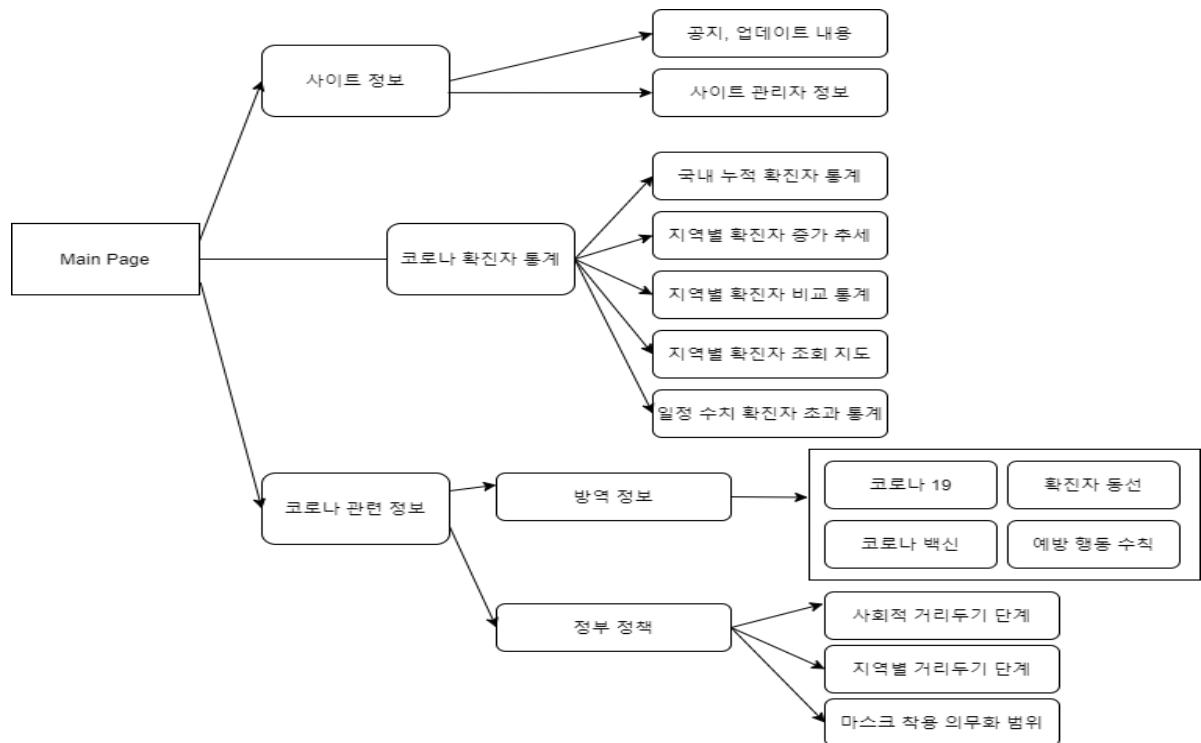
그 외에는 기존 플랫폼의 미진한 점과, 예상되는 이용자 요구사항을 분석해 저희가 독자적인 플랫폼이 될 만한 컨텐츠 목록들을 추가로 구성할 예정입니다. 세부적인 사항은 이후 사용자 가이드의 사이트 메뉴 설명에서 정리합니다.

2. 사용자 가이드

2. 사이트 메뉴 분류와 사용자 이용 가이드

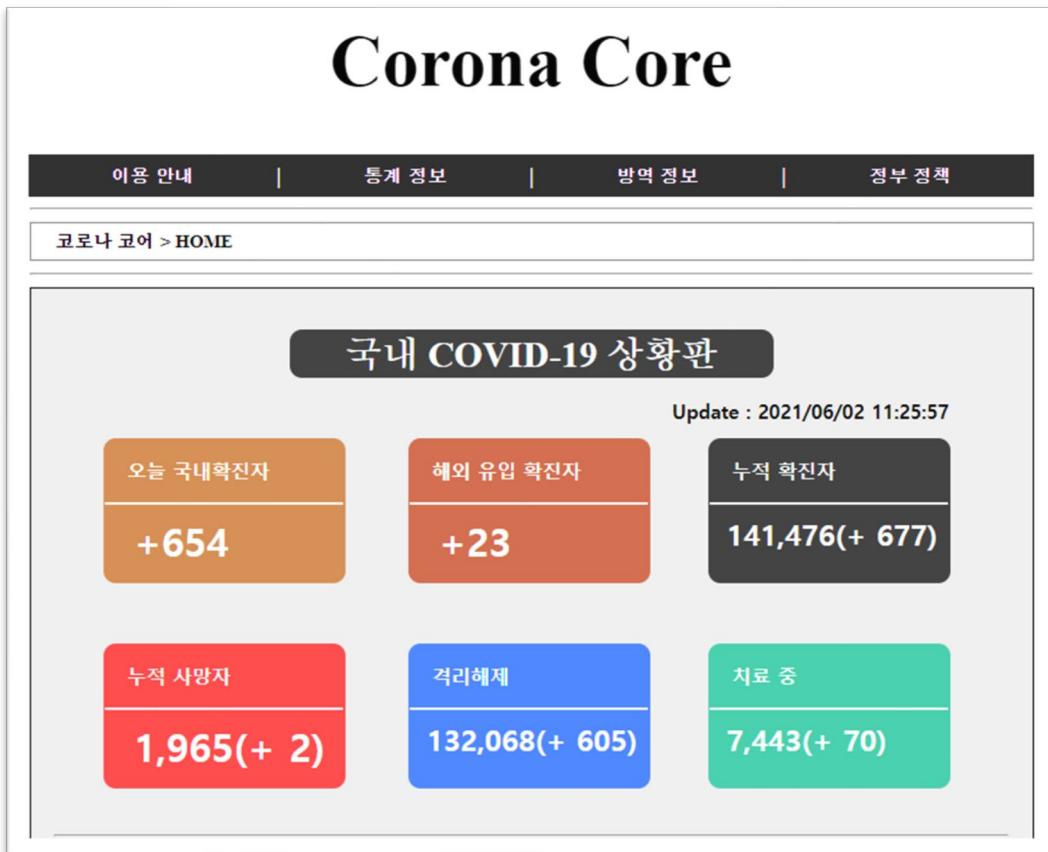
1. 페이지 분류

Corona Core는 정적 웹 개발 기술을 바탕으로 설계된 코로나 관련 정보 사이트입니다. Corona Core의 상단 메뉴바는 Drop Down 방식으로 작동하며 크게 사이트 운영 정보, 2)코로나 확진자 데이터 통계 3)코로나 방역 정보, 4)코로나 관련 정부 정책메뉴로 구성되어 있습니다.



1.1 Main Page

먼저 사이트 로고와 각 메뉴로 이동할 수 있는 상단 드롭다운 메뉴,
그리고 실시간 코로나 확진자를 확인할 수 있는 실시간 상황판이 마련되어 있고,



밑에는 각종 코로나 정보 사이트로 이동할 수 있는 이미지 슬라이더가 작성되어 있습니다.

이미지 슬라이더에서는 질병관리본부, 외교부, CDC 센터, 존스홉킨스 CSSE, WHO 등 각종 코로나 정보 사이트로 이동할 수 있습니다.



이어서 각종 사이트 메뉴를 빠르게 살펴보고 이동할 수 있는 메뉴 정보창과 링크가 있고,

<h3>통계 정보</h3>	
국내 누적 확진자 통계 지역별 확진자 증가 추세 지역별 확진자 비교 통계 특정 수치 초과일 조회 지역별 확진자 숫자 조회	확진자 수치를 기간별로 조회할 수 있습니다 특정 일자의 지역 별 감염 증감을 보여줍니다. 특정 지역의 감염 양상을 보여줍니다. 입력한 확진자 수치를 넘긴 날짜를 검색할 수 있습니다. 대한민국 지도에서 지역별 확진자 숫자를 조회할 수 있습니다
<h3>방역 정보</h3>	
코로나 19 코로나 백신 확진자 동선 예방 행동수칙	코로나 바이러스 정보를 제공합니다 코로나 백신 정보를 제공합니다 지역별 확진자 동선 링크를 제공합니다 기본적인 예방 행동 수칙을 확인할 수 있습니다
<h3>정부 정책</h3>	
사회적 거리두기 정책 지역별 거리두기 단계 마스크 착용 의무화범위	전체 OR 단계별 사회적 거리두기 정책을 확인할 수 있습니다. 지역별 거리두기 단계를 확인할 수 있습니다. 마스크 착용 방침과 위반 시 과태료를 확인할 수 있습니다.

사이트 하단 FOOTER에는 간략한 주의사항과
누르면 Corona Core 사이트를 SNS로 공유할 수 있는 SNS 공유 버튼이 마련되어 있습니다.

- 주의 사항 -

민간이 취합한 집계이므로 공식적인 근거 자료로 활용될 수 없으며
본 사이트의 정보 사용에서 발생하는 문제의 책임은 전적으로 사용자에게 있습니다.

corona.core.kr@gmail.com



 Facebook

공유하려면 Facebook 계정에 로그인하세요.

이메일 또는 전화:

비밀번호:

로그인

계정을 잊으셨나요?

새 계정 만들기



1.2 이용 안내 메뉴

이용 안내 메뉴는 사이트 운영에 관련한 두 하위 메뉴를 포함하고 있습니다.

공지사항 : 사이트를 사용하는 사용자에게 공지를 전달하는 페이지입니다.

관리자가 직접 마크다운 언어를 통해 관리하고 있습니다.

The screenshot shows a '공지사항' (Notice) page. At the top, there is a header with tabs: '이용 안내' (selected), '통계 정보', '방역 정보', and '정부 정책'. Below the header, a breadcrumb navigation shows '이용 안내 > 공지사항'. The main content area is titled 'Corona Core 공지사항'. It features a post by 'Corona Core' with the title '2021-05-27 도메인 수정' and a timestamp '마지막 업데이트 2021-05-12'. There is also a 'Follow' button. Below this, there is a section titled 'Recent posts' with another post titled '2021-05-13 test'.

About Us : 사이트 관리자에 대한 간단한 정보(e-mail, 개발 의도 등)을 포함하는 페이지입니다.

The screenshot shows an 'About Us' page. At the top, there is a header with tabs: '이용 안내' (selected), '통계 정보', '방역 정보', and '정부 정책'. Below the header, a breadcrumb navigation shows '이용 안내 > About Us'. The main content area is titled 'About Us'. It includes a paragraph about the project being a result of a software engineering capstone project at Dankook University. It also mentions the use of public data APIs and the source code being open-sourced. There is a section for contact information with email addresses for Kim Hyung-Ok, Yoon Si-jun, Choi Won-seok, and a general contact email for the project.

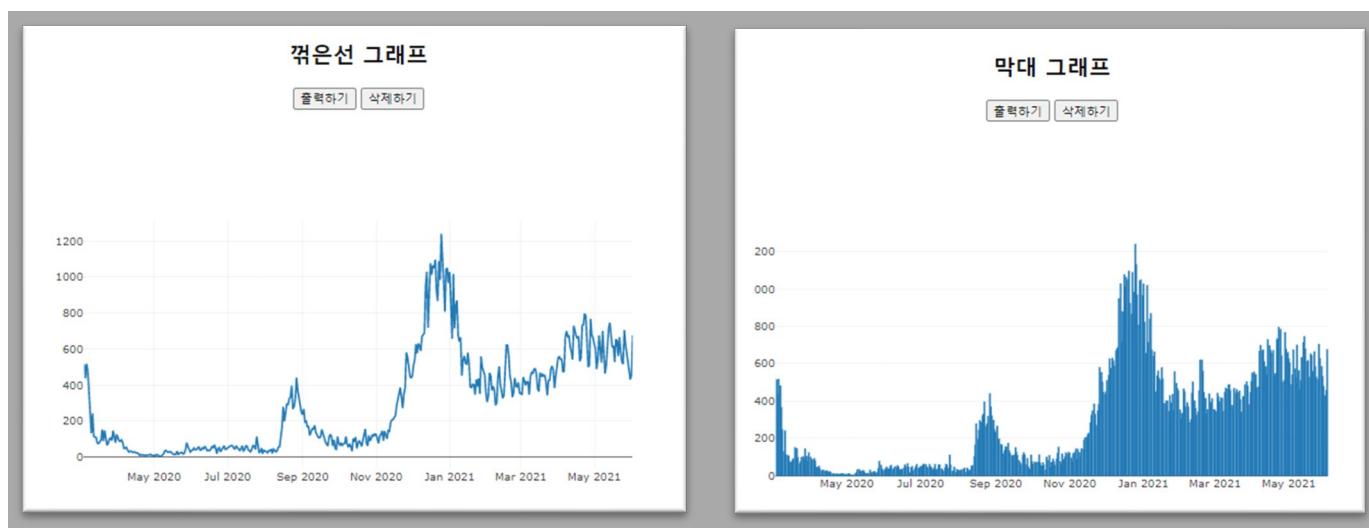
1.3 통계 정보

공공데이터 API에서 따온 데이터를 가공한 각 그래프를 확인할 수 있습니다.

1.3.1 코로나 전체 기간 일일 확진자 그래프

통계 정보
국내 누적 확진자 통계
지역별 확진자 증가 추세
지역별 확진자 비교 통계
특정 수치 초과일 조회
지역별 확진자 숫자 조회

확진자 수치를 기간별로 조회할 수 있습니다. 일반적인 확진자 현황 그래프와는 달리 코로나 확진자를 기간별로 나누어 조회할 수 있습니다. 사용자는 원하는 차트를 선택해 출력하거나 다운로드 할 수 있습니다.



1.3.2 날짜 & 지역별 확진자 증가 추세

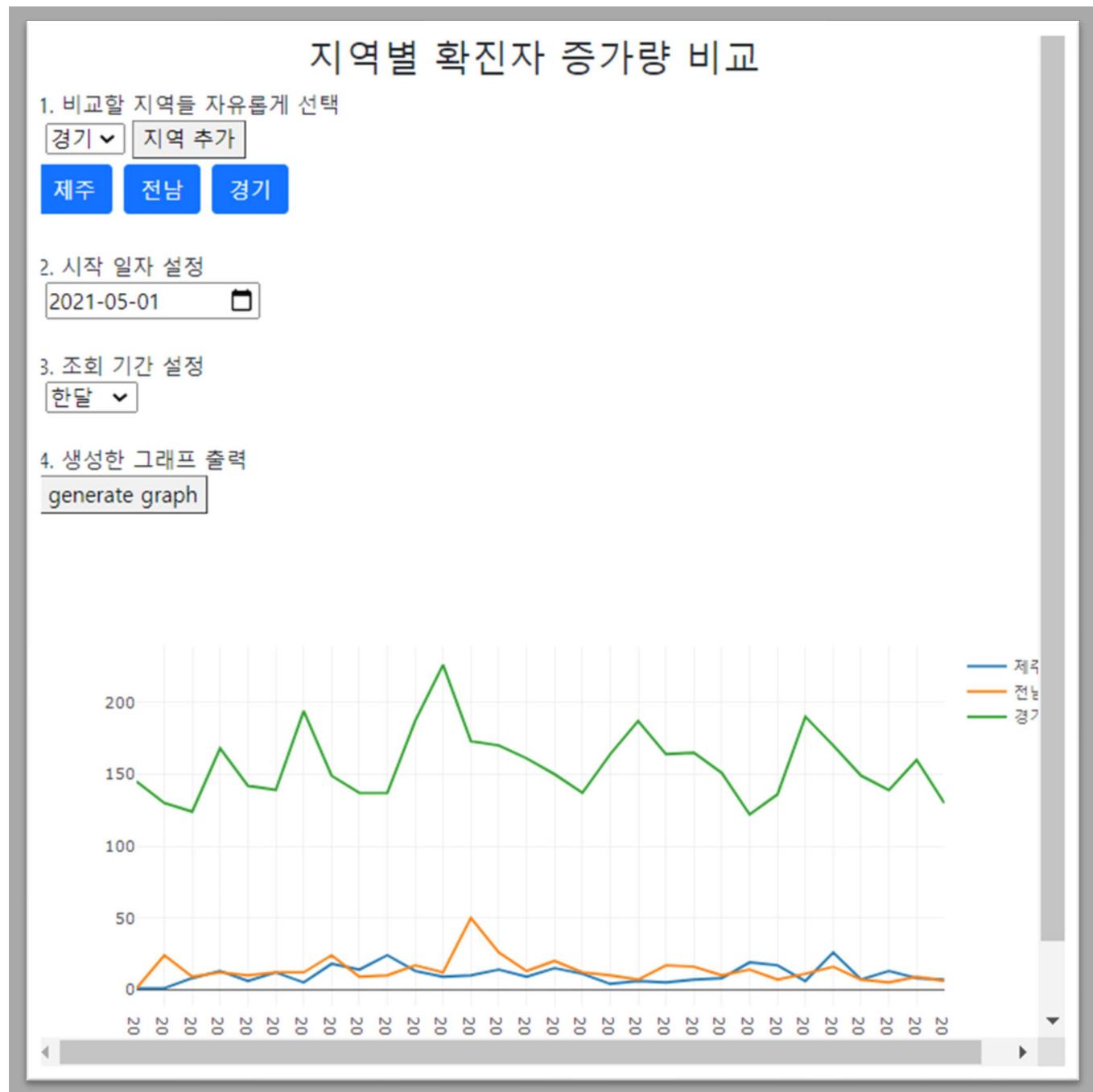
특정 일자의 지역별 감염증감을 보여줍니다. 원하는 그래프를 선택하면 사이트에서 해당 기록에 해당하는 파이 차트와 바 차트를 출력해 줍니다.

- 원형 그래프 : 지역별 관리 시스템 수준을 확인할 수 있습니다.
- 막대 그래프 : 정확한 지역별 확진자 수치를 확인할 수 있습니다.



1.3.3 지역별 확진자 비교 통계

특정 지역의 지역별 감염 양상을 보여줍니다. 지역별 관리 실태와 해당 지역의 원활한 방역 상황의 적절성을 확인할 수 있습니다.



1.3.4 일정 확진자 조회

날짜가 아닌 확진자 수치를 기준으로 데이터를 출력합니다. 입력된 숫자보다 많은 확진자가 발생한 일자를 출력하여 표로 나타냅니다.

400

특정 코로나 확진자 수치 조회 일자 검색하기

sum	date
516	20-03-04
438	20-03-05
518	20-03-06
483	20-03-07
441	20-08-27
581	20-11-26
555	20-11-27
503	20-11-28
450	20-11-29
438	20-11-30
451	20-12-01
511	20-12-02

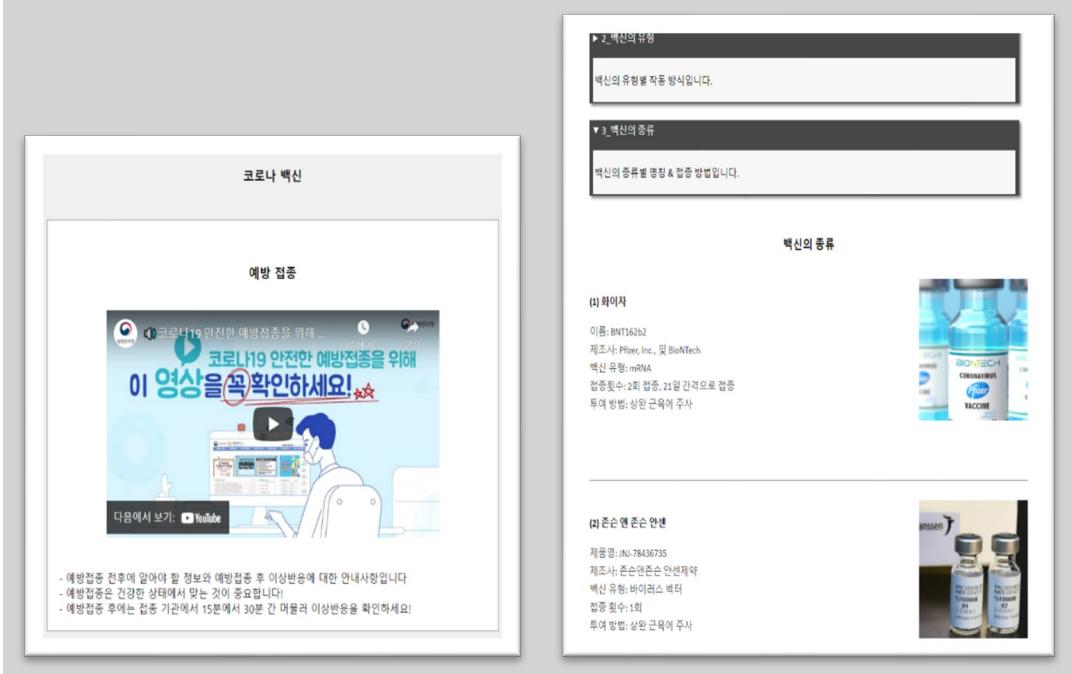
1.3.5 지역별 확진자 지도

전국 지도에서 지역별 확진자 수치를 조회할 수 있습니다. 원하는 날짜와 조사할 지역을 선택하면 해당 날짜의 지역별 확진자 수치를 조회할 수 있습니다.





1.4 방역 정보

구분	내용
코로나 19	코로나 바이러스의 기본 정보와 병원생물학 정보를 전달합니다.
코로나 백신	<p>백신 접종방법, 백신 원리, 백신 유형, 백신 종류를 설명입니다.</p>  <p>[Corona core 코로나 백식 page]</p>
확진자 동선	지자체별 확진자 동선 정보를 제공합니다. 사용자가 각 링크를 통해 소속된 지자체 정보를 확인할 수 있는 인터페이스를 제공합니다.
예방 행동수칙	<p>시민들이 기본적으로 코로나에 대응해 취할 수 있는 예방 행동수칙을 보여줍니다.</p> <p>이미지 파일과 각종 링크가 사용됩니다.</p>

1.5 정부 정책

사회적 거리두기 단계

전체적인 사회적 거리두기 정책과 단계별 사회적 거리두기 정책을 확인할 수 있습니다.

- 전체 사회적 거리두기 정책 : 이미지로 제공, 이미지 확대 기능 사용
- 단계별 다중이용시설 관리 정책 : 텍스트로 제공, 접기/펴기 기능 사용

사회적 거리두기 정책

전체 사회적 거리두기 정책

클릭 시 이미지가 확대됩니다

1단계	1.5단계	2단계	2.5단계	3단계
생활방역	지역적 유형 단계	지역 유형 급속 전파, 전국적 확산 개시	전국적 유형 본격화	전국적 대유행
생활 속 거리 두기	지역적 유형 개시	지역 유형 급속 전파, 전국적 확산 개시	전국적 유형 본격화	전국적 대유행
통상적인 밤역 및 의료체계의 감당 가능한 범위 내에서 유행 통제 중	특정 권역에서 의료체계의 통상 대응 범위를 위협하는 수준으로 1주 이상 유형 지속	1.5단계 조치 후에도 지속적 유행 증가 양상을 보이며, 유행이 전국적으로 확산되는 조짐 관찰	의료체계의 통상 대응 범위를 초과하는 수준으로 전국적 유행이 1주 이상 지속 또는 확대	전국적으로 급격하게 환자가 증가하여 의료체계 불리 위험 예측

단계별 다중이용시설 관리 정책

클릭 시 박스가 펼쳐집니다

▶ 1단계
생활 속 거리 두기

▶ 1.5단계
지역적 유형 개시

▶ 2단계
지역 유형 급속 전파, 전국적 확산 개시

마스크 착용 의무화 범위

- 마스크 착용 방침과 위반 시의 과태료를 확인할 수 있습니다.

3. 개발자 가이드

다음은 코로나 코어 사이트의 유지보수와 확장을 위한 개발자 메뉴얼입니다. 저희 코로나 코어 사이트는 기본적으로 다음 목록에 적힌 기술이 포함되어 있고, 개발자들이 아래 자료를 활용해 사이트를 관리할 수 있도록 메뉴얼을 작성했습니다. 사용된 기술들 중 간단한 기술은 개략적으로 살펴보고 중요한 기술은 자세한 설명을 담아 설명할 수 있도록 했습니다.

사이트에 사용된 기술 목록

< 기본 플랫폼 기술구성 >

- 01. Github Pages : 사이트 구성하는 깃허브 클라우드 서비스
- 02. Github jekyll : 메인 홈페이지와 업데이트 내역 부분의 관리자 페이지
- 03. Footer & SNS btn : Footer 섹션과 SNS 공유 버튼 안내
- 04. 도메인 등록 : CoronaCore.site 도메인 등록과 갱신 관련 안내

< 통계 기능 안내 >

- 05. 데이터 크롤링 : 중본 사이트에서 코로나 Data 추출
- 06. Danfo.JS(5) : 공공 데이터에서 코로나 Data 추출 & 가공 (코로나 통계 그래프)
- 07. heroku : 관리자가 heroku를 통해 서버의 공공API Data의 최신 Data 범위 조정

< 기타 기능 안내 >

- 08. 드롭다운 메뉴 : 사이트 상단 메뉴에서 마우스 호버 시 펼쳐지는 세로 메뉴 바
- 09. Detail/Summary : 기본적인 코로나 정보에 대한 접기 펼치기 기능 (JS 포함)
- 10. image scale 조정 : 이미지 파일 확대 – 축소 (JS 포함)
- 11. 이미지 슬라이더 : index 페이지에서 여타 코로나 사이트로 이동하는 이미지 슬라이더

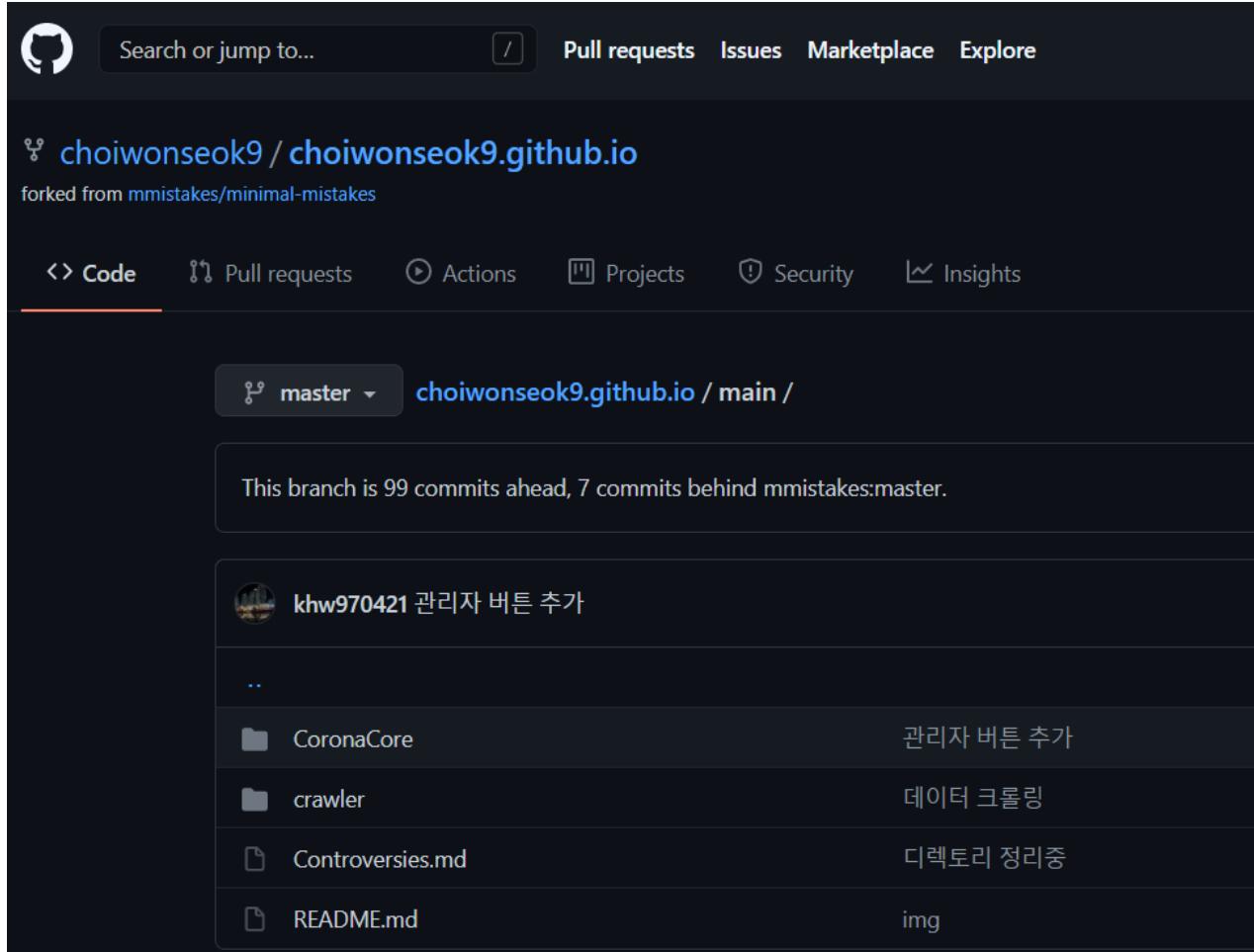
< 기본 플랫폼 기술구성 >

01. Github Pages

깃허브 Pages는 깃헙 레포지토리에 작성된 웹개발 프로젝트를 배포해주는 클라우드 서비스입니다. 저희 Corona Core는 이를 통해 저희 정적 웹 서비스를 배포중이며, 만약의 상황을 대비해 다른 호스팅 서비스인 네틀리파이도 대안으로 준비하고 있습니다. CoronaCore은 현재 CoronaCore.site라는 도메인을 부여해 github.io라는 url이 주소창에 뜨진 않습니다. 사용되고 있는 깃허브 주소는

<https://github.com/choiwonseok9/choiwonseok9.github.io>이며

GITHUB PAGES 는 기본적인 웹개발 레포지토리를 바로 웹 호스팅해주는 무료 서비스이므로, 다른 개발자들이 새로 이에 대한 사용 방법을 배울 필요가 없고, 개발에 참여할 시 collaborator 자격을 부여받고 github desktop 등 도구로 간편히 CoronaCore 의 개발 업무를 수행할 수 있습니다.



02. Github Jekyll

Jellyll은 markdown으로 작성된 문서를 HTML로 변환하여 웹사이트를 구축할 수 있도록 돋는 Static Website generator이며 Ruby로 작성되어 있습니다. 저희 코로나 코어에서는 메인 홈페이지 우측에서 관리자가 관리자 페이지를 게시하여 업데이트 내역을 보여줄 수 있도록 작성하였습니다. 기본적으로 GitHub Pages에서 HTML, CSS, Javascript만으로도 웹사이트를 구축할 수 있으나 markdown을 사용하여 웹사이트를 generate하기 위해 Jellyll을 사용하기로 결정하였습니다. 로컬 환경에서 Jellyll을 사용하여 웹사이트를 작성/테스트하고 GitHub repository에 웹 리소스를 push하면 매우 간단히 Website를 Hosting할 수 있는 구조가 되겠습니다.

기본적인 사용법은 깃허브 튜토리얼을 통해 배울 수 있으며,

사용되는 파일 디렉토리와 코드는 저희 CoronaCore 최상단 디렉토리에서 확인할 수 있습니다.

The screenshot shows a GitHub repository named 'CoronaCore'. On the left, there's a tree view of files and folders. On the right, a list of commits is displayed, each with a timestamp and a brief message. The commits include updates to layouts, posts, and assets, along with configuration files like .gitignore, travis.yml, and Gemfile.

Commit Message	Date
Create 2021-05-27-update-0527post.md	6 days ago
Update .sass	21 days ago
Update .assets	21 days ago
Add .main	8 hours ago
Update .gitignore	21 days ago
Update .travis.yml	21 days ago
Create CNAME	6 days ago
Update Gemfile	21 days ago
Update LICENSE	21 days ago
Update README.md	22 days ago
Update Rakefile	21 days ago
Update .config.yml	21 days ago
Update banner.js	21 days ago
Add default config and home index page	5 years ago
Update package.json	21 days ago
Update staticmap.yaml	21 days ago

The screenshot shows the GitHub profile page for 'Corona Core'. It features a header with tabs for '이용 안내', '통계 정보', '방역 정보', and '정부 정책'. Below the header, there's a section for '공지사항' (Announcements) with a link to 'Corona Core 공지사항'. A sidebar on the right lists recent posts and a sidebar on the left shows a navigation menu.

The screenshot shows the live website for 'Corona Core'. The top navigation bar includes '이용 안내', '통계 정보', '방역 정보', and '정부 정책'. The main content area features a 'HOME' button and a large '국내 COVID-19 상황판' (Domestic COVID-19 Situation Board). This board displays real-time statistics: '오늘 국내 확진자 +654', '해외 유입 확진자 +23', '누적 확진자 141,476(+ 677)', '누적 사망자 1,965(+ 2)', '격리 해제 132,068(+ 605)', and '치료 중 7,443(+ 70)'. To the right, a sidebar lists recent posts with titles like '2021-05-27 도메인 설정', '2021-05-13 test', '2021-05-11 업데이트', '2021-05-03 업데이트', and '2021-05-02 업데이트', each with a brief description and a 'Follow' button.

03. Footer loading & SNS share 버튼

Footer 와 SNS 공유 버튼에 관한 내용입니다. 우선 각 메뉴 파일에서 Head 에 작성한 다음 Jquery 코드로 인해

```
<script src="//code.jquery.com/jquery-1.11.0.min.js"></script>
<script type="text/javascript">
    $(document).ready( function() {
        $("#footer").load("../footer.html");
        $("#header").load("../header.html");
    });
</script>
```

바디 부분에서 사이트 최하단의 `<div id="footer"></div>` 자리에 Footer 가 불러오기됩니다.

개발자는 이로써 모든 사이트 메뉴 하단에 똑같은 footer 파일을 작성할 필요가 없고 footer.html 하나의 단일 파일만 관리하면 됩니다. 참고로 footer에 표현되는 SNS 쉐어 버튼은 다음과 같은 자바스크립트 코드와 html 버튼 부분으로 나누어집니다.

+) SNS share button

```
<!-- SNS 공유용 주소 연결 용 -->
<script type="text/javascript" async>
    var url_default_ks = "https://story.kakao.com/share?url=";
    var url_default_fb = "https://www.facebook.com/sharer/sharer.php?u=";
    var url_default_tw_txt = "https://twitter.com/intent/tweet?text=";
    var url_default_tw_url = "&url=";
    var url_default_band = "http://band.us/plugin/share?body=";
    var url_route_band = "&route=";
    var url_default_naver = "http://share.naver.com/web/shareView.nhn?url=";
    var title_default_naver = "&title=";
    var url_this_page = location.href;
    var title_this_page = document.title;
    var url_combine_ks = url_default_ks + url_this_page;
    var url_combine_fb = url_default_fb + url_this_page;
    var url_combine_tw = url_default_tw_txt + document.title + url_default_tw_url + url_this_page;
    var url_combine_band = url_default_band + encodeURI(url_this_page) + '%0A' + encodeURI(title_this_page) + '%0A' + '&route=tistory.com';
    var url_combine_naver = url_default_naver + encodeURI(url_this_page) + title_default_naver + encodeURI(title_this_page);
</script>
<br>
```


 - 주의 사항 -

 민간이 취합한 집계이므로 공식적인 근거 자료로 활용될 수 없으며
 본 사이트의 정보 사용에서 발생하는 문제의 책임은 전적으로 사용자에게 있습니다.


```
<br>
corona.core.kr@gmail.com<br><br>
<!-- SNS share buttons -->
<div style="width: 100%; text-align: center; margin-bottom: 64px;">
<!-- Facebook share button -->
<a href="" onclick="window.open(url_combine_fb, '', 'scrollbars=no, width=600, height=600'); return false;">

</a>
<!-- Twitter share button -->
<a href="" onclick="window.open(url_combine_tw, '', 'scrollbars=no, width=600, height=600'); return false;">

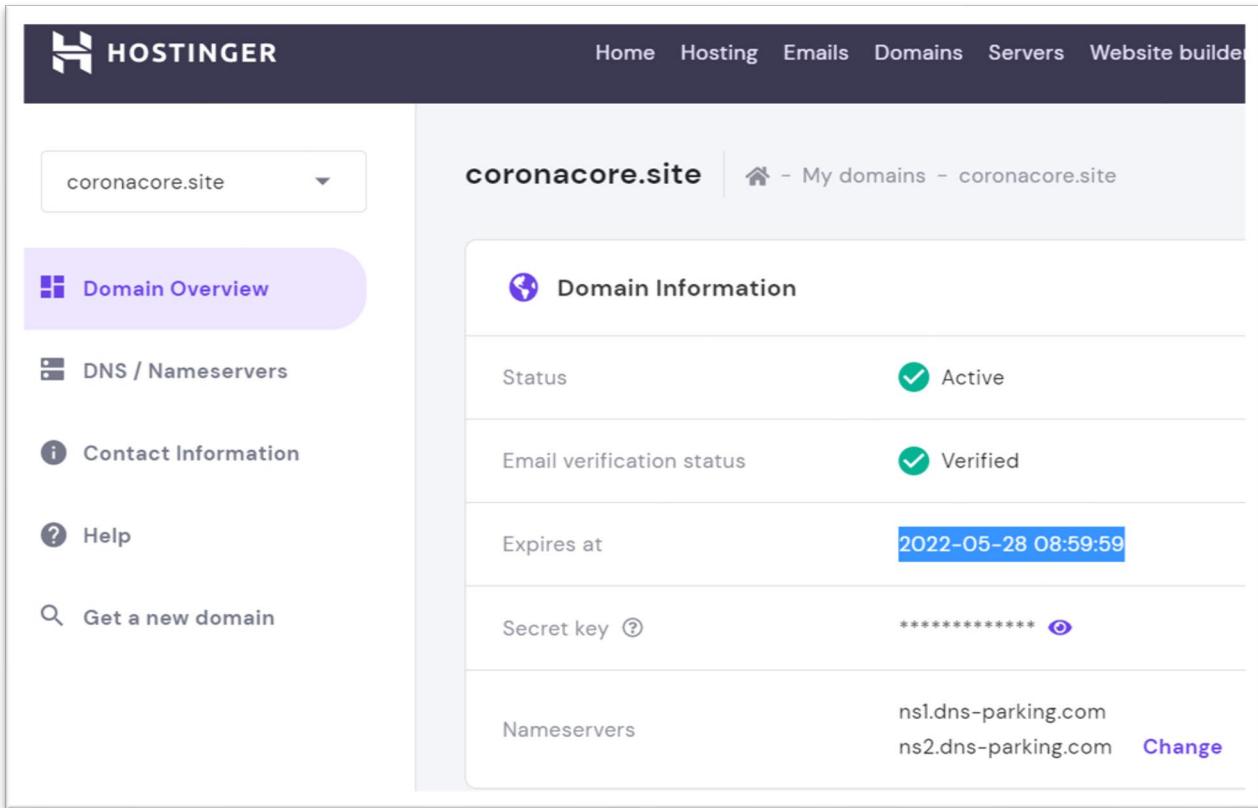
</a>
<!-- Naver share button -->
<a href="" onclick="window.open(url_combine_naver, '', 'scrollbars=no, width=600, height=600'); return false;">

</a>
<!-- Kakao story share button -->
<a href="" onclick="window.open(url_combine_ks, '', 'scrollbars=no, width=600, height=600'); return false;">

</a>
</div>
<br>
```

04. 도메인 등록

다음은 저희가 구매한 도메인 CoronaCore.site에 대한 내용입니다. 도메인 구매는 1년치를 구매한 상태이며 만료 한달 전인 2022년 6월 28일 전에 도메인을 구매하면 계속해서 사용이 가능합니다.



The screenshot shows the Hostinger domain management interface. On the left, there's a sidebar with a dropdown menu set to 'coronacore.site'. Below it are links for 'Domain Overview' (which is highlighted in purple), 'DNS / Nameservers', 'Contact Information', 'Help', and 'Get a new domain'. The main content area is titled 'coronacore.site' and shows 'My domains - coronacore.site'. Under 'Domain Information', it lists the following details:

Field	Value
Status	Active
Email verification status	Verified
Expires at	2022-05-28 08:59:59
Secret key	***** ⓘ
Nameservers	ns1.dns-parking.com ns2.dns-parking.com Change

사용되는 주요 DNS record는 다음과 같습니다.

A	@	0	185.199.111.153	14400	Delete	Edit
A	@	0	185.199.110.153	14400	Delete	Edit
A	@	0	185.199.109.153	14400	Delete	Edit
A	@	0	185.199.108.153	14400	Delete	Edit
CNAME	www	0	coronacore.site	14400	Delete	Edit
CNAME	autodiscover	0	autodiscover.mail.hostinger.co m	300	Delete	Edit
CNAME	choiwonseok9.github.io	0	coronacore.site	300	Delete	Edit

05. 데이터 크롤링

CoronaCore은 보건복지부의 코로나바이러스감염증19 사이트의 실시간 Data를 추출해 코로나 코어 사이트로 크롤링하고 있습니다. 사용되는 기술은 파이썬 beautiful soup 라이브러리로

<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

Node.py 파일을 실행하면 date 파일의 실시간 상황판 데이터와 stage 파일의 지역별 사회적 거리두기 단계 값이 바뀌게 됩니다. 작동 방식은 정규표현식 입력값에 따라 원하는 값이 추출되는 방식입니다.

date.html	2021-05-27 오후 5:46	Chrome HTML Do...	1KB
date2.html	2021-06-02 오전 11:25	Chrome HTML Do...	1KB
dead_corona.html	2021-06-02 오전 11:25	Chrome HTML Do...	1KB
korea_corona.html	2021-06-02 오전 11:25	Chrome HTML Do...	1KB
mini_corona.html	2021-06-02 오전 11:25	Chrome HTML Do...	1KB
node.py	2021-05-28 오후 4:12	Python File	8KB
stage.html	2021-06-02 오전 11:25	Chrome HTML Do...	1KB
tit.html	2021-06-02 오전 11:25	Chrome HTML Do...	1KB
tit2.html	2021-06-02 오전 11:25	Chrome HTML Do...	1KB
world_corona.html	2021-06-02 오전 11:25	Chrome HTML Do...	1KB

```
# 해외 유입 확진자
ultra_fine_dust = data2[1].find('span',{'class':'data'}).text
print("해외 유입 확진자", ultra_fine_dust)

#누적확진자
data3 = soup.find('ul',{'class':'liveNum'})    #
data4 = data3.findAll('li')
mini = data4[0].find('span',{'class':'num'}).text  #index1부터 시작한 이유는 (누적)을 없애기 위해서
print("누적확진환자",mini)
mini2 = data4[0].find('span',{'class':'before'}).text
print("전일대비",mini2)

#사망자
dead = data4[3].find('span',{'class':'num'}).text  #index1부터 시작한 이유는 (누적)을 없애기 위해서
print("사망자",dead)
dead_today = data4[3].find('span',{'class':'before'}).text
print("오늘 사망자",dead_today)
```

06. Danfo.JS 와 통계 그래프

Danfo.JS 로 공공API 데이터에서 추출한 코로나 자료를 어떻게 가공했는가에 대한 내용입니다.

현재 CoronaCore 두번째 메뉴인 submenu_2에 코로나 통계 그래프에 관한 모든 파일이 들어있는데요,

Img_Slider	2021-05-27 오후 5:46	파일 폴더
submenu_1	2021-05-27 오후 5:46	파일 폴더
submenu_2	2021-06-01 오전 4:37	파일 폴더
submenu_3	2021-05-27 오후 5:47	파일 폴더
submenu_4	2021-05-27 오후 5:46	파일 폴더
footer.html	2021-05-27 오후 5:46	Chrome HTML Doc... 3KB
header.html	2021-05-27 오후 5:46	Chrome HTML Doc... 0KB
hello.html	2021-05-27 오후 5:46	Chrome HTML Doc... 1KB
index.html	2021-06-01 오전 4:56	Chrome HTML Doc... 25KB
style.css	2021-05-27 오후 5:46	CSS 스타일시트 ... 8KB

우선 파일 구성은 Graph1~5의 각 통계 그래프 메뉴에서

g1~g4의 각 통계 그래프와 /Kmap 의 g5 그래프를 IFRAME으로 로딩하는 방식이며,

로딩된 각각의 그래프는 다시 폴더 내 JS파일과 Data/Date 파일들을 참조해 그래프를 출력하고 있습니다.

.idea	2021-05-27 오후 5:46	파일 폴더
Data	2021-05-27 오후 5:46	파일 폴더
js	2021-06-01 오전 4:38	파일 폴더
kMap	2021-06-01 오전 4:31	파일 폴더
g1_Daily_Total_Increase.html	2021-06-01 오전 7:08	Chrome HTML Docu... 2KB
g2_PieChart_BarChart.html	2021-06-01 오전 7:27	Chrome HTML Docu... 2KB
g3_CompareCity.html	2021-06-01 오후 3:29	Chrome HTML Docu... 2KB
g4_Exceeded_Dates.html	2021-06-01 오전 4:31	Chrome HTML Docu... 1KB
graph_style.css	2021-05-27 오후 5:46	CSS 스타일시트 문서 1KB
graph1.html	2021-06-01 오전 7:28	Chrome HTML Docu... 5KB
graph2.html	2021-06-01 오전 4:57	Chrome HTML Docu... 5KB
graph3.html	2021-06-01 오후 3:12	Chrome HTML Docu... 5KB
graph4.html	2021-06-01 오전 7:22	Chrome HTML Docu... 5KB
graph5.html	2021-06-01 오후 3:31	Chrome HTML Docu... 5KB
Readme.md	2021-05-31 오후 12:24	Markdown File 3KB

1) 누적 확진자 통계 (g1_Daily_Total_Increase.js)

```
lineButton.addEventListener( type: 'click',startLineChart);
barButton.addEventListener( type: 'click',startBarChart);
cancelLine.addEventListener( type: 'click',cancelLineChart);
cancelBar.addEventListener( type: 'click',cancelBarChart);

async function load () {
  const get_date = [];
  const get_sum = [];

  const datas = await Set_Date() Promise<any>
    .then(values=>values.map(date =>dfd.read_csv(` ${url}${date}.csv`)) Promise<any>
    .then(data=>Promise.all(data));

  datas.forEach(data => {
    get_sum.push(data.body.items.item_incDec.data[data.body.items.item_incDec.data.length - 1]);
    get_date.push(data.body.items.item_createDt.data[0].slice(2,10));
  })

  let df_sum = new dfd.DataFrame({sum:get_sum},{index:get_date}); //df_sum은 Series 형태이므로 DataFrame 형태로 변환

  document.querySelector( selectors: '.but').addEventListener( type: 'click', listener: ()=>downloadCSV(df_sum.index_arr,df_sum.data));

  return df_sum;
  //df_sum.plot('bar').bar();
}

}


```

차트하나에 꺾은선그래프와 막대그래프를 선택하여 확인 할 수 있고 해당 그래프의 Set_Date.js 코드의 Set_Date 함수를 통해 필요한 csv 파일을 가져와 더한 값을 계산해 보여줍니다.

2) 지역별 확진자 증가 추세 (g2_PieChart_BarChart.js)

```
const datas = await Set_Date() Promise<any>
  .then(values=>values.map(date =>dfd.read_csv(` ${url}${date}.csv`)) Promise<any>
  .then(data=>Promise.all(data));

let endDate = datas.datas.length-1].body.items.item_createDt.data[0].slice(0,10);

document.querySelector( selectors: '#range').setAttribute( qualifiedName: 'min', value: 1583247600);
document.querySelector( selectors: '#range').setAttribute( qualifiedName: 'max', value: new Date(endDate).getTime() / 1000);
document.querySelector( selectors: '#range').setAttribute( qualifiedName: 'step', value: 86400);

document.querySelector( selectors: '#range2').setAttribute( qualifiedName: 'min', value: 1583247600);
document.querySelector( selectors: '#range2').setAttribute( qualifiedName: 'max', value: new Date(endDate).getTime() / 1000);
document.querySelector( selectors: '#range2').setAttribute( qualifiedName: 'step', value: 86400);

$(document).ready(function() {
  $('#range').change(function(){

    var val = $(this).val();
    let result = Unix_timestamp(val);
    document.querySelector( selectors: '#div1').innerHTML= result;
    dfd.read_csv(` ${url}${result.replace( searchValue: /-/gi, replaceValue: '').slice(2,8)}.csv`)
    .then(
      function(data) {
        const incDec_Length_Except_Sum = data.body.items.item_incDec.data.length-1;
        const gubun_Length_Except_Sum = data.body.items.item_gubun.data.length-1;

        let df = new dfd.DataFrame({
          Price: data.body.items.item_incDec.data.slice(0,incDec_Length_Except_Sum), //표의 맨 아래 합계를 제거한 내용들
          Location : data.body.items.item_gubun.data.slice(0,gubun_Length_Except_Sum),
          Type: data.body.items.item_gubun.data.slice(0,gubun_Length_Except_Sum)
        })
        document.querySelector( selectors: '.but').addEventListener( type: 'click', listener: ()=>downloadCSV(df.Location.data,df.Price.data));
        df.plot("plot_div").pie({ values: "Price", labels: "Type" })
      }
    )
    .catch(() => {
      alert('저장된 데이터 이외의 날짜를 클릭했습니다.');
    })
  })
})



}


```

Install Dependencies
From package.json

바 슬라이더를 이용하여 드래그 시 원하는 날짜의 도시별 확진자 비율을 원형그래프 또는 막대그래프로 확인 할 수 있고 그래프를 삭제할 수 있습니다.

3) 지역별 확진자 비교 통계 (g3_CompareCity.js)

```
addButton.addEventListener( type: 'click' ,addEventListener1);
button.addEventListener( type: 'click',getServerData);

function getServerData() {
  async function load() {
    try {
      let cityIndex =[],getDate=[];
      let totalContent={};

      const datas = await Promise.all(setDate()).map(date => {
        return dfd.read_csv(`$url${date}.csv`)
      })
      //모든 데이터 값 읽어오기

      addCityIndex(datas,cityMember,cityIndex); //해당 지역을 가진 index 찾기
      checkIndex(datas,cityIndex,totalContent); //해당 지역을 가진 인덱스마다의 객체의 프로퍼티인 배열에 값을 넣는다
      addCityDate(datas,getDate); //해당 지역을 가진 index를 통해 날짜 배열인 getDate저장

      const df_sum1 = new dfd.DataFrame(totalContent, {index: getDate}); // totalContent와 날짜인 getDate를 통해 df_sum 생성
      df_sum1.plot('plot').line(); //결과 출력

      // 해야할 부분 : 추가한 도시 버튼 클릭시 삭제기능 구현 , 전반적 코드 정리 필요
    }
    catch(err){
      alert('데이터가 존재하지 않거나 잘못되었습니다.')
    }
  }
  load();
}
```

비교할 지역을 다양하게 추가할 수 있고 추가한 지역을 다시 클릭시 삭제기능을 구현하고 checkIndex 함수를 통해 중복된 지역을 방지하고 조회한 날짜만큼의 도시들마다의 확진자 수 증가량을 비교 할 수 있습니다.

4) 특정 수치 초과일 조회 (g4_Exceeded_Dates.js)

```
async function load (value) {
  const get_date = [];
  const get_sum = [];
  try {
    const datas = await Set_Date();
    .then(values=>values.map(date => dfd.read_csv(`$url${date}.csv`)) 
    .then(data=>Promise.all(data));

    datas.forEach(data => {
      get_sum.push(data.body.items._item._incDec.data[data.body.items._item._incDec.data.length - 1]);
      get_date.push(data.body.items._item._createDt.data[0].slice(2, 10));
    })
    let newSum = [];
    let newDate = [];
    get_sum.forEach((sum, index : number ) => {
      if (sum > value) {
        newSum.push(sum);
        newDate.push(get_date[index]);
      }
    })
    const DataFrame = {'sum': newSum, 'date': newDate};
    const df = new dfd.DataFrame(DataFrame);
    df.plot('table').table();
  }
  catch(err){
    alert(err);
  }
}
```

특정 숫자를 입력 후 enter 시 특정 숫자(value)와 날짜마다의 확진자수(sum)을 비교하여 특정 숫자보다 많은 날짜와 확진자수를 표(table)를 통해 보여줍니다.

5) 지역별 확진자 숫자 조회 (korea.js)

```
let input = document.getElementById('input')
let Plot_Button = document.getElementById('plot');
let service;

Plot_Button.addEventListener('click', listener: function(){
    dfd.read_csv(`http://coronacore.site/main/CoronaCore/01_code/submenu_2/Data/Date/${input.value.replace(/-/gi,'').slice(2,8)}.csv`)
    .then(
        function(data) {
            const incDec_Length_Except_Sum = data.body_items_item_incDec.data.length-1;
            const gubun_Length_Except_Sum = data.body_items_item_gubun.data.length-1;

            df = new dfd.DataFrame({
                Price: data.body_items_item_incDec.data.slice(0,incDec_Length_Except_Sum), //표의 맨 아래 합계를 제거한 내용들
            })
        }

        service = df.Price.data;
        let isGunyeok = service.length == 18 ? 1 : 0;

        let j = document.getElementById('label-Jeju-do')
        let gn = document.getElementById('label-Gyeongsangnam-do')
        let gb= document.getElementById('label-Gyeongsangbuk-do')
        let jb = document.getElementById('label-Jeollabuk-do')
        let jn = document.getElementById('label-Jeollanam-do')
        let cb= document.getElementById('label-Chungcheongbuk-do')
        let cn= document.getElementById('label-Chungcheongnam-do')
        let gw= document.getElementById('label-Gangwon-do')
        let gg = document.getElementById('label-Gyeonggi-do')
        let sj = document.getElementById('label-Sejong')
        let us = document.getElementById('label-Ulsan')
        let dj= document.getElementById('label-Daejeon')
        let gj= document.getElementById('label-Gwangju')
        let ic= document.getElementById('label-Incheon')
        let dg= document.getElementById('label-Daegu')
        let bs= document.getElementById('label-Busan')
        let su= document.getElementById('label-Seoul')

        })
    )
}

Install Dependencies
```

원하는 날짜를 지정한 후 웹 브라우저 지도 내에서 해당 날짜의 도시를 마우스를 대면 해당 도시의 확진자 수를 알 수 있습니다.

07. HEROKU

Heroku는 개발자가 클라우드에서 애플리케이션을 완전히 구축, 실행 및 운영할 수 있도록 지원하는 서비스형 플랫폼(PaaS)입니다

관리자 모드의 작동을 위해 유동적으로 동작하는 서버가 필요하고 이를 위해 Node.js와 RestApi를 잘 처리 할 수 있는 express 프레임워크를 이용하여 코드를 작성 후 heroku를 이용하여 서버에 올려 필요한 요청에 따라 서버에서 값을 바꿀 수 있습니다. 현재는 app.js 형태로 관련된 파일들을 heroku 서버에 올려둔 상태입니다.

app.js

```
app.route('/api/covidDate')
  .get(async(req,res)=>{
    const result = {success:true};
    try{
      const json = await db.getData()
      result.data = json.date;
    }catch(err){
      result.success = false;
      result.err = err;
    }
    res.json(result);
  })
  .put(async(req,res)=>{
    const result = {success: true}
    const date = req.body.date;
    const idx = req.params.parent
    try {
      const json = await db.getData()
      json.date = date
      await db.setData(json)
    } catch (err) {
      result.success = false
      result.err = err
    }
    res.json(result)
  })
}
```

Set_Date.js

```
function requestApiAndGetData() {
  return fetch( input: 'https://coronaproject.herokuapp.com/api/covidDate' ).then(res => res.json()).then(obj => {
    |   return obj.data
  })
}

function calculateDateArray(end){
  let sdt = new Date(start); // 2020년 03월 04일 부터 시작
  let edt = new Date(end);
  let dateDiff = Math.ceil( x: (edt.getTime()-sdt.getTime())/(1000*3600*24));

  let Year,Month,Day;           // 각 날짜별 날짜 생성
  const date_array = [];        // 해당 필요부분 넣을 배열 생성
  for(let i=0;i<dateDiff;i++)   // 2020년 3월 4일부터 마지막 데이터 csv까지의 날짜의 차이에서 +1 한 값이 150 (ex 현재는 7월 31일까지)
  {
    sdt.setDate(sdt.getDate()+1) // 3월 4일 계산 후 하루씩 증가
    Year = sdt.getFullYear().toString().slice(2,4); // 2020년이 아닌 뒤의 두자리 수만 필요하므로 slice 사용
    Month = (sdt.getMonth()+1).toString().length==1? '0'+(sdt.getMonth()+1).toString() :(sdt.getMonth()+1).toString(); // 한자리 수 인경우 앞에 0을 붙인다.
    Day = sdt.getDate().toString().length==1? '0'+sdt.getDate().toString() : sdt.getDate().toString();
    date_array.push(Year+Month+Day); //합친 내용을 배열로 만들어 준다.
  }
  return date_array; //해당 배열을 반환한다.
}

async function Set_Date(){
  let data = await requestApiAndGetData();
  let date_array = await calculateDateArray(data);
  return date_array;
}
```

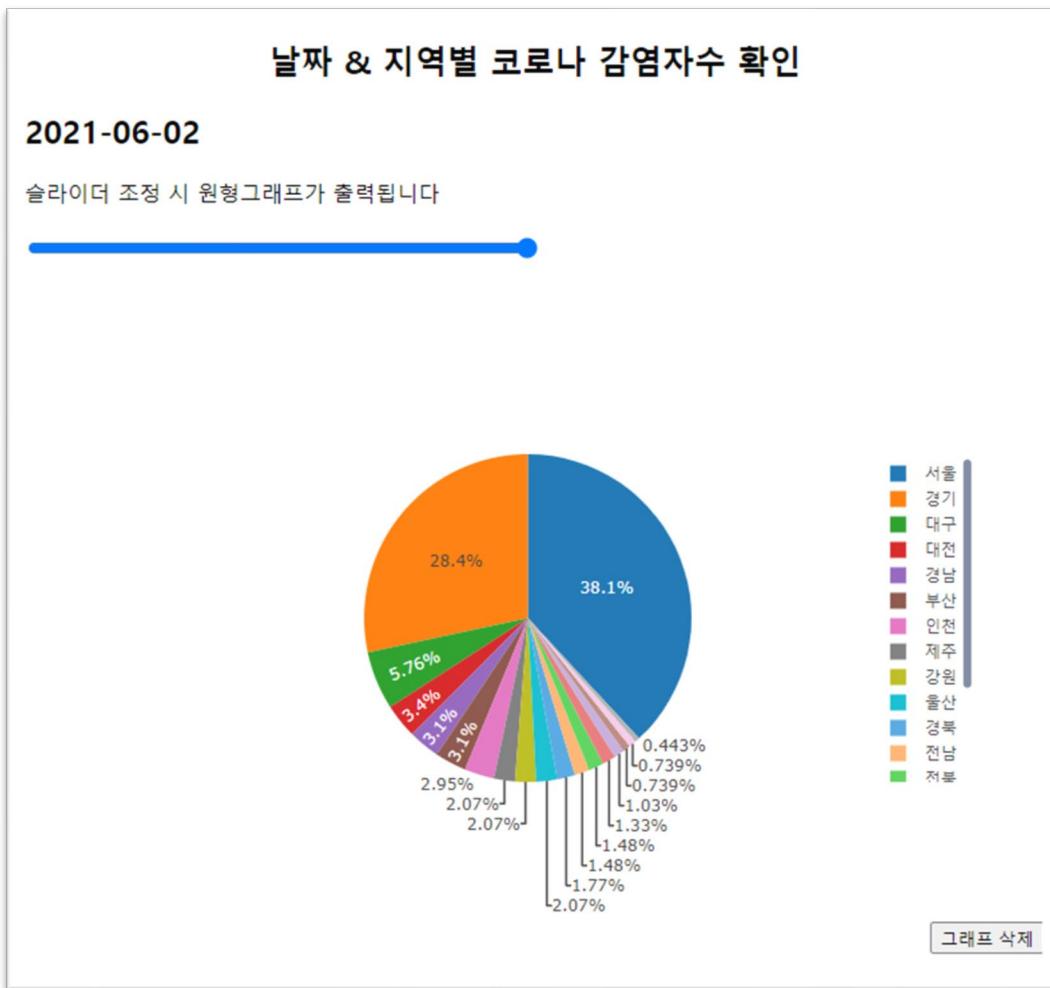
서버에 올렸던 <https://coronaproject.herokuapp.com/api/covidDate>에 간단히 날짜 데이터를 바꿀 수 있습니다.

fetch를 통해 서버에 요청하여 해당 마지막 날짜를 알아오는 requestApiAndGetData 함수를 통해 비동기로 받아온 data코드를 적용하여 calculateDateArray 함수를 통해 필요한 날짜 배열을 얻어올 수 있습니다.

Heroku 관리자 모드 작동시키기

```
← → ⌂ coronaproject.herokuapp.com/api/covidDate  
{"success":true,"data":"2021-06-02"}
```

현재 서버에 2021-06-02 데이터가 적용된 상태로



이를 통해 최대로 나올 수 있는 날짜는 2021-06-02 인데

관리자 버튼

coronacore.site 내용:

비밀번호를 입력해주세요

확인 **취소**

이용 안내 | 통계 정보 | 방역 정보 | 정부 정책

코로나 코어 > HOME

This screenshot shows a password input interface. At the top left is a '관리자 버튼' (Admin Button). Below it, the text 'coronacore.site 내용:' and '비밀번호를 입력해주세요' are displayed. A large input field is centered below the text. At the bottom right of the input field are two buttons: a blue '확인' (Confirm) button and a white '취소' (Cancel) button. Below the input field is a dark horizontal bar containing navigation links: '이용 안내', '통계 정보', '방역 정보', and '정부 정책'. At the very bottom is a footer bar with the text '코로나 코어 > HOME'.

여기서 관리자 버튼과 비밀번호를 누른 후 (현재 1234)

관리자 버튼

coronacore.site 내용:

업데이트에 적용할 날짜를 입력해주세요(yyyy-mm-dd)

확인 **취소**

이용 안내 | 통계 정보 | 방역 정보 | 정부 정책

코로나 코어 > HOME

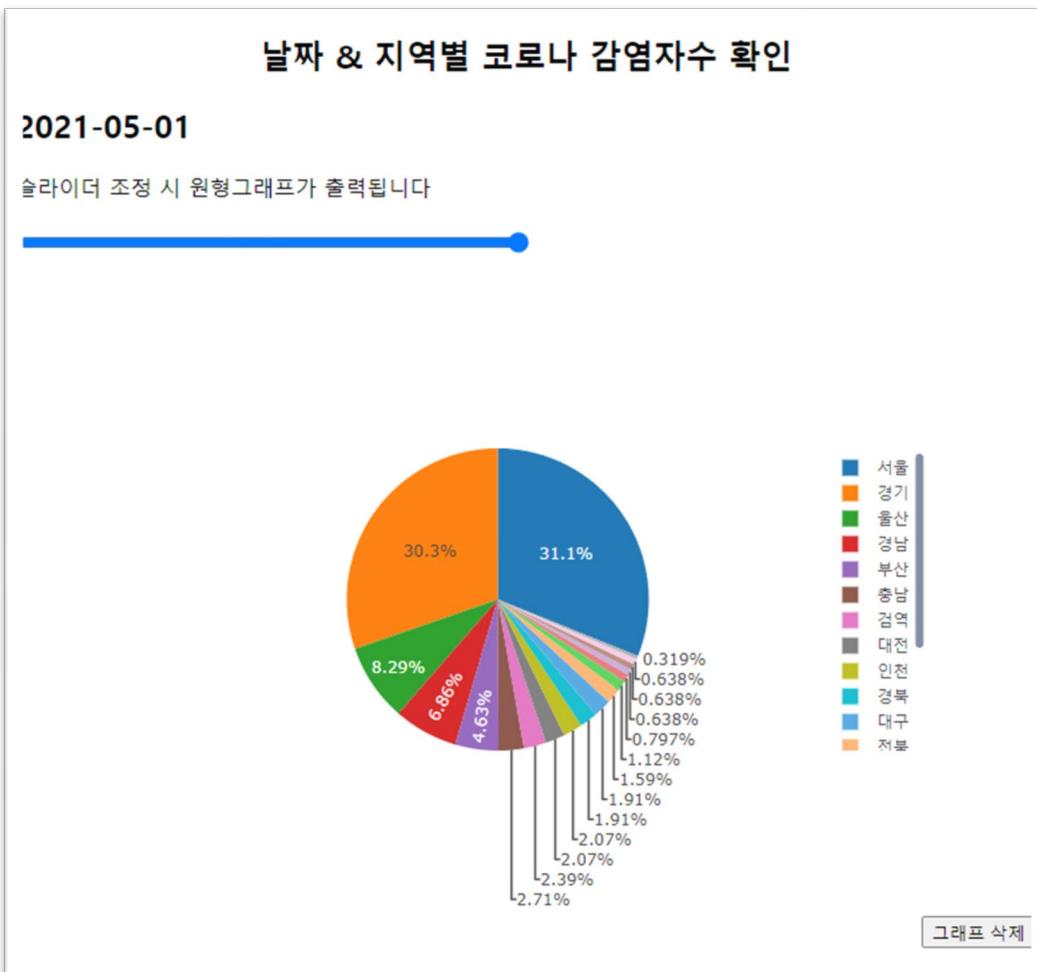
This screenshot shows a date input interface. At the top left is a '관리자 버튼' (Admin Button). Below it, the text 'coronacore.site 내용:' and '업데이트에 적용할 날짜를 입력해주세요(yyyy-mm-dd)' are displayed. A large input field is centered below the text, with a placeholder '(yyyy-mm-dd)'. At the bottom right of the input field are two buttons: a blue '확인' (Confirm) button and a white '취소' (Cancel) button. Below the input field is a dark horizontal bar containing navigation links: '이용 안내', '통계 정보', '방역 정보', and '정부 정책'. At the very bottom is a footer bar with the text '코로나 코어 > HOME'.

업데이트 될 날짜를 지정할 수 있습니다. (2021-05-01로 지정해 보겠습니다.)

```
← → ⌂ coronaproject.herokuapp.com/api/covidDate
{"success":true,"data":"2021-05-01"}
```

이를 통해 서버에서 최신 코로나 집계일 변수 값이 바뀌는 것을 볼 수 있습니다.

이어 마지막 슬라이더 날짜값도 2021-05-01로 관리자 모드를 통해 데이터가 추가되었을때 입력을 바꾸면 코드의 변수 수정 없이 해당 날짜에 관하여 처리를 진행할 수 있습니다.



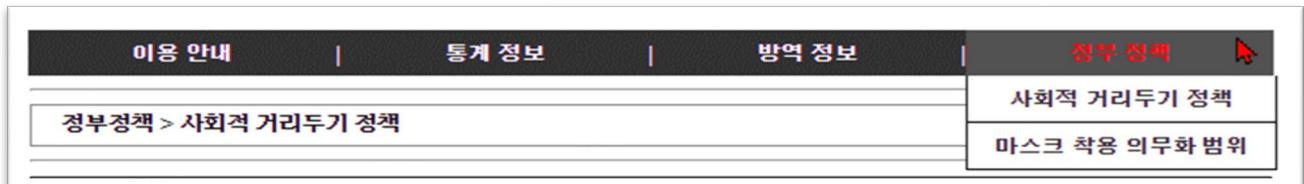
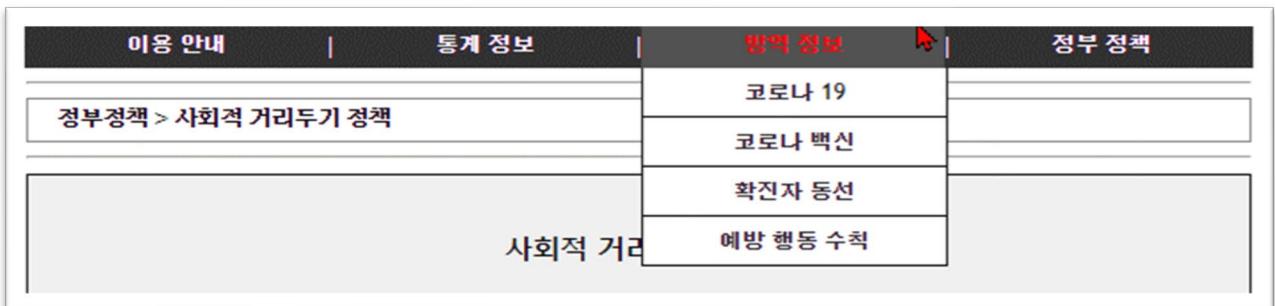
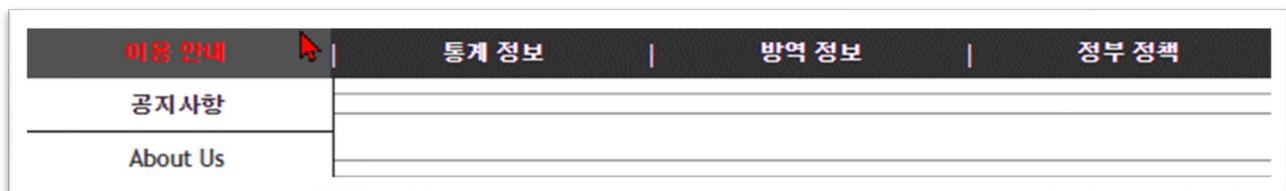
< 기타 기능 안내 >

08. 드롭다운 메뉴

CoronaCore 사이트는 사이트 상단 메뉴에서 마우스 호버 시 세로로 펼쳐지는 드롭다운 메뉴 구성을 통해 각각의 빈 페이지들을 연결하고 세부 메뉴와 컨텐츠를 작성하는 방식으로 작성되었습니다.

<https://unikys.tistory.com/369>

위 사이트는 Corona Core 사이트 제작에 사용된 상단 드롭다운 메뉴 제작 튜토리얼 사이트입니다. 메뉴 중 세로형 드롭다운 메뉴 튜토리얼을 참조해 HTML CSS 구조를 확인할 수 있습니다. 드롭다운 메뉴는 기본적으로 javascript 없는 transform 태그로만 구현해 사용해 사이트 접근 속도를 높일 수 있도록 작성하였습니다.



09. Detail/Summary

기본적인 코로나 정보에 대한 접기 펼치기 기능이 Detail/Summary 방식으로 구현되어 있습니다.



다만 텍스트 박스 하나를 열면 다른 하나는 닫히는 기능을 구현하기 위한 일부 자바스크립트가 코드가 사용되고있으며, 해당 부분은 다음과 같습니다.

```
<!--다른 summary를 열었을 때 기존에 열었던 summary 닫기-->
<script type="text/javascript">
// Fetch all the details element.
const details = document.querySelectorAll("details");
// Add the onclick listeners.
details.forEach((targetDetail) => {
    targetDetail.addEventListener("click", () => {
        // Close all the details that are not targetDetail.
        details.forEach((detail) => {
            if (detail !== targetDetail) {
                detail.removeAttribute("open");
            }
        });
    });
});
</script>
```

10. image scale 조정

사용자가 이미지 파일을 상세히 살펴볼 수 있는 이미지 스케일 기능도 내장되어 있습니다.

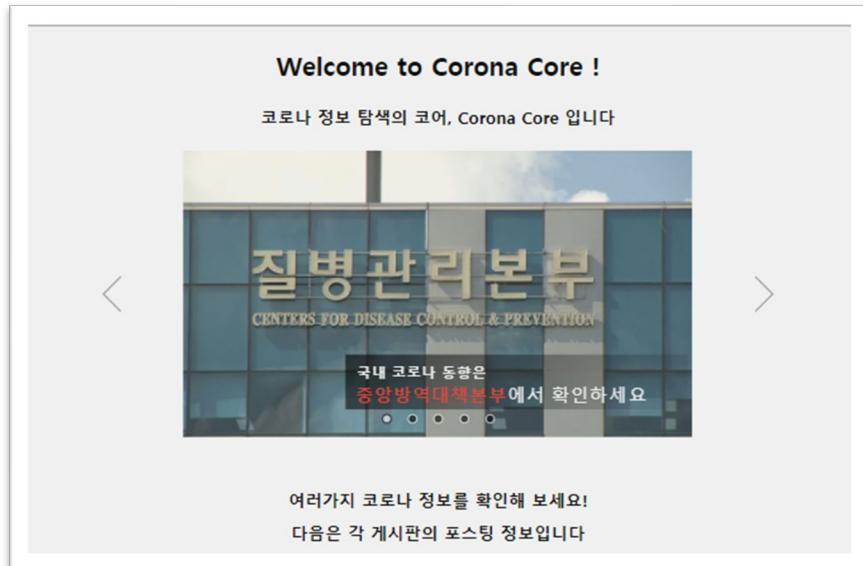
사용자가 이미지를 한번 누르면 이미지가 확대되고 다시 누르면 축소됩니다.

다음은 위 내용에 해당하는 자바스크립트 코드입니다.

```
<!-- 이미지 확대 -->
<script>
function img_click(event) {
    // 첫번째 클릭 시 확대
    if(clk==0){
        clk=1;
        event.target.style.transform = "scale(1.5)";
        event.target.style.zIndex = 1;
        event.target.style.transition = "all 0.3s";
    }
    // 두번째 클릭 시 축소
    else if (clk==1){
        clk=0;
        event.target.style.transform = "scale(1)";
        event.target.style.zIndex = 0;
        event.target.style.transition = "all 0.3s";
    }
    else{
        alert("err!")
    }
}
</script>
```

11. 이미지 슬라이더

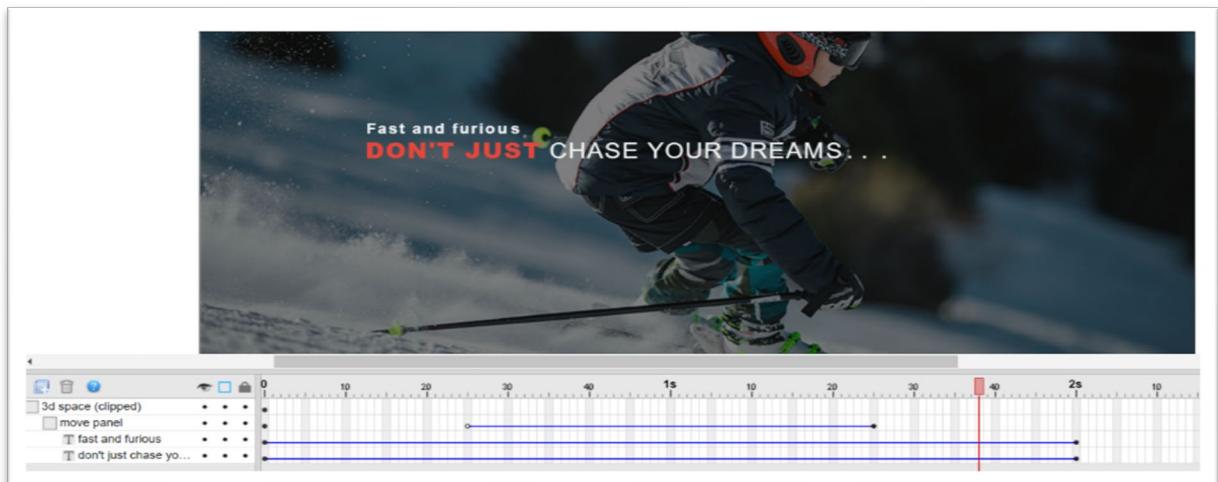
아래는 저희가 사용한 이미지 슬라이더입니다.



이미지 슬라이더는 JSSOR (<https://www.jssor.com/demos/image-slider.slider>) 이미지 슬라이더 편집기를 사용해 제작했습니다. 해당 사이트에서 적절한 샘플을 찾은 후 edit에 들어가서 적절히 슬라이더를 편집한 것입니다. 사이트는 사용자가 GUI 방식으로 어렵지 않게 편집할 수 있게 구성되어 있습니다.

사용 방법은 일반적인 영상 편집 프로그램을 다루는 방식과 비슷합니다. 텍스트 박스나 무브 패널을 작성한 후, 이를 타임라인에 알맞게 배치하고 이미지 링크를 걸어주는 방식입니다.

Gui를 활용해 외부 프로그램을 사용했기 때문에 상세한 슬라이더 작동 방식 전체를 설명하기는 어렵지만, 개발자는 standard.html 파일에서 이미지와 텍스트에 해당하는 부분을 찾아서 어렵지 않게 고칠 수 있을 것입니다.



12. 댓글 기능 (DISCUS API)

저희 CoronaCore 에선 DISQUS 사에 댓글 서비스를 신청해 저희 사이트에서 댓글 기능을 작동하도록 해주는 링크 코드를 받아 활성화하였습니다. 현재 댓글 기능과 댓글 데이터 스토리지는 DISQUS 사에서 관리해주고 있으며, 관리자 계정으로 설정한 이메일 계정에는 댓글이 올 때마다 알림이 오고, 관리자는 DISQUS 사에 접속해서 CoronaCore에 활성화한 댓글들에 대해서 관련된 통계를 확인할 수 있는 상태입니다.

```
<!-- 댓글 기능 -->
<div class="chatting-box">
  <div class="chatting-box-outside">
    <div class="chatting-box-inside">
      <div id="disqus_thread"></div>
      <script>
        /**
         * RECOMMENDED CONFIGURATION FOR ALL PLUGINS
         */
        var disqus_config = function () {
          this.page.url = PAGE_URL;
          this.page.identifier = PAGE_IDENTIFIER;
        };
        (function() { // DON'T EDIT BELOW THIS LINE
          var d = document, s = d.createElement('script');
          s.src = 'https://coronacore-site.disqus.com/embed.js';
          s.setAttribute('data-timestamp', +new Date());
          (d.head || d.body.appendChild(s));
        })();
      </script>
      <noscript>Please enable JavaScript to view the <a href="https://disqus.com/?ref_noscript">comments powered by Disqus.</a></noscript>
    </div>
  </div>
<!-- Footer -->
<div id="footer"></div>
```

2 Comments coronacore.site ⓘ Disqus' Privacy Policy

CoronaCore Sort by Best

Join the discussion...

JES 13 hours ago 사용자 계정으로 작성한 댓글입니다.

CoronaCore 14 hours ago 관리자 계정으로 작성한 댓글입니다.

Subscribe Add Disqus to your site Do Not Sell My Data DISQUS



4. 설계와 디자인

4.1. 모듈 간 인터페이스

저희는 웹 사이트를 개발하므로 기본적인 정적 웹 사이트 제작 기술과 인터페이스가 사용됩니다. 웹 사이트 제작에 사용되는 기술들의 기본적인 연결은 다음과 같습니다.

사이트 플랫폼

사이트 플랫폼은 기본적으로 Github Pages 위에서 작동하는 웹 플랫폼에 정적 웹 사이트에서 관리자가 사이트를 관리할 수 있는 Github Jekyll 웹 생성기가 더해져, 관리자는 Mark Down 언어로 작성된 md 파일을 포스팅 형식으로 사용자에게 전달할 수 있습니다.

Github Pages	기본 레포지토리와 정적 페이지용 플랫폼 제공 (github.io)
Github Jekyll	정적 웹사이트 생성 툴
	MarkDown문서를 통한 포스팅 기능 제공

사이트 개발 언어

그리고 사이트 개발 언어는 기본적으로 아래의 표와 같이 기본적인 웹 개발 언어인 HTML, CSS, JavaScript 와 데이터 크롤링을 구현하기 위한 PYTHON 언어로 작성되었습니다.

구분	내용
HTML	기본적인 사이트 설계와 페이지들의 연결
CSS	웹 디자인
JavaScript	웹의 동적인 기능 구현
	주요 기능인 통계 정보를 사용자에게 보여주기 위한 Danfo.js (통계, 시각화)
python	beautifulsoup, requests 라이브러리를 이용하여 원하는 데이터를 수집하고 이를 html파일로 저장하기 위해 사용됩니다.

4.2. Database

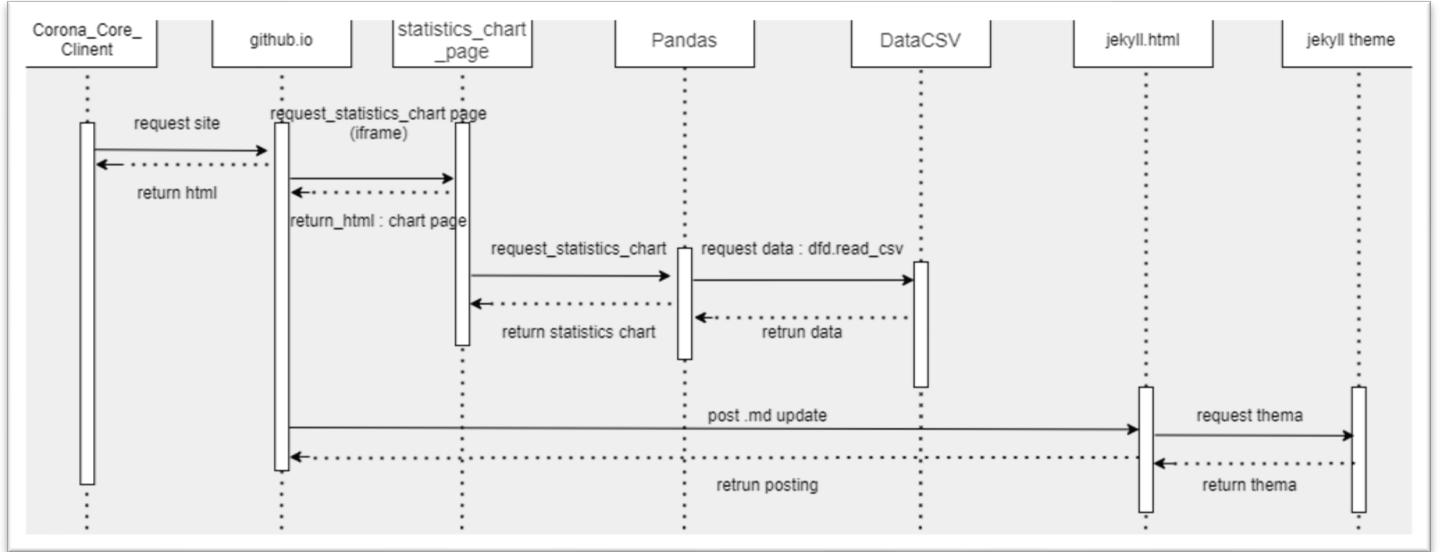
저희 CoronaCore은 공공데이터 포털에서 받을 수 있는 오픈 API 데이터를 활용해 통계 그래프를 제작하였으며, 사용된 통계 데이터의 구성은 다음과 같습니다.

	Corona core사이트의 통계 기능에 사용한 data set
	공공데이터 포털에 포함되어 있는 다른 data

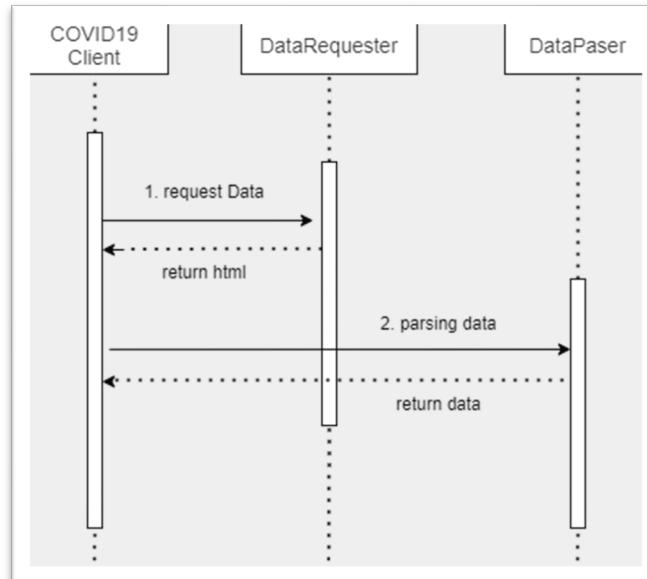
항목명(국문)	항목명(영문)	항목크기	항목구분	샘플데이터	항목설명
결과코드	resultCode	2	필수	0	결과코드
결과메시지	resultMsg	50	필수	OK	결과메시지
한 페이지 결과 수	numOfRows	4	필수	10	한 페이지 결과 수
페이지 번호	pageNo	4	필수	1	페이지번호
전체 결과 수	totalCount	4	필수	3	전체 결과 수
게시글번호(국내 시도별 발생현황 고유값)	SEQ	30	필수	130	게시글번호(국내 시도별 발생현황 고유값)
등록일시분초	CREATE_DT	30		15:59:0	등록일시분초
사망자 수	DEATH_CNT	15		204	사망자 수
시도명(한글)	GUBUN	30		부산	시도명(한글)
시도명(중국어)	GUBUN_CN	30		null	시도명(중국어)
시도명(영어)	gubunEn	30		null	시도명(영어)
전일대비 증감 수	INC_DEC	15		39	전일대비 증감 수
격리 해제 수	ISOL_CLEAR_CNT	15		6973	격리 해제 수
10 만명당 발생률	QUR_RATE	30		20.1	10 만명당 발생률
기준일시	STD_DAY	30		2020년 3월 13일 00시	기준일시
수정일시분초	UPDATE_DT	30		null	수정일시분초
확진자 수	DEF_CNT	15	옵션	13561	확진자 수
격리중 환자수	ISOL_ING_CNT	15	옵션	9	격리중 환자수
해외유입 수	OVER_FLOW_CNT	15	옵션	14	해외유입 수
지역발생 수	LOCAL_OCC_CNT	15	옵션	7	지역발생 수

4.3 사이트 동작 관련 sequence diagram

아래는 코로나 코어 사이트의 서버상의 동작을 표현하는 sequence diagram입니다. 전체 웹 서비스 구동에 관한내용이 아닌 저희가 사용하는 정적 페이지에 대한 부분만 구성했으며, 사용되는 모듈간의 동작을 클래스화하여 나타내었습니다.



또한 웹 크롤러 동작에 대한 다이어그램도 다음과 같이 간단히 구성하였습니다.



4.4 risk assessment(위험 평가)

Identify Risks	Plan Risk Responses
사이트 구축을 위한 github web hosting 서비스 문제 발생 가능성 (사이트 구축 비용 추가 발생 위험)	대체 web hostin platform : Netlify, GCP(google cloud platform), Microsoft Azure, AWS(Amazon Web service) 등 의 정적 웹 호스팅 환경 사용
데이터, 컨텐츠 정보에 대한 저작권 위반	자유로운 활용이 보장되는 공공데이터 사용, 공공누리 유형조건(출처표시, 상업적 이용 금지, 변경 금지)에 따라 저작물 사용

4.5 coding style guidelines

HTML _ CSS

- 기본적인 HTML 드롭다운 메뉴바 레이아웃으로 전체 틀을 작성하였습니다.
- 우선 전체 코드에 적용되는 CSS 파일을 기본적으로 만들어 전체 코드에 적용하였고, 각 메뉴별로 필요한 CSS 는 해당 메뉴 폴더별로 작성하였으며, 각각의 HTML 파일에 독자적으로 사용되는 CSS 파일은 해당 HTML 코드 상단에 CSS를 작성하는 식으로 코드를 작성하였습니다.
- CSS 의 기본적인 Class 와 ID 개념에 부합하게 디자인을 분류했습니다.

JAVASCRIPT

- 파일을 모듈화하여 같은 함수를 자주 쓰는 것을 줄여 코드 간소화와 가독성을 높이고 각 파일마다 함수로 분리하여 함수의 이름에 의미를 부여하여 어떤 의도의 함수인지 알 수 있게 하였습니다. 오히려 가독성이 떨어지지 않을 정도의 간단한 주석처리를 진행하였습니다.

PYTHON

- 파이썬이 활용된 부분은 데이터 크롤링 파트에서 beautiful soup 라이브러리가 활용된 부분입니다. Import 부분에서 request 와 beautiful soup 모듈을 임포트하면 데이터 크롤링을 위한 python 파일 실행에 문제는 없을 것 입니다.
-

4.6 team structure

(팀 구조와 작업 분담)

팀원	역할 분담
김형욱	<ul style="list-style-type: none">- 깃허브 PAGES 플랫폼과 Corona Core 아이디어 원안 제공- 공공데이터API data 추출해 Danfo.js로 5메뉴, 7종의 통계 그래프 작성- Heroku를 사용한 관리자 공공API data 최신화 구현<ul style="list-style-type: none">1. 코로나 전체 기간 일일 확진자 GRAPH(*2)2. 날짜 & 지역별 확진자 증가 추세 GRAPH (*2)3. 지역별 확진자 비교 GRAPH4. 특정 수치 초과일 조회5. 지역별 확진자 지도
윤시준	<ul style="list-style-type: none">- 드롭다운형 상단 메뉴와 페이지 레이아웃 구성- 메인 홈페이지의 이미지 슬라이더 작성- 사이트 Footer와 SNS 공유버튼 작성- 구글 검색어와 사이트 도메인 등록- 그 외 TEXT 메뉴와 컨텐츠 구성
최원석	<ul style="list-style-type: none">- Python beautiful soup를 통한 데이터 크롤링<ul style="list-style-type: none">└ 실시간 상황판 part└ 사회적 거리두기 단계 지역별 단계 part- Github Jekyll을 활용해 사이트 관리자의 마크다운 페이지 작성- 그 외 TEXT 메뉴와 컨텐츠 구성- 사이트 상단 이동 꺽쇠 Anchor 작성

5. 테스팅

	메뉴	Test	ch k
통계 정보	공통	통계 기능들이 페이지에서 보여지는가	V
		원하는 날짜의 확진자 데이터가 정확히 출력되는가	V
		확진자 데이터가 데이터 추가 후에도 차트 출력이 되는가	V
		페이지에서 데이터 다운로드가 진행이 되는가	V
	국내 누적 확진자 통계	누적 환진자 데이터가 line chart로 출력이 되는가	V
	지역별 확진자 증가세	지역별 확진자 데이터가 Pie, Bar로 각각 출력이 되는가	V
	특정 수치 초과일 조회	출력되는 표의 데이터가 날짜순서로 출력이 되는가	V
	지역별 확진자 수 조회	지역별로 지도상에 확진자 숫자를 보여주는가	V
일반 정보 (이용안내, 방역정보, 정부정책, 메인 페이지)	공통	지역별 집계를 조회할 수 있도록 지역 추가/삭제 기능을 구현하고 버튼 입력시마다 갱신되도록 했는가	V
		사용자에게 이미지, 텍스트 정보들이 모두 출력되는가	V
		컨텐츠에 맞는 내용이 작성되어 있는가	V
		해상도가 달라져도 페이지의 내용을 확인할 수 있는가	V
		zoom, slide, details 등 기능 사용 시 가려지는 부분이 없는가	V
		link가 정확한 경로를 가르키고 있는가	V
	메인	관리자 포스팅이 모두 출력되는가	V
		포스팅 내용 추가 후 반영되는가	V

6. Github link & Reference

6.0 코로나 코어

http://coronacore.site/main/CoronaCore/01_code/index.html

6.1 Github link

깃허브 <https://github.com/choiwonseok9/choiwonseok9.github.io>

6.2 Data Source

공공데이터API

<http://openapi.data.go.kr/openapi/service/rest/Covid19/getCovid19SidoInfStateJson?serviceKey=u%2BXsTVV1nl13bsl5mxFNCaZ0o48loSbVj4pQoNm2xFONwLswAgYcNrabZ9jBp7mIdKQZSgYV7NBAjOyHH6cr%2Fg%3D%3D&startCreateDt=20201231&endCreateDt=20201231>

6.3 Corona Reference

코로나바이러스감염증19 (보건복지부)

<http://ncov.mohw.go.kr/>

Corona Board

<https://coronaboard.kr/>

Corona Live

<https://corona-live.com/>

6.4 Domain & API

도메인

<https://www.hostinger.com/>

이미지 슬라이더

<https://www.jssor.com/>

댓글 기능 API

<https://disqus.com/home/>

7. 분석 및 소감

이하는 저희 팀에서 본 Corona Core 프로젝트에 대한 아쉬운 부분과 잘 해냈다고 자평한 부분입니다.

FrameWork를 사용하지 못한 아쉬움

저희 CoronaCore은 개발전에 react나 vue와 같은 것들을 지식을 갖추고 적용했다면 어땠을까 하는 진행에 대한 아쉬움이 있습니다. 이를 통해 중복되는 작업을 줄이고, 효율적인 업무를 가져갈 수 있었을 것으로 생각합니다.

반응형 웹사이트 미 구현에 대한 아쉬움

마찬가지로 모바일의 시대에 프론트에 대한 중요한 요소 중 하나인 반응형 웹사이트 구현을 못한 것이 아쉽습니다.

코로나 논쟁 메뉴 미구현

준비한 컨텐츠를 다 사용하지 못했습니다. 자료 수집은 마쳤으나 프로젝트 방향성을 위해 생략한 내용입니다. 자세한 내용은 깃허브 링크에 controversies 에도 간략히 정리되어 있습니다.

비동기 처리에 대한 이해도 부족

웹에 대한 서버에서 데이터를 가져와 처리하는 것에 대한 비동기 처리에 대한 이해도의 부족으로 인한 시간 소비와 class형태가 아닌 기능구현이 목적이 커서 함수형 코드로 작성한 부분에 대한 아쉬움이 있습니다.

교수님 피드백 처리

저희는 사이트에 부족한 UI로 지적받은 부분을 매번 최대한 개선하려 했습니다. 아래 화면은 마지막 수업에서 지적해 주신 통계 그래프의 시점-종점을 분리한 부분입니다. 수업마다 남겨주신 피드백이 사이트 개선에 큰 도움이 되었던 것 같습니다.

날짜 & 지역별 코로나 감염자수 확인

● 시작 날짜 입력
연도 - 월 - 일

● 종료 날짜 입력
연도 - 월 - 일

날짜 설정 버튼

슬라이더 조정 시 원형그래프가 출력됩니다