

# 캡스톤 디자인 | 최종결과 보고서

프로젝트 제목(국문): 노션 연동 블로그 서비스

프로젝트 제목(영문): Notion-linked Blog Service

|                         |         |
|-------------------------|---------|
| 프로젝트 팀(원): 학번: 20197121 | 이름: 정지용 |
| 프로젝트 팀(원): 학번: 20181605 | 이름: 홍현표 |
| 프로젝트 팀(원): 학번: 20177123 | 이름: 김태언 |

1. 중간보고서의 검토결과 심사위원의 '수정 및 개선 의견'과 그러한 검토의견을 반영하여 개선한 부분을 명시하시오.

해당 사항 없음

2. 기능, 성능 및 품질 요구사항을 충족하기 위해 본 개발 프로젝트에서 적용한 주요 알고리즘, 설계방법 등을 기술하시오.

## 2.1 회원가입 기능

이메일 인증을 구현하기 위해 Google에서 제공하는 계정 기반 SMTP 서버 활용 및 Redis를 활용한 인증 코드 관리

### 앱 비밀번호로 로그인

도움말: 앱 비밀번호는 권장되지 않으며 대부분의 경우 필요하지 않습니다. 계정을 안전하게 보호하려면 'Google 계정으로 로그인'을 사용하여 앱을 Google 계정에 연결하세요.

앱 비밀번호는 16자리 비밀번호로, 보안 수준이 낮은 앱 또는 기기가 Google 계정에 액세스할 수 있도록 권한을 부여합니다. 앱 비밀번호는 [2단계 인증](#)이 사용 설정된 계정에서만 이용할 수 있습니다.

### 앱 비밀번호를 사용해야 하는 경우

도움말: iOS 11 이상을 실행하는 iPhone 및 iPad에는 앱 비밀번호가 필요하지 않습니다. 대신 'Google 계정으로 로그인'을 사용하세요.

앱에서 'Google 계정으로 로그인'을 제공하지 않는다면 다음 방법 중 하나를 이용하면 됩니다.

- 앱 비밀번호 사용
- 보안 수준이 높은 앱 또는 기기로 전환

### 앱 비밀번호 만들고 사용하기

중요: 앱 비밀번호를 만들려면 Google 계정에 2단계 인증이 필요합니다.

2단계 인증을 사용 중이며 로그인할 때 '잘못된 비밀번호' 오류가 표시된다면 앱 비밀번호를 사용해 볼 수 있습니다.

1. [Google 계정](#) 으로 이동합니다.
2. [보안을](#) 선택합니다.
3. 'Google에 로그인'에서 [2단계 인증](#)을 선택합니다.
4. 페이지 하단에서 [앱 비밀번호](#)를 선택합니다.
5. 앱 비밀번호를 사용할 위치를 기억하는 데 도움이 되는 이름을 입력합니다.
6. [생성](#)을 선택합니다.
7. 앱 비밀번호를 입력하려면 화면에 표시된 안내를 따릅니다. 앱 비밀번호는 기기에서 생성되는 16자 코드입니다.
8. [완료](#)를 선택합니다.

2단계 인증을 설정했는데 앱 비밀번호를 추가하는 옵션을 찾을 수 없다면 다음과 같은 이유 때문일 수 있습니다.

- Google 계정에 [보안 키](#)에만 2단계 인증이 설정되어 있습니다.
- 직장, 학교 또는 다른 조직 계정에 로그인한 상태입니다.
- Google 계정에 [고급 보호](#)가 설정되어 있습니다.

## 2.2 무한 스크롤 기능

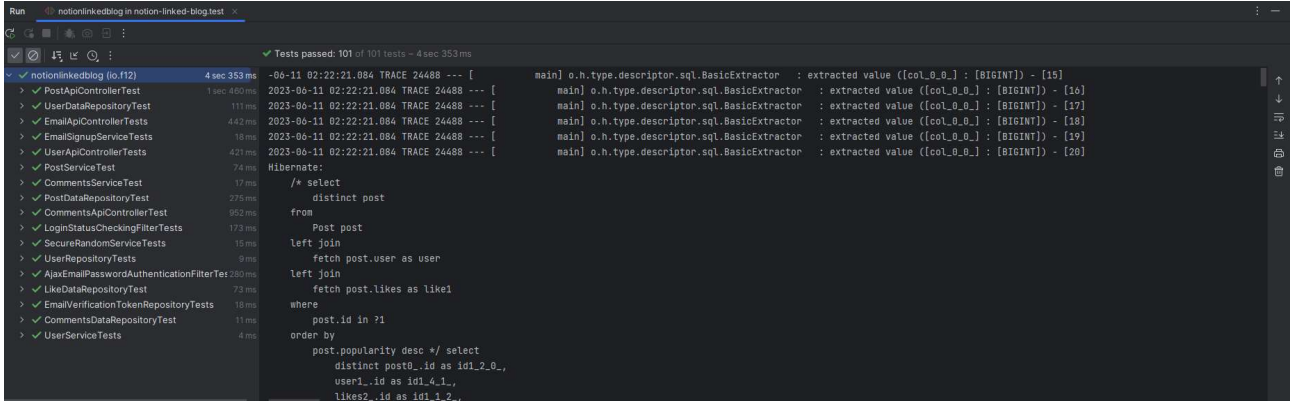
보다 나은 사용자 경험을 위해 Web API 중 IntersectionObserverAPI를 활용한 React Hook 개발 및 InfiniteScroll 기능 개발 후 적용

## 2.3 이미지 업로드 속도 개선

readAsDataURL과 다르게 동기적으로 실행되는 createObjectURL과 revokeObjectURL을 사용하여 업로드 속도 개선

## 2.4 유닛 테스트

테스트를 통해 문제 발생 시 정확하게 어느 부분이 잘못되었는지를 빨리 확인할 수 있게 해 주며 프로그램의 안정성이 높아진다. 또한 내가 수정한 코드가 어느 범위까지 영향을 미치고 있는지 간단하게 확인 가능



## 2.5 애자일 방법론

스프린트를 만들어 정해진 기한내에 목표치를 정하고 정기적인 스크럼을 통해 서로의 진행상황을 공유하고 문제 발생시 문제 상황을 공유



그림 - Daily Scrum을 통해 현재 진행상황과 문제 사항을 공유

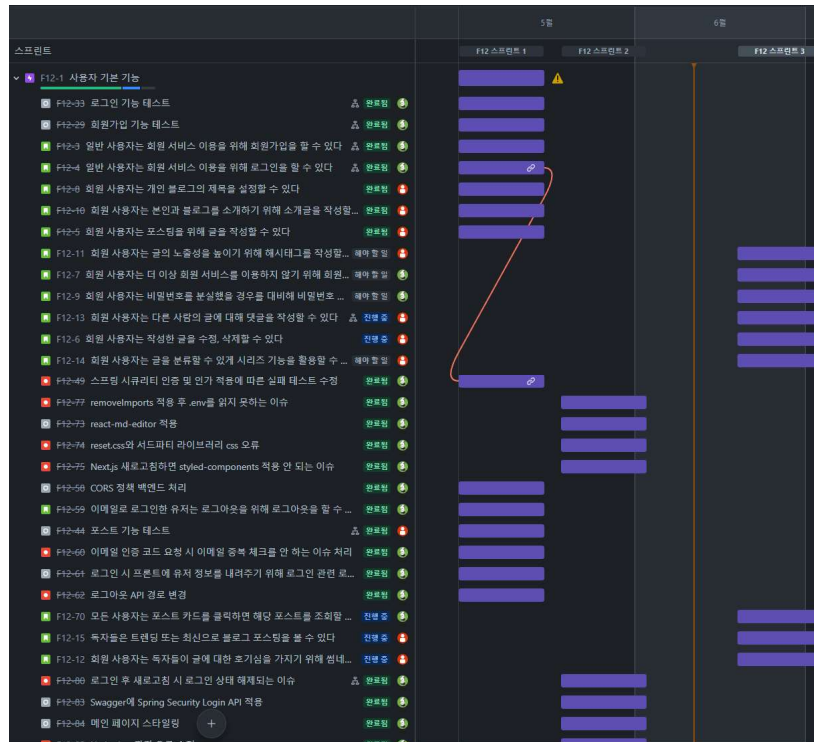


그림 - 스프린트를 통해 진행 사항 공유

## 2.6 인기순 조회 기능

비 개인화 추천 알고리즘 적용(Hacker News Algorithm + Reddit Formula Algorithm 변형)

a. Hacker News Algorithm – 조회수를 통한 인기도 측정

$$score = \frac{pageviews - 1}{(age + 2)gravity}$$

pageviews: 조회수, age: 현재시각 - 기사업로드 날짜,

gravity: 중력상수(오래된 기사일수록 score값이 작아지도록 조정)

따라서 조회수가 높을수록 점수가 높아지지만, 시간이 지남에 따라 오래된 뉴스는 조회수가 높더라도 낮은 점수를 가지게 되고 중력상수를 통해 시간의 흐름에 따른 점수 하락폭을 정할 수 있다.

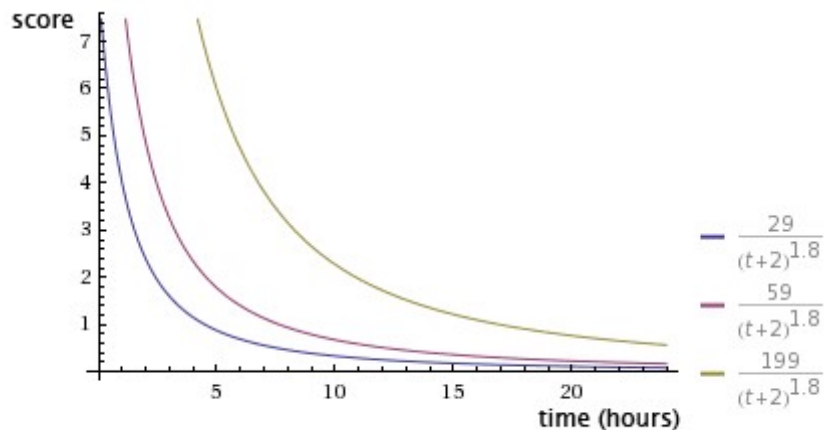


그림 - 중력상수가 같을 때 시간에 따른 점수 변화

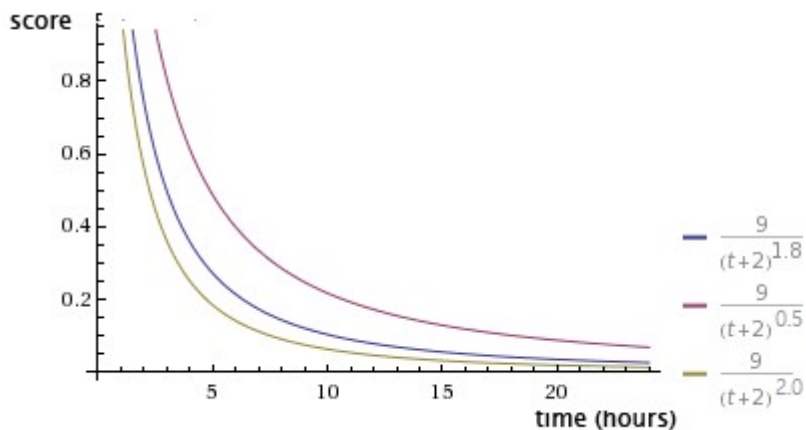


그림 - 조회수가 같을 때 시간에 따른 점수의 변화

위 두 그림을 통해 중력상수를 통해 시간이 지남에 따라 일정하게 점수를 줄일 수 있음과 함께 중력상수를 통해 하락 폭을 조절 할 수 있음을 확인하였다.

b. Reddit Formula

미국의 커뮤니티 사이트 Reddit의 점수화 알고리즘으로 인기도(좋아요) 와 뉴스의 나이를 반영하여 점수를 측정한다.

$$score = \log_{10}(\max(|ups - downs|, 1)) + \frac{sign(ups - downs) \cdot seconds}{45000}$$

ups: 추천 수, downs: 비추천 수

sign(x): if  $x > 0$ , return 1, elif  $x < 0$ , return -1 else, return 0

seconds: 게시물이 업로드된 시간

먼저, 로그함수를 사용하여 충분히 인기도가 높은 포스트에 대해서는 가산점을 적게 준다.

이후에 최근의 뉴스에게 더 높은 점수를 부여하여 점수를 측정한다.

## 2.7 GitHub를 활용한 코드 리뷰 및 작업 추적 - GitHub의 Pull Request를 활용하여 서로의 변경 사항을 교차 검증하여 문제 사항을 미리 확인하고 리뷰를 통해 코드 품질 개선

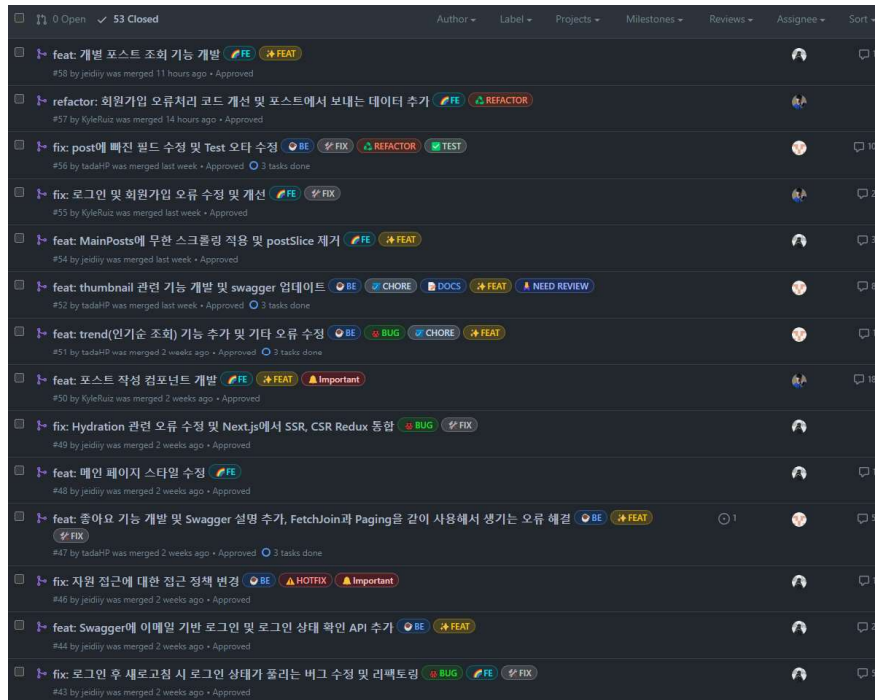


그림 - PR 목록

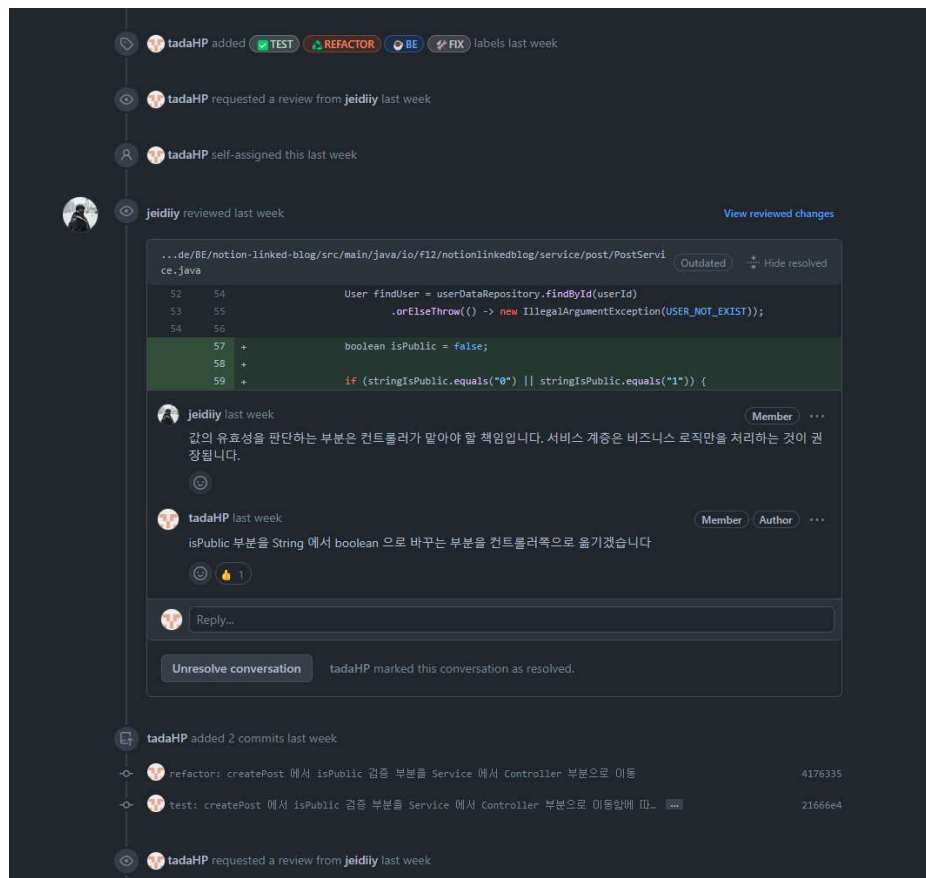


그림 - PR에서의 토론

# Bug: 회원 조회시 로그인 요청 오류 #45

Closed
tadaHP opened this issue 2 weeks ago · 3 comments

tadaHP commented 2 weeks ago

Member · ...

## Commit Status

- 알수없음

## When the Bug occurs

- 로그인을 하지 않은 상태에서 회원을 조회한 경우 로그인을 해야한다는 오류 발생 -> 로그인을 하지 않아도 유저 조회는 가능해야함

## Screenshots

Body Cookies (1) Headers (13) Test Results

401 Unauthorized
54 ms
534 B

Pretty Raw Preview Visualize Text

```
1 {
  "loginUrl": "/api/login/email",
  "message": "로그인 후 이용해 주세요"
}
```

tadaHP added BE BUG labels 2 weeks ago

jeidiy commented 2 weeks ago

Member · ...

로그인하지 않고 특정 유저를 조회하는 상황이 어떤 상황이 있을까요?

jeidiy commented 2 weeks ago

Member · ...

로그인하지 않고 특정 유저를 조회하는 상황이 어떤 상황이 있을까요?  
생각해보니 특정 유저의 블로그나 프로필을 조회할 때 필요할 것 같군요

tadaHP commented 2 weeks ago

Member Author · ...

넵 그런데 로그인되어있지 않으면 조회가 불가능하더라고요. 어느부분에서 로그인체크가 들어가는건지 몰라 issue로 일단 띄워놨습니다

jeidiy mentioned this issue 2 weeks ago

fix: 자원 접근에 대한 접근 정책 변경 #46

Merged

jeidiy closed this as completed 2 weeks ago

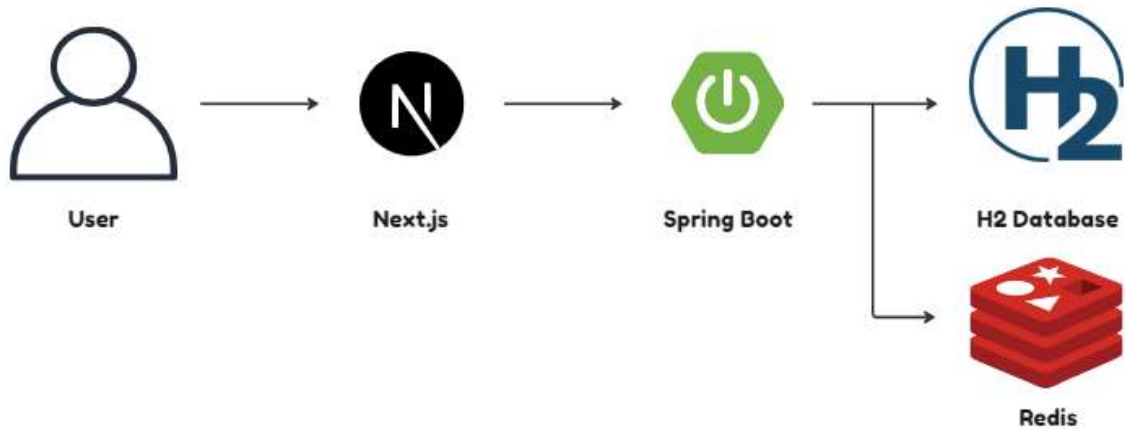
그림 - Issue를 통한 오류 처리 과정

- 7 -



3. 요구사항 정의서에 명시된 기능 및 품질 요구사항에 대하여 최종 완료된 결과를 기술하시오. (작성요령: 최종 완료된 시스템에 대해 아래의 산출물을 작성합니다)

- 소프트웨어(또는 하드웨어, 시스템) 아키텍처 또는 전체 시스템 구성도



- API 명세서

a. 사용자 API

| 요구사항 명칭     | 회원 정보 조회             | HTTP Method  | GET |
|-------------|----------------------|--|-----|
| 요구사항 상세설명   | id에 해당하는 사용자의 정보를 조회 |  |     |
| URL         | /api/users/{id}      |  |     |
| 응답 코드/ 성공여부 | 설명                   | 응답 바디  |     |
| 200/성공      | 회원 정보 조회 성공          | application/json<br><pre>{   "id": "Integer",   "username": "string",   "email": "string",   "profile": "string",   "introduction": "string",   "blogTitle": "string",   "githubLink": "string",   "instagramLink": "string" }</pre> |     |
| 401/실패      | 미허가                  | application/json<br><pre>{   "errorMessage": "string",   "errorCode": "Integer" }</pre>  |     |
| 404/실패      | 존재하지 않는 리소스 접근       |  |     |
| 415/실패      | 지원하지 않는 미디어 타입       |  |     |

| 요구사항 명칭     | 회원 정보 변경   | HTTP Method  | PUT |
|-------------|--|--|-----|
| 요구사항 상세설명   | id에 해당하는 사용자의 정보를 변경   |  |     |
| URL         | /api/users/{id}  |  |     |
| 요청 바디       | <pre> application/json  {   "username": "string",   "profile": "string",   "introduction": "string",   "blogTitle": "string",   "githubLink": "string",   "instagramLink": "string" } </pre> |  |     |
| 응답 코드/ 성공여부 | 설명   | 응답 바디  |     |
| 201/성공      | 회원 정보변경 성공(추후 변경)  | <pre> application/json  {   "id": "Integer",   "username": "string",   "email": "string",   "profile": "string",   "introduction": "string",   "blogTitle": "string",   "githubLink": "string",   "instagramLink": "string" } </pre> |     |
| 401/실패      | 미허가  | <pre> application/json  {   "loginUrl": "string",   "message": "string" } </pre>   |     |
| 404/실패      | 존재하지 않는 리소스 접근   | <pre> application/json  {   "errorMessage": "string",   "errorCode": "Integer" } </pre>  |     |
| 415/실패      | 지원하지 않는 미디어 타입   |  |     |

| 요구사항 명칭     | 회원 정보 삭제             | HTTP Method  | DELETE |
|-------------|----------------------|--|--------|
| 요구사항 상세설명   | id에 해당하는 사용자의 정보를 삭제 |  |        |
| URL         | /api/users/{id}      |  |        |
| 응답 코드/ 성공여부 | 설명                   | 응답 바디  |        |
| 204/성공      | 회원 정보삭제 성공           | none   |        |
| 401/실패      | 미허가                  | <pre>application/json {   "loginUrl": "string",   "message": "string" }</pre>        |        |
| 404/실패      | 존재하지 않는 리소스 접근       | <pre>application/json {   "errorMessage": "string",   "errorCode": "Integer" }</pre> |        |
| 415/실패      | 지원하지 않는 미디어 타입       |  |        |

| 요구사항 명칭     | email을 이용한 회원가입  | HTTP Method  | POST |
|-------------|--|--|------|
| 요구사항 상세설명   | email 인증 후 회원가입시 사용하는 API입니다   |  |      |
| URL         | /api/users/email/signup  |  |      |
| 요청 바디       | <pre>application/json {   "username": "string",   "email": "string",   "password": "string",   "profile": "string",   "introduction": "string",   "blogTitle": "string",   "githubLink": "string",   "instagramLink": "string" }</pre> |  |      |
| 응답 코드/ 성공여부 | 설명   | 응답 바디  |      |
| 200/성공      | 회원 정보 생성 성공  | <pre>application/json {   "id": "Integer", }</pre>                                   |      |
| 401/실패      | 미허가  | <pre>application/json {   "errorMessage": "string",   "errorCode": "Integer" }</pre> |      |
| 404/실패      | 존재하지 않는 리소스 접근   |  |      |
| 415/실패      | 지원하지 않는 미디어 타입   |  |      |

-b. 포스트 API

| 요구사항 명칭     | 포스트 id로 포스트 조회   | HTTP Method  | GET |
|-------------|------------------|--|-----|
| 요구사항 상세설명   | id에 해당하는 포스트를 조회 |  |     |
| URL         | /api/posts/{id}  |  |     |
| 응답 코드/ 성공여부 | 설명               | 응답 바디  |     |
| 200/성공      | 포스트 조회 성공        | <pre> application/json {   "postId": "Integer",   "title": "string",   "content": "string",   "viewCount": "Integer",   "likes": "Integer",   "requestThumbnailLink": "string",   "description": "string",   "createdAt": "dateTime",   "countOfComments": "Integer",   "author": "string",   "avatar": "string" }</pre> |     |
| 401/실패      | 미허가              | <pre> application/json {   "errorMessage": "string",   "errorCode": "Integer" }</pre>  |     |
| 404/실패      | 존재하지 않는 리소스 접근   |  |     |
| 415/실패      | id 미존재           |  |     |

| 요구사항 명칭     | 포스트 내용 수정   | HTTP Method   | PUT |
|-------------|---|---|-----|
| 요구사항 상세설명   | id에 해당하는 포스트의 정보를 수정  |   |     |
| URL         | /api/posts/{id}   |   |     |
| 요청 바디       | <pre> application/json {   "title": "string",   "content": "string" }</pre> |   |     |
| 응답 코드/ 성공여부 | 설명  | 응답 바디   |     |
| 302/성공      | 회원 정보변경 성공(추후 변경)   |   |     |
| 401/실패      | 작성자과 수정시도자 불일치  | <pre> application/json {   "errorMessage": "string",   "errorCode": "Integer" }</pre> |     |
| 404/실패      | 존재하지 않는 리소스 접근  |   |     |
| 415/실패      | 지원하지 않는 미디어 타입  |   |     |

|             |                  |  |        |
|-------------|------------------|--|--------|
| 요구사항 명칭     | 포스트 삭제           | HTTP Method  | DELETE |
| 요구사항 상세설명   | id에 해당하는 포스트를 삭제 |  |        |
| URL         | /api/posts/{id}  |  |        |
| 응답 코드/ 성공여부 | 설명               | 응답 바디  |        |
| 204/성공      | 포스트 삭제 성공        | none   |        |
| 401/실패      | 미허가              | <pre>application/json {   "loginUrl": "string",   "message": "string" }</pre>        |        |
| 404/실패      | 존재하지 않는 리소스 접근   | <pre>application/json {   "errorMessage": "string",   "errorCode": "Integer" }</pre> |        |
| 415/실패      | 지원하지 않는 미디어 타입   |  |        |

|             |   |  |      |
|-------------|---|--|------|
| 요구사항 명칭     | 포스트 생성  | HTTP Method  | POST |
| 요구사항 상세설명   | 포스트 생성시 사용하는 API  |  |      |
| URL         | /api/posts  |  |      |
| 요청 바디       | <pre> application/json  {   "thumbnail": "string",   "title": "string",   "content": "string",   "description": "string",   "isPublic": "string" } </pre> |  |      |
| 응답 코드/ 성공여부 | 설명  | 응답 바디  |      |
| 201/성공      | 포스트 생성 성공   | <pre> application/json  {   "postId": "Integer",   "title": "string",   "content": "string",   "viewCount": "Integer",   "likes": "Integer",   "requestThumbnailLink": "string",   "description": "string",   "createdAt": "dateTime",   "countOfComments": "Integer",   "author": "string",   "avatar": "string" } </pre> |      |
| 401/실패      | 회원 미 로그인  | <pre> application/json  {   "loginUrl": "string",   "message": "string" } </pre>   |      |
| 404/실패      | 회원 데이터 미존재  | <pre> application/json  {   "errorMessage": "string",   "errorCode": "Integer" } </pre>  |      |
| 415/실패      | 지원하지 않는 미디어 타입  |  |      |

|             |                          |   |      |
|-------------|--------------------------|---|------|
| 요구사항 명칭     | 포스트 좋아요                  | HTTP Method   | POST |
| 요구사항 상세설명   | 포스트에 좋아요 클릭시 사용하는 API    |   |      |
| URL         | /api/posts/like/{postId} |   |      |
| 응답 코드/ 성공여부 | 설명                       | 응답 바디   |      |
| 201/성공      | 좋아요 상태 변경 성공             |   |      |
| 401/실패      | 회원 미 로그인                 | application/json<br><pre>{   "loginUrl": "string",   "message": "string" }</pre>        |      |
| 404/실패      | 회원 데이터 미존재               | application/json<br><pre>{   "errorMessage": "string",   "errorCode": "Integer" }</pre> |      |
| 415/실패      | 지원하지 않는 미디어 타입           |   |      |

|             |                                      |   |     |
|-------------|--------------------------------------|---|-----|
| 요구사항 명칭     | 인기순으로 포스트 조회                         | HTTP Method   | GET |
| 요구사항 상세설명   | 메인페이지에서 사용하는API, 인기순으로 작성된 포스트들을 가져옴 |   |     |
| URL         | /api/posts/trend/{pageNumber}        |   |     |
| 응답 코드/ 성공여부 | 설명                                   | 응답 바디   |     |
| 200/성공      | 포스트 조회 성공                            | <pre> Application/json {   "pageSize": "Integer",   "pageNow": "Integer",   "elementsSize": "Integer",   "posts": [     {       "postId": "Integer",       "title": "string",       "content": "string",       "viewCount": "Integer",       "likes": "Integer",       "requestThumbnailLink": "string",       "description": "string",       "createdAt": "dateTime",       "countOfComments": "Integer",       "author": "string",       "avatar": "string"     }   ] }</pre> |     |
| 400 / 실패    | 파라미터(페이지 번호) 미존재 혹은<br>파라미터 타입 오류    | <pre> Application/json {   "errorMessage": "string",   "errorCode": "Integer" }</pre>   |     |
| 401 / 실패    | Unauthorized                         |   |     |
| 404 / 실패    | Not Found                            |   |     |
| 415 / 실패    | Unsupported Media Type               |   |     |



|             |  |  |     |
|-------------|--|--|-----|
| 요구사항 명칭     | Title로 포스트 조회  | HTTP Method  | GET |
| 요구사항 상세설명   | title 이 포함되어있는 포스트들을 가져옴   |  |     |
| URL         | /api/posts/title   |  |     |
| 요청 바디       | <pre> Application/json {   "param": "string",   "pageNumber": "Integer" } </pre> |  |     |
| 응답 코드/ 성공여부 | 설명   | 응답 바디  |     |
| 200/성공      | 포스트 조회 성공  | <pre> Application/json {   "pageSize": "Integer",   "pageNow": "Integer",   "elementsSize": "Integer",   "posts": [     {       "postId": "Integer",       "title": "string",       "content": "string",       "viewCount": "Integer",       "likes": "Integer",       "requestThumbnailLink": "string",       "description": "string",       "createdAt": "dateTime",       "countOfComments": "Integer",       "author": "string",       "avatar": "string"     }   ] } </pre> |     |
| 400 / 실패    | 파라미터(페이지 번호) 미존재 혹은<br>파라미터 타입 오류  | <pre> Application/json {   "errorMessage": "string",   "errorCode": "Integer" } </pre>   |     |
| 401 / 실패    | Unauthorized   |  |     |
| 404 / 실패    | Not Found  |  |     |
| 415 / 실패    | Unsupported Media Type   |  |     |

|             |                                  |   |     |
|-------------|----------------------------------|---|-----|
| 요구사항 명칭     | imageName에 해당하는 이미지 조회           | HTTP Method   | GET |
| 요구사항 상세설명   | imageName에 해당하는 이미지를 가져옴         |   |     |
| URL         | /api/posts/thumbnail/{imageName} |   |     |
| 응답 코드/ 성공여부 | 설명                               | 응답 바디   |     |
| 200/성공      | 포스트 조회 성공                        | <pre> Application/json {   "pageSize": "Integer",   "pageNow": "Integer",   "elementsSize": "Integer",   "posts": [     {       "postId": "Integer",       "title": "string",       "content": "string",       "viewCount": "Integer",       "likes": "Integer",       "requestThumbnailLink": "string",       "description": "string",       "createdAt": "dateTime",       "countOfComments": "Integer",       "author": "string",       "avatar": "string"     }   ] }</pre> |     |
| 400 / 실패    | 파라미터(페이지 번호) 미존재 혹은 파라미터 타입 오류   | <pre> Application/json {   "errorMessage": "string",   "errorCode": "Integer" }</pre>   |     |
| 401 / 실패    | Unauthorized                     |   |     |
| 404 / 실패    | Not Found                        |   |     |
| 415 / 실패    | Unsupported Media Type           |   |     |

|             |                                      |   |     |
|-------------|--------------------------------------|---|-----|
| 요구사항 명칭     | 최신순으로 포스트 조회                         | HTTP Method   | GET |
| 요구사항 상세설명   | 메인페이지에서 사용하는API, 최신순으로 작성된 포스트들을 가져옴 |   |     |
| URL         | /api/posts/newest/{pageNumber}       |   |     |
| 응답 코드/ 성공여부 | 설명                                   | 응답 바디   |     |
| 200/성공      | 포스트 조회 성공                            | <pre> Application/json {   "pageSize": "Integer",   "pageNow": "Integer",   "elementsSize": "Integer",   "posts": [     {       "postId": "Integer",       "title": "string",       "content": "string",       "viewCount": "Integer",       "likes": "Integer",       "requestThumbnailLink": "string",       "description": "string",       "createdAt": "dateTime",       "countOfComments": "Integer",       "author": "string",       "avatar": "string"     }   ] }</pre> |     |
| 400 / 실패    | 파라미터(페이지 번호) 미존재 혹은<br>파라미터 타입 오류    | <pre> Application/json {   "errorMessage": "string",   "errorCode": "Integer" }</pre>   |     |
| 401 / 실패    | Unauthorized                         |   |     |
| 404 / 실패    | Not Found                            |   |     |
| 415 / 실패    | Unsupported Media Type               |   |     |

|             |  |  |     |
|-------------|--|--|-----|
| 요구사항 명칭     | Content로 포스트 조회  | HTTP Method  | GET |
| 요구사항 상세설명   | content 가 포함되어 있는 포스트들을 가져옴  |  |     |
| URL         | /api/posts/content   |  |     |
| 요청 바디       | <pre> Application/json {   "param": "string",   "pageNumber": "Integer" } </pre> |  |     |
| 응답 코드/ 성공여부 | 설명   | 응답 바디  |     |
| 200/성공      | 포스트 조회 성공  | <pre> Application/json {   "pageSize": "Integer",   "pageNow": "Integer",   "elementsSize": "Integer",   "posts": [     {       "postId": "Integer",       "title": "string",       "content": "string",       "viewCount": "Integer",       "likes": "Integer",       "requestThumbnailLink": "string",       "description": "string",       "createdAt": "dateTime",       "countOfComments": "Integer",       "author": "string",       "avatar": "string"     }   ] } </pre> |     |
| 400 / 실패    | 파라미터(페이지 번호) 미존재 혹은<br>파라미터 타입 오류  | <pre> Application/json {   "errorMessage": "string",   "errorCode": "Integer" } </pre>   |     |
| 401 / 실패    | Unauthorized   |  |     |
| 404 / 실패    | Not Found  |  |     |
| 415 / 실패    | Unsupported Media Type   |  |     |

-c. 이메일 회원가입 API

| 요구사항 명칭     | 인증 코드 전송                    | HTTP Method   | POST |
|-------------|-----------------------------|---|------|
| 요구사항 상세설명   | 입력한 이메일로 인증 코드를 전송하기 위한 API |   |      |
| URL         | /api/email                  |   |      |
| 요청 바디       | text/plain<br>"string"      |   |      |
| 응답 코드/ 성공여부 | 설명                          | 응답 바디   |      |
| 204/성공      | 정상적으로 입력한 이메일에 인증 코드 전송 성공  | none  |      |
| 400/실패      | 이메일 중복 또는 이메일 형식 오류         | application/json<br>{<br>"errorMessage": "string",<br>"errorCode": "Integer"<br>} |      |
| 415/실패      | 지원하지 않는 미디어 타입              |   |      |

| 요구사항 명칭     | 인증 코드 검증                      | HTTP Method   | POST |
|-------------|-------------------------------|---|------|
| 요구사항 상세설명   | 사용자가 요청한 인증 코드가 유효한지 검증하는 API |   |      |
| URL         | /api/email/code               |   |      |
| 요청 바디       | text/plain<br>"string"        |   |      |
| 응답 코드/ 성공여부 | 설명                            | 응답 바디   |      |
| 204/성공      | 인증 코드 검증 성공                   | none  |      |
| 400/실패      | 잘못된 인증 코드 요청                  | application/json<br>{<br>"errorMessage": "string",<br>"errorCode": "Integer"<br>} |      |
| 415/실패      | 지원하지 않는 미디어 타입                |   |      |

-d. 이메일 로그인 API

| 요구사항 명칭     | 이메일 기반 로그인  | HTTP Method  | POST |
|-------------|---|--|------|
| 요구사항 상세설명   | 회원가입한 이메일과 비밀번호를 기반으로 로그인하는 API   |  |      |
| URL         | /api/login/email  |  |      |
| 요청 바디       | <pre>application/json {   "email": "string",   "password": "string" }</pre> |  |      |
| 응답 코드/ 성공여부 | 설명  | 응답 바디  |      |
| 200/성공      | 로그인 성공  | none   |      |
| 400/실패      | 이메일 또는 비밀번호 오류  | <pre>application/json {   "errorMessage": "string",   "errorCode": "Integer" }</pre> |      |
| 415/실패      | 지원하지 않는 미디어 타입  |  |      |

-e. 로그인 상태확인 API

| 요구사항 명칭     | 로그인 상태 확인                        | HTTP Method | GET |
|-------------|----------------------------------|-------------|-----|
| 요구사항 상세설명   | 유저가 로그인한 상태였는지를 세션 기반으로 확인하는 API |             |     |
| URL         | /api/users/login-status          |             |     |
| 응답 코드/ 성공여부 | 설명                               | 응답 바디       |     |
| 200/성공      | 로그인 중이었던 유저임을 나타냄                | none        |     |
| 204/성공      | 로그인하지 않았거나 세션이 만료된 유저임을 나타냄      |             |     |

- E-R 다이어그램/DB 설계 모델(테이블 구조)



4. 구현하지 못한 기능 요구사항이 있다면 그 이유와 해결방안을 기술하십시오

| 최초 요구사항                                  | 구현 여부(미구현, 수정, 삭제 등)      | 이유(일정부족, 프로젝트 관리미비, 팀원변동, 기술적 문제 등) |
|--|---------------------------|-------------------------------------|
| 회원가입 및 로그인 시 이메일 및 OAuth2를 활용한 소셜 로그인 제공 | OAuth2를 활용한 소셜 로그인 기능 미구현 | 일정 부족                               |
| 포스트 작성시 시리즈 기능으로 특정 포스트들을 묶는 기능 제공       | 시리즈 기능 미구현                | 일정 부족                               |
| 기능과 관련된 부가적인 예외처리                        | 미구현                       | 일정 부족                               |

5. 요구사항을 충족시키지 못한 성능, 품질 요구사항이 있다면 그 이유와 해결방안을 기술하십시오.

| 분류(성능, 속도 등) 및 최초 요구사항 | 충족 여부(현재 측정결과 제시)                                   | 이유(일정부족, 프로젝트 관리미비, 팀원변동, 기술적 문제 등) |
|------------------------|---|-------------------------------------|
| 컴포넌트 리랜더링 최적화(성능 및 속도) | 현재 전체적으로 컴포넌트가 리랜더링되는 상황<br>→ 관련된 컴포넌트만 리랜더링 되도록 구현 | 기술적 문제 및 일정부족                       |

6. 최종 완성된 프로젝트 결과물(소프트웨어, 하드웨어 등)을 설치하여 사용하기 위한 사용자 매뉴얼을 작성하시오.

1. 웹 브라우저를 통하여 서비스에 접속한다
2. 모든 사용자는 블로그에 작성된 글을 읽을 수 있다.
3. 회원가입 후 로그인한 사용자는 글에 대한 작성 및 수정기능을 사용할 수 있다.

## 7. 캡스톤디자인 결과의 활용방안

현재 많은 개발자들이 개인적으로 기록하거나 공유할 만한 내용들은 블로그를 통해 보여지고 있고 협업과 같이 많은 사람들과 일을 진행할 때는 'Notion'이라는 응용 프로그램을 사용하고 있습니다. 이중에서 노션에 있는 글을 개인 혹은 회사 홍보, 공유 목적으로 블로그에 올리는 사람들은 더 쉽고 빠르게 블로그에 글을 공유함으로써 원하는 목적에 달성할 수 있다.

### 7.1 사회적 측면

간편하고 빠른 지식 공유가 가능하므로 앞으로 개발자들은 더욱 더 쉽고 빠르게 지식을 공유할 수 있는 사회적 기대효과를 얻을 수 있다.

### 7.2 경제적 측면

아직 플랫폼을 정하지 못하였거나 노션만 사용하였던 사용자들에게 플랫폼 선택이나 Copy And Paste 같은 반복적인 작업을 하는데 수행하는 시간과 비용을 아낄 수 있다는 이점이 있다.

### 7.3 예상되는 기대효과

기존의 블로그와 같은 기능을 제공함과 동시에 공유가 불편했던 노션의 내용들을 이번 프로젝트를 통하여 더 편리하고 정확하게 글의 공유방법 제공하여 사용자들의 편의성을 극대화하는 효과를 얻을 수 있다.