

# Final report

디즈 데이즈(These Days) : 최신 트렌드 큐레이션 서비스



과목 : 오픈소스 SW 활용

교수님 : 송인식 교수님

학과 : 모바일시스템공학과

학번 : 32210799, 32213306

이름 : 김수진, 이영주

이메일 : [soojin0929@dankook.ac.kr](mailto:soojin0929@dankook.ac.kr)

[dldl0010987@dankook.ac.kr](mailto:dldl0010987@dankook.ac.kr)

깃허브 주소 : <https://github.com/sooxany>

<https://github.com/0-zoo>

제출일 : 2025.01.14

# 목차

<b>1. 프로젝트 개요</b>	<b>3</b>
1.1 프로젝트명	
1.2 프로젝트 배경 및 필요성	
1.3 프로젝트 목표	
1.4 주요 타겟 사용자	
<b>2. 제공 기능</b>	<b>4</b>
2.1 기능 요구사항	
2.1.1 주요 기능 요약	
2.1.2 기능별 상세 설명	
2.2 비기능 요구사항	
2.2.1 성능	
2.2.2 보안	
2.2.3 확장성	
<b>3. 설계 및 구현 내용</b>	<b>8</b>
3.1 전체 시스템 아키텍처	
3.2 주요 구성 요소	
3.3 구현 세부 내용	
3.3.1 UML 다이어그램	
3.3.2 데이터베이스 설계	
3.3.3 API 설계	
3.3.4 디렉토리 구조 설명	
3.4 역할 분담	
<b>4. 테스트 결과</b>	<b>18</b>
4.1 테스트 화면	
4.2 테스트 평가	
4.2.1 구현 성공	
4.2.2 구현 실패 및 원인 분석	
4.2.3 개선 방안	
<b>5. 참고 문헌</b>	<b>23</b>
<b>6. 후기</b>	<b>24</b>
6.1 32210799 김수진 후기	
6.2 32213306 이영주 후기	

# 1. 프로젝트 개요

## 1.1 프로젝트명

디즈 데이즈 (These Days) : 최신 트렌드 큐레이션 서비스

## 1.2 프로젝트 배경 및 필요성

현대 사회에서 정보의 양은 기하급수적으로 증가하고 있으며, 특히 뉴스와 SNS에서는 최신 정보와 트렌드가 빠르게 변한다. 이에 오히려 사용자는 시간을 들여 다양한 매체에서 정보를 찾고, 또 그 정보를 선별하는 데 많은 노력을 기울여야 하는 현실이다. 이러한 과정에서 과도한 정보에 의한 사용자의 피로도 증가, 편향된 정보 등의 문제도 발생하고 있다. 또한, 각 언론사나 SNS 플랫폼에서 제공되는 정보는 종종 특정한 시각이나 편향을 가지기도 한다.

이에 따라, 빠르게 변화하는 트렌드와 실시간 정보를 실시간으로 제공하면서도 중립적이고 다양한 출처의 정보를 제공하는 서비스가 필요하다. 우리는 이러한 문제를 해결하기 위해, 사용자에게 최소한의 시간 투자로 최신 정보를 제공하는 서비스인 '디즈 데이즈(These Days)'를 구상하였다.

## 1.3 프로젝트 목표

해당 프로젝트는 사용자가 최신 정보와 트렌드를 실시간으로 확인할 수 있도록 지원하는 서비스이다. 핵심 목표는 사용자에게 편향되지 않은 다양한 정보를 빠르게 제공하는 것이다.

## 1.4 주요 타겟 사용자

해당 프로젝트의 주요 타겟 사용자는 다음과 같다.

- 정보의 빠른 습득을 원하는 사용자 : 바쁜 일정을 가진 직장인이나 학생들과 같이 최소한의 시간 투자로 최신 정보를 실시간으로 보길 원하는 사용자.
- 편향되지 않은 정보를 원하는 사용자 : 다양한 출처에서 정보를 확인하고 싶지만 특정 매체나 플랫폼에 의존하지 않기를 원하는 사용자.
- 트렌드에 민감한 사용자 : SNS와 뉴스 등 다양한 정보를 통해 최신 트렌드를 파악하려는 사용자.

## **2. 제공 기능**

### **2.1 기능 요구사항**

#### **2.1.1 주요 기능 요약**

##### **1. 실시간 핫 뉴스 제공**

목적) 사용자가 최신 뉴스를 실시간으로 확인하고 필요한 정보를 빠르게 얻을 수 있도록 지원하는 것이다.

- 다양한 뉴스 사이트에서 최신 정보를 크롤링하여 자동으로 업데이트하고 사용자에게 실시간으로 제공한다.
- 각 뉴스는 간단한 요약과 함께 제공되며, 클릭 시 전문 내용을 확인 가능하다.

##### **2. 핫 유튜브 동영상 제공**

목적) 트렌드에 민감한 사용자에게 최신 인기 동영상을 제공한다.

- 유튜브 인기 급상승 동영상 순위를 바탕으로 동영상 목록을 제공한다.
- 각 동영상은 간단한 설명과 링크를 포함하며, 클릭 시 동영상을 바로 시청이 가능하다.

##### **3. 콘텐츠 요약 제공**

목적) 많은 정보를 빠르게 소비하려는 사용자를 위해 요약된 정보를 제공한다.

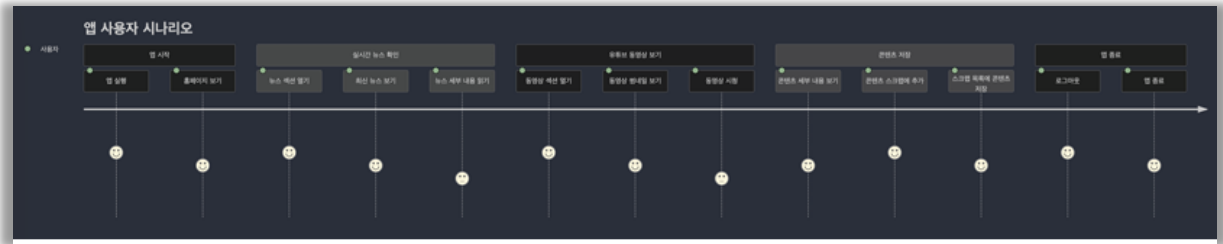
- 주요 뉴스 기사와 웹사이트를 크롤링한 데이터를 간단히 요약하여 사용자에게 제공한다.
- 최소한의 시간으로 핵심 정보를 파악할 수 있도록 지원한다.

##### **4. 스크랩 기능**

목적) 관심 있는 콘텐츠를 저장하여 재방문할 수 있도록 지원한다.

- 사용자가 관심 있는 콘텐츠를 저장하여 나중에 다시 확인 가능하다.
- 스크랩 항목은 관리 기능(삭제, 수정 등)을 제공한다.

## 2.1.2 기능별 상세 설명 및 앱의 흐름



### 1. 앱 시작 및 로그인/회원가입

- 기능 세부사항
  - 회원가입:** 새 사용자가 이메일, 비밀번호, 이름 등을 입력하여 계정을 생성한다.
  - 로그인:** 기존 사용자가 이메일과 비밀번호를 입력해 자신의 계정으로 접근한다.
  - JWT 기반 인증:** 사용자의 로그인 상태를 유지하기 위해 보안 토큰(JWT)을 사용한다.
- 사용 흐름
  - 앱 실행 시 첫 화면에 로그인 화면이 표시된다.
  - 2-1. 기존 사용자 - 아이디와 비밀번호 입력 후 '로그인' 버튼 클릭한다.
  - 2-2. 새 사용자 - '회원가입' 버튼을 눌러 필요한 정보를 입력 후 계정을 생성. 회원가입 성공 시 로그인 화면으로 이동한다.
  - 로그인 성공 시 메인 화면으로 이동한다.

### 2. 실시간 핫 뉴스 제공 (메인 화면)

- 기능 세부사항
  - 뉴스 크롤링:** 주요 뉴스 사이트에서 자동으로 정보를 수집하여 제공한다.
  - 실시간 업데이트:** 새로운 뉴스가 입력되면 즉시 리스트에 반영한다.
  - 간단 요약 제공:** GPT API 를 이용하여 뉴스 본문 요약을 통해 사용자가 빠르게 이해할 수 있도록 지원한다.
  - 링크 연결:** 클릭 시 원문 페이지로 이동하여 상세 내용을 확인 가능하다.

- 사용 흐름

1. 로그인 후 바로 보이는 메인 화면 역할을 한다.
2. 최신 뉴스 리스트를 탐색하며 원하는 기사를 클릭한다.
3. 클릭 시 상세 페이지 또는 원문 사이트로 연결된다.
4. 새로 고침 시 새로운 뉴스 업데이트된다.
5. 우측의 별표 모양의 아이콘을 클릭하여 노란색 별로 바뀌면 스크랩 성공이다.

### 3. 핫 유튜브 동영상 제공

- 기능 세부사항

- **유튜브 인기 영상 데이터 수집:** 유튜브 API 를 활용하여 실시간으로 급상승 동영상을 수집한다.
- **실시간 순위 반영:** 조회수, 좋아요, 트렌드 데이터를 기준으로 인기 목록을 갱신한다.
- **동영상 설명 및 바로가기 링크 제공:** 각 동영상에 제목, GPT API 를 이용한 요약 설명한다.

- 사용 흐름

1. 메인 화면에 들어온 후, 하단의 네비게이션 바에서 Youtube 버튼 클릭 시 핫 유튜브 동영상 제공 화면으로 이동한다.
2. 인기 동영상 목록을 확인하며 원하는 동영상 클릭한다.
3. 클릭 시 유튜브 앱 또는 웹 브라우저로 연결되어 동영상 시청 가능하다.
4. 우측의 별 모양의 아이콘을 클릭하여 노란색 별로 바뀌면 스크랩 성공이다.

### 4. 스크랩 기능

- 기능 세부사항

- **스크랩 저장:** 사용자가 콘텐츠를 탐색 중 별 모양의 아이콘 클릭 시 해당 항목이 저장된다.
- **스크랩 관리:** 스크랩한 콘텐츠는 목록에서 삭제 가능하다.
- **스크랩 목록:** 저장된 모든 항목을 한 화면에서 확인 가능하다.
- **사용 흐름**
  1. 사용자가 뉴스나 유튜브 동영상을 탐색하다가 관심 있는 콘텐츠 발견한다.
  2. 별 모양의 아이콘을 눌러 저장한다.
  3. 스크랩 목록에서 저장된 콘텐츠를 확인하거나 삭제한다.

## 5. 콘텐츠 요약 제공

- **기능 세부사항**
  - **데이터 크롤링 :** 주요 뉴스 및 유튜브 콘텐츠를 수집한다.
  - **GPT API 연동 :** AI 를 활용하여 뉴스 기사나 유튜브 동영상 주요 내용을 요약한다.
  - **요약본 제공 :** 간단한 텍스트를 통해 사용자에게 제공한다.

## 6. 앱 종료

- 화면 종료 시 앱도 종료된다.

## 2.2 비기능 요구사항

### 2.2.1 성능

실시간 정보 제공 : 클라이언트의 요청 시 최신 콘텐츠를 빠르게 제공하며, 서버는 빠르고 안정적인 응답을 보장한다.

- **대용량 데이터 처리 :** 유튜브 인기 동영상, 뉴스 요약 등의 데이터를 실시간으로 크롤링하고 요약하여 사용자에게 즉시 제공한다.

### 2.2.2 보안

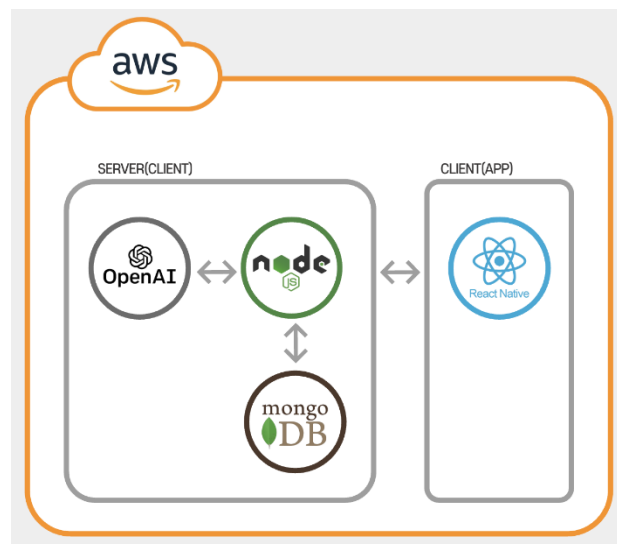
- 사용자 인증 및 개인정보 보호 : 로그인 기능을 제공하고, 사용자의 개인정보는 안전하게 보호한다. 또한 비밀번호는 JWT 방식을 이용하여 암호화 후 DB 에 저장된다.
- 데이터 암호화 : 데이터 전송 시 SSL/TLS 암호화를 사용하여 보안을 강화한다.

### 2.2.3 확장성

- 플랫폼 확장성 : ios 와 안드로이드 등과 같은 모바일 디바이스에서 해당 서비스가 원활히 실행되도록 한다.
- 기능 확장성 : 국내 콘텐츠 뿐만 아니라 해외 콘텐츠 제공, 혹은 전문화된 콘텐츠 제공을 위한 시스템 확장성을 보장한다.

## 3. 설계 및 구현 내용

### 3.1 아키텍처 다이어그램



#### 1. AWS (외부 인프라)

- 애플리케이션 전체가 AWS 환경에서 호스팅되고 실행된다.
- EC2, S3, 또는 Lambda 같은 AWS 서비스를 사용한다.

#### 2. Server (Client)



- 애플리케이션의 서버 부분을 구성하며, 백엔드와 데이터베이스, 외부 API 가 포함된 주요 처리 영역이다.
- 구성 요소
  - OpenAI
    - GPT API 를 활용한 외부 서비스로, 텍스트 요약과 같은 AI 기능을 제공한다.
    - Node.js 서버와 직접 통신하며 AI 기반 기능을 수행한다.
  - Node.js
    - 백엔드 서버로, 클라이언트 요청을 처리하고 외부 API(OpenAI) 및 데이터베이스(MongoDB)와 상호작용한다.
    - 애플리케이션의 비즈니스 로직을 처리한다.
  - MongoDB
    - 애플리케이션 데이터베이스로, 사용자 정보, 스크랩 데이터, 기타 애플리케이션 관련 데이터를 저장하고 관리한다.

### 3. Client (App)

- 애플리케이션의 프론트엔드 부분으로, 사용자와 직접 상호작용하는 React Native 기반의 모바일 애플리케이션이다.
- 구성 요소
  - React Native
    - 크로스 플랫폼 모바일 애플리케이션 프레임워크로, 사용자 인터페이스를 제공한다.
    - 백엔드(Node.js)와 REST API 를 통해 통신하며, 데이터를 요청하거나 수신한다.

## 3.2 주요 구성 요소

### 1. 프론트엔드

- 기술 스택 : React Native (Expo 프레임워크).
- 역할 : 사용자 인터페이스(UI) 제공 및 사용자와 백엔드 간의 데이터 통신.

- 주요 기능
  - 뉴스 및 동영상 리스트 제공.
  - 사용자 입력(로그인/회원가입, 스크랩 등) 처리.
  - REST API 호출을 통해 데이터를 가져와 화면에 렌더링.

## 2. 백엔드

- 기술 스택 : Node.js(Express 사용)
- 역할 : 데이터 처리, API 요청 처리, 비즈니스 로직 실행
- 주요 기능
  - 사용자 인증(JWT 토큰).
  - 크롤링된 뉴스와 동영상 데이터를 클라이언트에 제공.
  - 스크랩 데이터를 관리 및 저장.
  - 외부 API(GPT API)와 통합.

## 3. 크롤러

- 기술 스택 : Python
- 역할 : 웹 크롤링을 통해 실시간 뉴스 및 유튜브 데이터 수집
- 주요 기능
  - 여러 뉴스 사이트 및 YouTube 에서 데이터를 가져옴.
  - 데이터 정리 후 데이터베이스에 저장.
  - 주기적인 크롤링 작업 스케줄링

## 4. 데이터베이스

- 기술 스택 : MongoDB
- 역할 : 구조화된 데이터 저장 및 관리
- 주요 데이터 구조
  - 사용자 데이터(로그인 정보)
  - 뉴스 및 유튜브 동영상 콘텐츠 데이터
  - 스크랩 데이터(사용자가 저장한 항목)

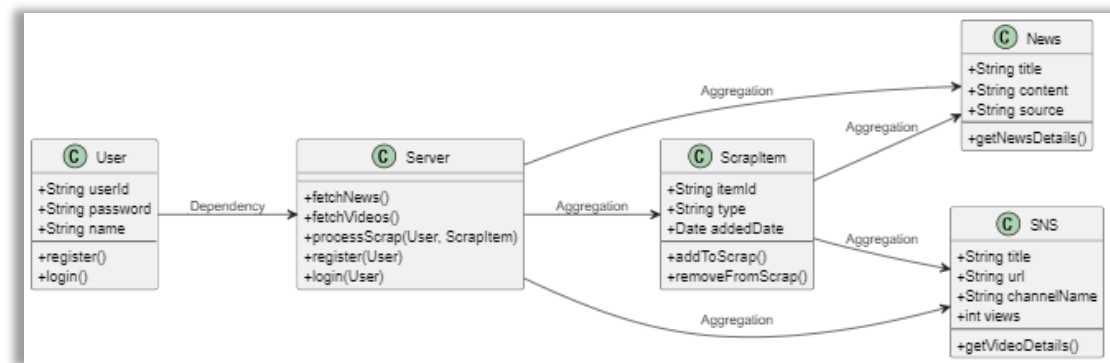
## 5. 외부 API

- GPT API : 콘텐츠 요약 기능 제공

- Youtube API : 인기 동영상 데이터를 가져오기 위한 API 호출

## 3.3 구현 세부 내용

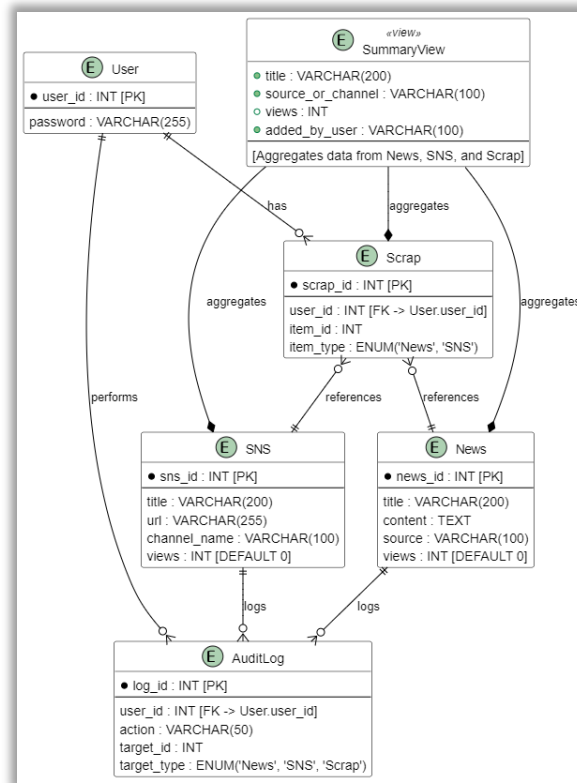
### 3.3.1 UML 다이어그램



UML 다이어그램은 위와 같다. 구체적인 디렉토리 구조와 각 파일의 역할은 3.3.4 디렉토리 구조에서 설명하겠다.

### 3.3.2 데이터베이스 설계

아래의 ERD(Entity-Relationship Diagram)은 데이터베이스 구조를 시각화한 것이다. 주요 테이블과 그들 간의 관계를 나타내고 있다.



#### 1. User 테이블

- 사용자를 나타낸다.
- user\_id : 기본 키(Primary Key)로, 각 사용자를 고유하게 식별한다.
- Password : 사용자의 비밀번호를 저장한다.

#### 2. News 테이블

- 뉴스 항목을 저장한다.
- news\_id : 기본 키이다.
- title, content, source 와 같은 뉴스 관련 정보를 포함한다.

#### 3. SNS 테이블

- Youtube 항목을 저장한다.
- sns\_id : 기본 키이다.
- title, url, channel\_name 과 같은 SNS 관련 정보를 포함한다.

#### 4. Scrap 테이블

- 사용자가 스크랩한 뉴스나 SNS 항목을 저장한다.
- scrap\_id : 기본 키이다.
- user\_id : User 테이블과 연결되며, 스크랩을 한 사용자를 나타낸다.
- item\_id 와 item\_type 은 스크랩된 항목과 그 ID 를 저장한다.

#### 5. SummaryView (가상 뷰)

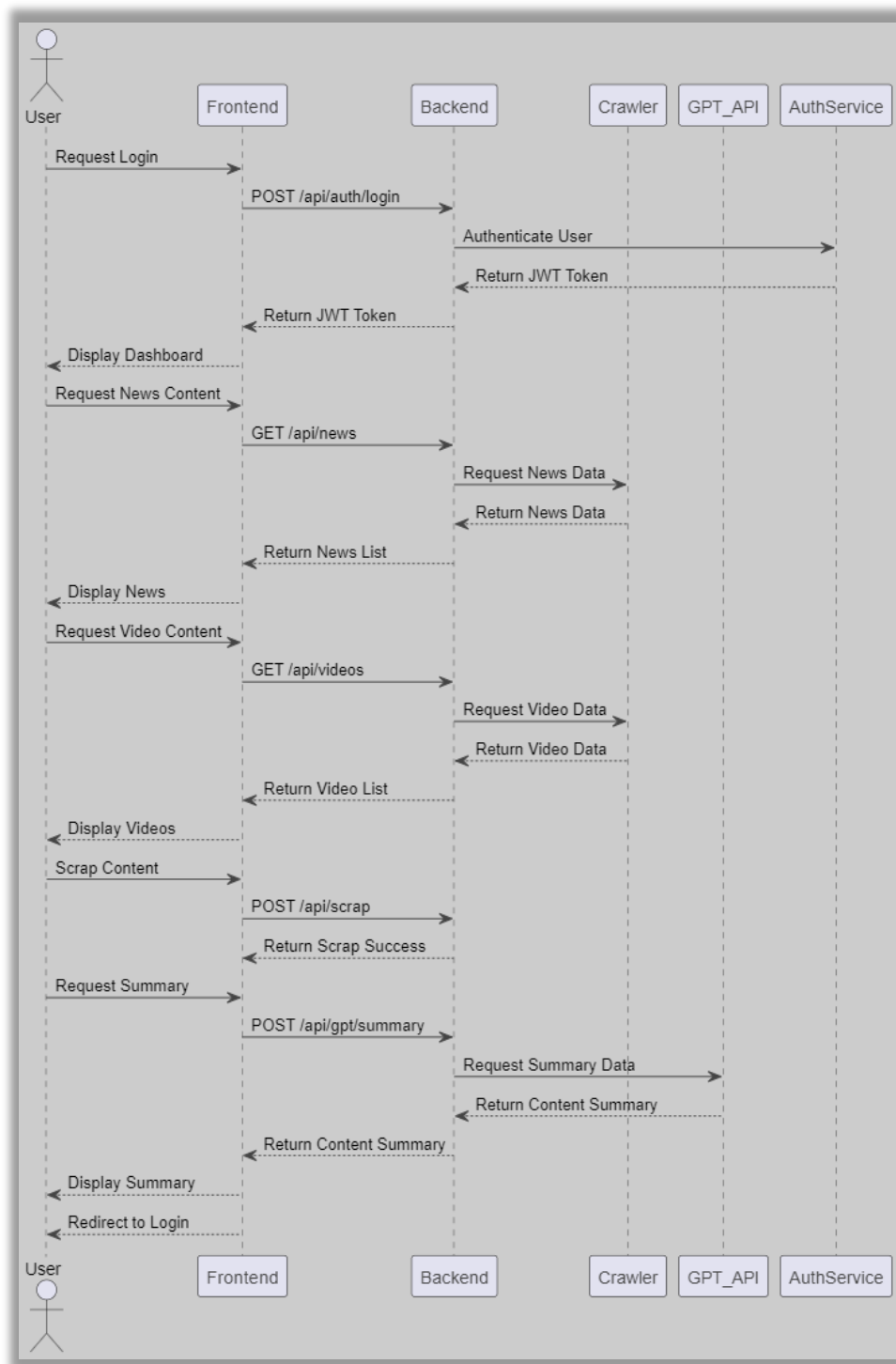
- News 와 SNS, Scrap 데이터를 종합하여 요약 정보를 제공한다.
- title, source\_or\_channel, views, added\_by\_user 등의 정보를 포함한다.

#### 6. AuditLog 테이블

- 사용자의 주요 행동 로그를 기록한다.
- log\_id : 기본 키이다.
- user\_id : User 테이블과 연결되어 로그를 남긴 사용자를 나타낸다.
- Action : 사용자가 수행한 작업을 기록한다.
- target\_id : 관련된 항목(News, SNS, Scrap 등)의 ID 를 저장한다.
- target\_type: 해당 항목의 유형을 나타낸다.

### 3.3.3 API 설계

아래의 시퀀스 다이어그램은 사용자와 시스템 간의 상호작용 과정을 단계적으로 나타낸다. 각 구성 요소는 Frontend, Backend, Crawler, GPT\_API, AuthService 로 나뉘어 있으며, 주요 기능 흐름을 설명한다.



**1. 사용자 로그인 요청** : 사용자가 로그인을 요청한다.

- Frontend 는 /api/auth/login 엔드포인트로 Backend 에 로그인 요청을 보낸다.
- Backend 는 AuthService 를 통해 사용자를 인증한다.
- 인증이 완료되면, AuthService 가 JWT 토큰을 반환하고, Backend 는 이를 Frontend 에 전달한다.
- Frontend 는 토큰을 저장하고, 대시보드를 표시한다.

**2. 뉴스 콘텐츠 요청** : 사용자가 뉴스 콘텐츠를 요청한다.

- Frontend 는 /api/news 엔드포인트로 Backend 에 요청을 보낸다.
- Backend 는 Crawler 를 호출하여 뉴스 데이터를 수집한다.
- Crawler 는 뉴스 데이터를 반환하고, Backend 는 이를 Frontend 에 전달한다.
- Frontend 는 사용자가 볼 수 있도록 뉴스 리스트를 표시한다.

**3. 비디오 콘텐츠 요청** : 사용자가 비디오 콘텐츠를 요청한다.

- Frontend 는 /api/videos 엔드포인트로 Backend 에 요청을 보낸다.
- Backend 는 Crawler 를 호출하여 비디오 데이터를 수집한다.
- Crawler 는 비디오 데이터를 반환하고, Backend 는 이를 Frontend 에 전달한다.
- Frontend 는 사용자가 볼 수 있도록 비디오 리스트를 표시한다.

**4. 스크랩 콘텐츠 요청** : 사용자가 특정 콘텐츠를 스크랩한다.

- Frontend 는 /api/scrap 엔드포인트로 Backend 에 스크랩 요청을 보낸다.
- Backend 는 요청을 처리하고 성공 상태를 Frontend 에 반환한다.
- Frontend 는 성공 메시지를 표시한다.

**5. 요약 요청** : 사용자가 콘텐츠 요약을 요청한다.

- Frontend 는 /api/gpt/summary 엔드포인트로 Backend 에 요약 요청을 보낸다.
- Backend 는 GPT\_API 를 호출하여 요약 데이터를 생성한다.

- GPT\_API 는 요약 데이터를 반환하고, Backend 는 이를 Frontend 에 전달한다.
- Frontend 는 요약 내용을 사용자에게 표시한다.

### 3.3.4 디렉토리 구조

#### 1. root 디렉토리

- .gitignore : Git 버전 관리에서 제외할 파일 및 디렉토리를 지정.
- package.json, package-lock.json : 종속성과 스크립 관리 및 Node.js 환경에서 필요한 의존성 명시
- requirements.txt : Python 프로젝트에서 필요한 모든 라이브러리와 패키지 의존성 명시

#### 2. Backend

- src/
  - config/db.js: 데이터베이스 연결 설정 파일. MongoDB 연결을 관리.
  - controllers/authController.js: 사용자 인증과 관련된 로직 처리 (예: 로그인, 회원가입).
  - models/User.js: 사용자 데이터 스키마 정의. MongoDB 와 상호작용.
  - routes/
    - auth.js : 사용자 인증 관련 API 경로 정의.
    - userRoutes.js : 사용자와 관련된 기타 기능(API 경로) 정의.
  - index.js : 모든 라우트를 통합하는 메인 파일.
  - news.py : Python 으로 작성된 크롤러 스크립트로, 뉴스 데이터를 수집.
- .env : 환경 변수 파일로 데이터베이스 URI, JWT 시크릿 키 등 민감한 정보를 저장.
- node\_modules/ : Node.js 환경에서 사용하는 모든 패키지가 설치된 디렉토리.



- package.json, package-lock.json : 종속성과 스크랩 관리 및 Node.js 환경에서 필요한 의존성 명시
- .gitignore : Git 버전 관리에서 제외할 파일 및 디렉토리를 지정.

### 3. Frontend

- .expo/ : Expo 프로젝트 설정 파일 디렉토리.
- Assets/ : 애플리케이션에서 사용하는 정적 파일(이미지, 아이콘 등)을 저장.
- Components/ : 재사용 가능한 UI 컴포넌트 저장.
- node\_modules : Node.js 환경에서 사용하는 모든 패키지가 설치된 디렉토리.
- Screens
  - loginScreen.js : 로그인 화면을 구성하는 컴포넌트.
  - registerScreen.js : 회원가입 화면을 구성하는 컴포넌트.
  - mainScreen.js : 메인 화면(뉴스와 동영상 리스트 제공)
  - snsScreen.js : 유튜브 실시간 급상승 동영상 리스트 제공.
  - scrapScreen.js : 사용자가 스크랩한 콘텐츠 리스트 제공.
- Styles
  - globalStyles.js : 공통 스타일을 정의한 파일.
  - Theme.js : 애플리케이션 테마 설정 (컬러, 폰트 등)
  - scrapStyles.js : 스크랩 화면 전용 스타일 정의.
- Utils/
  - api.js : 백엔드와 통신하기 위한 API 호출 함수를 정의하는 파일.
  - Constants.js : 프로젝트에서 공통으로 사용하는 상수 값을 관리하는 파일
  - localStorage.js : 사용자의 스크랩 데이터나 로그인 토큰을 로컬 스토리지에 저장 및 관리
- .gitignore : Git 버전 관리에서 제외할 파일 및 디렉토리를 지정.
- App.js : 애플리케이션의 루트 파일로 화면 전환과 전체 구조 관리.

- App.json : Expo 앱의 기본 설정 파일로, 앱 이름, 아이콘, 화면 방향 등 전반적인 앱 설정을 정의.
- Babel.config.js : 프로젝트의 Babel 설정 파일로, JavaScript 문법 변환을 관리.
- Index.js : React Native 앱의 진입점으로, App.js를 불러와 애플리케이션을 실행하는 역할.
- Package-lock.json, package.json : 종속성과 스크랩 관리 및 Node.js 환경에서 필요한 의존성 명시.

### 3.4 역할 분담

프로젝트 기간이 짧았던 만큼 팀원 간의 역할 분담을 세부적으로 나누기보다는 모든 구성원이 협력하여 구현에 참여하였다. 그러나 각자 더 주도적으로 기여한 영역을 기준으로, 프로젝트에서의 기여도를 나누어 설명하고자 한다.

- **32210799 김수진** : **프론트엔드** 파트를 담당하였다.

프론트엔드 구현은 React Native와 Expo 프레임워크를 사용하여 진행되었다. 이를 통해 크로스플랫폼 환경에서 애플리케이션을 효율적으로 개발할 수 있었다. 주요 구현 내용으로는 로그인 및 회원가입, 뉴스와 동영상 데이터 표시, 스크랩 기능, 콘텐츠 요약 기능을 하였다. 또한, REST API를 활용하여 실시간으로 뉴스와 동영상 데이터를 받아와 화면에 동적으로 렌더링하였다.

- **32213306 이영주** : **백엔드** 파트를 담당하였다.

Node.js를 기반으로 설계되었으며, 데이터 처리, 외부 API와의 통신을 담당했다. 주요 기능으로는 JWT를 활용한 사용자 인증, 뉴스 및 동영상 데이터를 제공하는 REST API, 스크랩 데이터를 관리하는 API, GPT API와 연동한 콘텐츠 요약 기능이 포함된다. 데이터 저장은 MongoDB를 사용하여 관리하였고, Python으로 작성된 크롤러를 통해 실시간 뉴스와 유튜브 데이터를 주기적으로 수집했다.

## 4. 테스트 결과

### 4.1 테스트 화면

터미널 창에 `npm run dev` 명령어를 입력하여 백엔드 서버를 실행한다. 아래의 터미널 창의 결과처럼 Server running 과 MongoDB connected 가 출력되면 백엔드 서버가 정상적으로 실행된 것이다.

```
PS C:\24-겨울\oss\OSS_test> cd backend
PS C:\24-겨울\oss\OSS_test\backend> npm run dev
>>

> backend@1.0.0 dev
> nodemon src/index.js

[nodemon] 3.1.9
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node src/index.js`
Server running on http://localhost:8000
MongoDB Connected...
```

백엔드가 실행된 상태에서 `npx expo start` 명령어를 입력하면 프론트엔드가 실행이 된다. Reactive native expo 를 사용하고 있기에 QR 코드가 뜨는데 이 QR 코드를 통해 앱이 실행된다. 이때, ios 혹은 안드로이드 앱 스토어에서 'expo-go' 애플리케이션을 미리 다운받아놔야한다.

```
PS C:\24-겨울\oss\OSS_test> cd frontend
PS C:\24-겨울\oss\OSS_test\frontend> npx expo start
Starting project at C:\24-겨울\oss\OSS_test\frontend
Starting Metro Bundler
The following packages should be updated for best compatibility with the installed expo version:
expo@52.0.24 - expected version: ~52.0.25
expo-status-bar@2.0.0 - expected version: ~2.0.1
react-native@0.76.5 - expected version: 0.76.6
Your project may not work correctly until you install the expected versions of the packages.

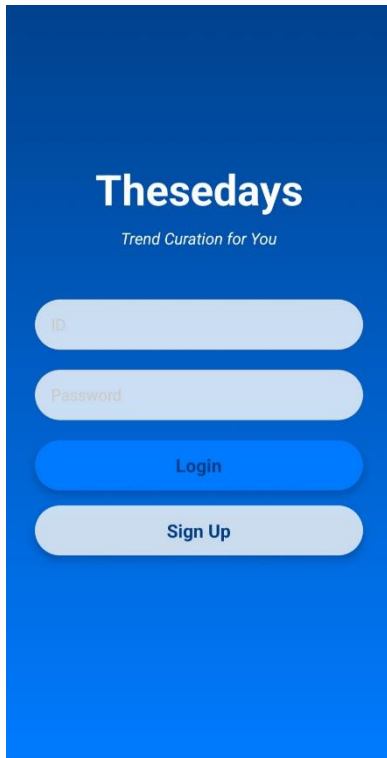
[QR Code]

> Metro waiting on exp://172.31.74.180:8081
> Scan the QR code above with Expo Go (Android) or the Camera app (iOS)
> Web is waiting on http://localhost:8081
> Using Expo Go
> Press s | switch to development build
> Press a | open Android
> Press w | open web
```

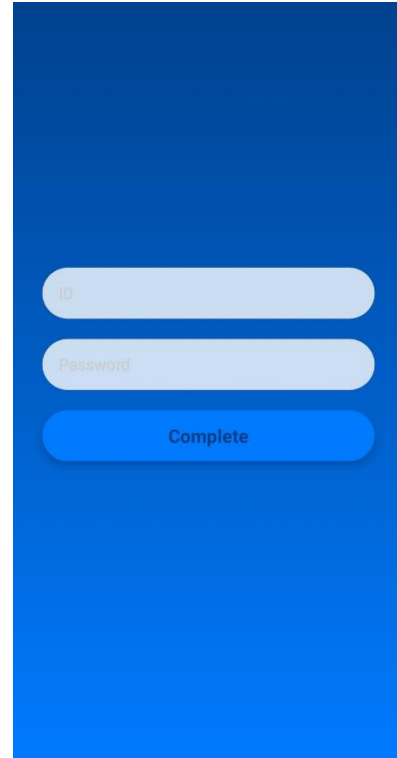


Expo-go 애플리케이션

앱의 첫 화면은 1-1 로그인 화면이다. 만약 기존 계정이 있다면 바로 login 을 하면 되고, 계정이 존재하지 않는다면 Sign Up 버튼을 눌러 1-2 회원가입 화면으로 넘어가 회원가입을 진행한다.

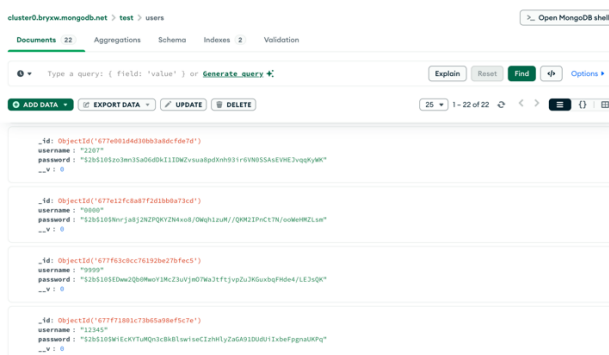


1-1. 로그인 화면



1-2. 회원가입 화면

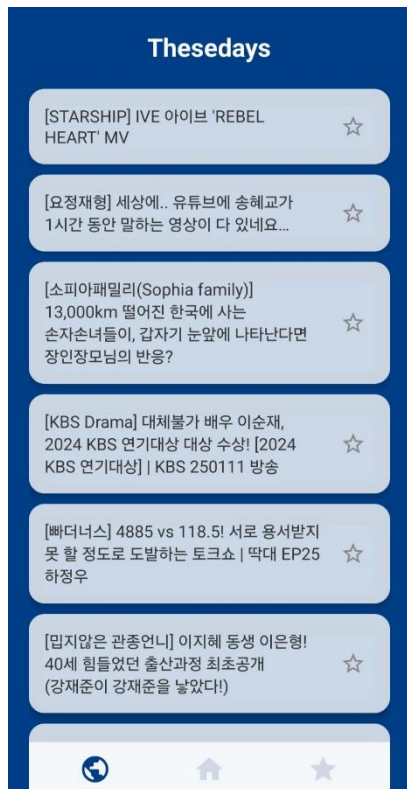
회원가입을 성공하면 MongoDB 에 회원가입을 한 계정이 저장된 것을 확인할 수 있다. 비밀번호는 JWT 토큰은 이용하여 암호화된 후 저장된다. 이후 1-1 로그인 화면으로 넘어간다.



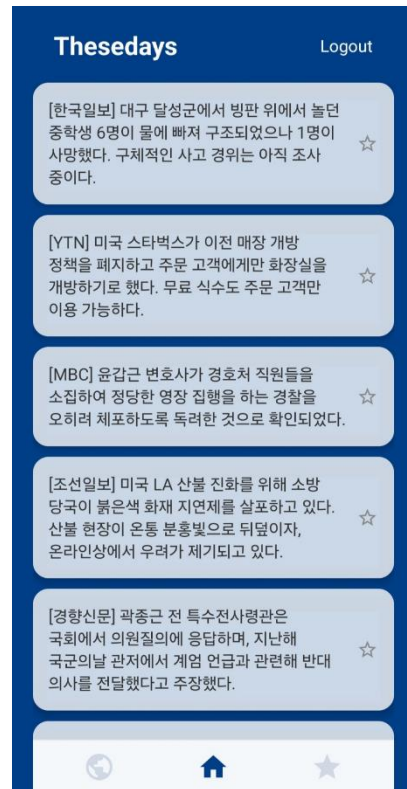
Document	_id	username	password
1	ObjectID('677e082d4630bb3abdcf6d7d')	'2288'	'52851051e03b035a060d4111262vua0pd0x031r0VW055AeVME3uqy4W'
2	ObjectID('677e12fcb87f2d3b0a73cd')	'9999'	'52851051e03b035a060d4111262vua0pd0x031r0VW055AeVME3uqy4W'
3	ObjectID('677f63b0cc7632b27bfc5')	'9999'	'52851051e03b035a060d4111262vua0pd0x031r0VW055AeVME3uqy4W'
4	ObjectID('677f73803c73b05a8bfc7e')	'12345'	'52851051e03b035a060d4111262vua0pd0x031r0VW055AeVME3uqy4W'

DB 에 계정이 저장된 모습

로그인에 성공하면 메인 화면이 뜬다. 메인 화면은 2-2 실시간 핫 뉴스 제공 화면이다. 하단의 네비게이션 바의 버튼을 통해 2-1 실시간 핫 동영상 제공 화면으로 이동할 수 있다.



2-1. 실시간 핫 동영상 제공 화면



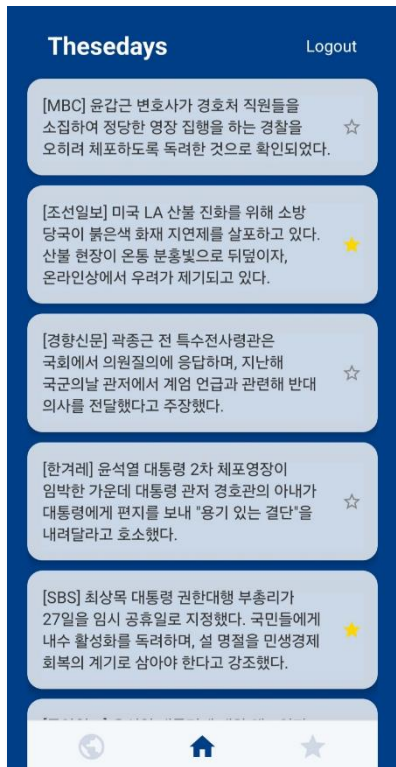
2-2. 실시간 핫 뉴스 제공 화면

앱 화면을 보면 뉴스 내용과 동영상 내용이 잘 요약되었음을 확인할 수 있다.

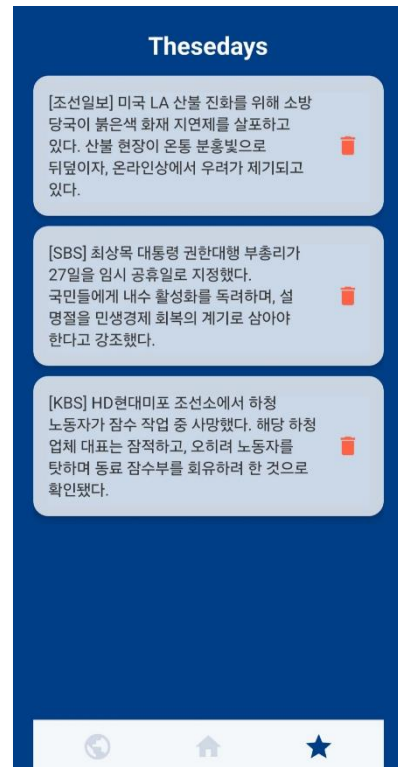
사용자가 원하는 콘텐츠를 스크랩 하고 싶으면 우측의 별 모양의 아이콘을 클릭하면 된다.

별 모양의 아이콘을 클릭하면 아래의 3-1 화면과 같이 별 아이콘이 노란색으로 변한다. 이는 스크랩이 되었다는 표시로 3-2 스크랩 화면을 확인해보면, 스크랩 목록이 저장된 것을 확인할 수 있다.

3-2 스크랩 화면에서 박스의 우측에 쓰레기통 아이콘을 누르면 스크랩한 내용이 스크랩 목록에서 삭제된다.



3-1. 스크랩한 뉴스 화면



3-2. 스크랩 화면

## 4.2 테스트 평가

### 4.2.1 구현 성공

제안서를 기준으로 이번 프로젝트 테스트 결과를 평가해보겠다.

#### 1. 로그인/회원가입 기능

- 회원가입을 하고 MongoDB 를 확인하면 회원가입을 한 계정이 존재한다.
- MongoDB 에 저장되어있는 사용자 계정으로 로그인 할 시 다음 화면으로 넘어간다.

#### 2. 뉴스 및 유튜브 콘텐츠 요청 기능

- 크롤링한 뉴스와 유튜브가 스크롤 바 형태로 뜬다.
- 바 안에는 요약 내용이 있고 바를 클릭 시 원문 링크로 넘어간다.

#### 3. 콘텐츠 스크랩 기능

- 뉴스 및 유튜브 콘텐츠 화면에서 스크랩 하고 싶은 내용의 별 모양 아이콘을 누르면 스크랩 화면에 해당 콘텐츠가 바 형태로 뜬다.
- 스크랩 콘텐츠를 삭제하고 싶을 시 쓰레기통 아이콘을 누르면 삭제된다.

#### 4. 요약 기능

- GPT API 를 통해 요약된 내용이 백엔드 API 를 호출하여 데이터를 받아와 바 안에 안전하게 렌더링 된다.

#### 5. AWS 배포

- EC2 인스턴스를 사용하여 백엔드 서버를 호스팅하고, S3 버킷을 활용하여 프론트엔드 정적 파일 배포를 성공했다.

### 4.2.2 구현 실패 및 원인 분석

프로젝트 초기에 **인스타그램 크롤링**을 통해 실시간 인기 콘텐츠와 사용자 맞춤형 데이터를 수집하여 트렌드 분석 기능을 제공하고자 하였다. 이를 통해 뉴스 및 유튜브 데이터를 포함한 소셜 미디어 전반의 데이터를 다룰 수 있는 통합 플랫폼을 구현하려는 계획이었다. 그러나 인스타그램 크롤링 과정에서 다음과 같은 문제가 발생하였다.

#### 1. 기술적 제한

- (1) 인스타그램의 크롤링 방지 메커니즘 : 비공개 API 와 복잡한 인증 절차(CSRF 토큰, 세션 쿠키)를 우회하기 어려웠다.
- (2) 로그인 요구 : 대부분의 데이터가 로그인된 세션에서만 접근 가능하며, 비공개 계정 데이터는 수집 불가능했다.

#### 2. 법적/정책적 문제

- (1) 인스타그램의 데이터 수집 정책 : 인스타그램은 서비스 약관에서 비인가 크롤링을 금지하고 있다.  
→ 크롤링 시도 중 IP 차단 또는 계정 정지의 위험이 있어 개발 중단했다.

### 4.2.3 개선 방안

#### 1. 공식 API 활용

- Instagram Graph API
  - 인증된 방식으로 인스타그램 데이터를 수집 가능하다.

- 공개 계정의 데이터(게시물, 좋아요 수 등)를 수집하여 서비스 약관 준수한다.
- 활용 전략
  - Graph API 를 통한 데이터 수집 및 분석 모듈 개발한다.
- 2. 데이터 수집 대안
  - 공개 데이터 활용
    - 인스타그램 크롤링 대신, 데이터 분석을 위해 트위터, 레딧 등 다른 공개 플랫폼 활용한다.
  - 유사 데이터 생성
    - 머신러닝 기반으로 인스타그램 데이터를 추정하거나 대체 가능한 소셜 데이터 제공한다.
- 3. 법적 검토
  - 데이터 수집과 관련된 법적 제약을 사전 검토하여 안정적인 서비스 운영 보장한다.

## 5. 참고 문헌

GPT-3 API Documentation: OpenAI. <https://beta.openai.com/docs/>

React Native Documentation: React Native. <https://reactnative.dev/docs/getting-started>

Node.js Documentation: Node.js. <https://nodejs.org/en/docs/>

MongoDB Documentation: MongoDB. <https://www.mongodb.com/docs/>

Instagram API: <https://developers.facebook.com/products/instagram/apis/>

[논문] Kunsan National Univ., *A proposal on a proactive crawling approach with analysis of state-of-the-art web crawling algorithms*. 2019.



## 6. 후기

### 6.1 32210799 김수진 후기

이번 프로젝트를 진행하며 많은 것을 배울 수 있는 소중한 경험을 했습니다. 특히, 교수님께서 제공해주신 수업 내용과 지도가 프로젝트의 전 과정을 이해하고 실행하는 데 큰 도움이 되었습니다. 이 수업을 통해 배운 이론과 기술들은 실무와 밀접하게 연관되어 있었으며, 실제 프로젝트를 진행하며 그 중요성을 체감할 수 있었습니다.

수업 시간에 배운 **Git의 활용 방법**은 프로젝트를 체계적으로 관리하고 팀원들과 효율적으로 협업하는 데 큰 도움이 되었습니다. 브랜치 관리와 커밋 메시지 작성의 중요성을 프로젝트 과정에서 깊이 이해할 수 있었습니다.

또한, 프로젝트를 진행하며 **디렉토리 구조를 체계적으로 설계**하는 것이 얼마나 중요한지를 깨달았습니다. 명확한 구조는 코드의 가독성을 높이고, 작업 간 충돌을 최소화하며, 팀원들과의 원활한 협업을 가능하게 했습니다. 특히, 서버 측에서 **package.json 파일과 같은 프로젝트 설정 파일**의 중요성을 느꼈습니다. 이런 파일들을 통해 데이터베이스 설정, API 키 관리, 네트워크 연결 등을 체계적으로 관리할 수 있었으며, 이를 통해 서버 환경을 효율적으로 구성할 수 있었습니다. 잘 구성된 설정 파일은 네트워크 연결 문제를 해결하고, 개발 및 배포 환경을 명확히 구분하는 데 큰 도움을 주었습니다.

프로젝트를 진행하는 동안 어려운 순간들도 있었지만, 교수님께서 가르쳐주신 체계적인 문제 해결 방법과 협업의 중요성 덕분에 팀원과 함께 성공적으로 완수할 수 있었습니다. 또한, 교수님께서 강조하신 프로세스와 실습이 이론적인 배경을 넘어 실질적인 구현 능력을 키우는 데 큰 역할을 했다고 생각합니다. 오픈소스 SW 활용 수업 또한, 앞으로의 학업과 진로에 있어서도 큰 도움이 될 것 같습니다. 바쁜 일정 속에서도 아낌없는 가르침과 지도를 해주신 교수님께 진심으로 감사드리며, 이번 프로젝트를 통해 한 단계 성장할 수 있는 계기가 되어 뜻깊었습니다. 감사합니다.

## 6.2 32213306 이영주 후기

이번 프로젝트를 통해 애플리케이션 개발의 전 과정을 직접 경험하며, 이론과 실무를 연결할 수 있는 매우 뜻깊은 시간이었습니다. 무엇보다도, 프로젝트를 진행하며 여러 기술적 도전과 협업의 과정을 통해 많은 것을 배우고 성장할 수 있었던 것은 모두 교수님의 세심한 지도와 격려 덕분이라고 생각합니다.

Node.js 와 MongoDB 를 활용하여 데이터를 안정적으로 처리하고 REST API 를 설계하였습니다. 이를 통해 클라이언트-서버 통신 구조를 명확히 이해할 수 있었으며, 현업에서의 개발 프로세스를 실질적으로 체감할 수 있었습니다. 프로젝트 진행 과정에서 발생한 문제들을 해결하며 git 을 이용한 협업의 중요성을 체감할 수 있었습니다. 이전의 프로젝트들과는 달리 이번에는 github 를 적극적으로 활용하며 프로젝트를 진행하였는데, 실제로 github 를 사용하며 여러 가지 문제를 맞닥뜨리고 해결해 나가면서 협업 도구의 활용 방법과 중요성을 깊이 이해할 수 있었습니다. 특히, 브랜치 관리와 충돌 해결 과정을 통해 협업 시 발생할 수 있는 문제들에 보다 능동적으로 대처하는 법을 배웠습니다. 이 과정에서 교수님께서 가르쳐주셨던 git 관련 지식들이 큰 도움이 되었습니다.

마지막으로, 프로젝트를 진행하며 팀원들과의 의사소통과 협업의 중요성을 다시 한번 깊이 깨달았습니다. 역할 분담과 책임감을 바탕으로 서로의 부족한 점을 채워주며 하나의 목표를 향해 나아가는 과정에서, 협력적 태도와 성실함의 의미를 몸소 느낄 수 있었습니다.

이번 프로젝트를 통해 많은 것을 배우고 성장할 수 있었던 것은 교수님의 지도 덕분이라고 생각합니다. 이 경험들을 바탕으로, 앞으로도 끊임없이 성장하며 더욱 노력하는 개발자가 되겠습니다. 감사합니다.