

# 09 PJT

# Vue를 사용한 동영상 검색 관리 서비스 구현

## 챕터의 포인트

- [도전] 동영상 검색 관리 서비스 구현
- 제출

**[도전]**  
**동영상 검색 관리 서비스 구현**

## | 공통 요구사항

- .gitignore 파일을 추가하여 불필요한 파일 및 폴더는 제출하지 않도록 합니다.
- 명시된 요구사항 이외에는 자유롭게 작성해도 무관합니다.
  - 요구사항 완성 후에 Bootstrap 을 이용하여 자유롭게 스타일링합니다.
- **Pinia** 는 선택사항입니다.
- YouTube API
  - Docs: <https://developers.google.com/youtube/v3/getting-started?hl=ko>

## | 세부 요구사항

- A. 동영상 검색결과 출력
- B. 동영상 상세 정보 출력
- C. 나중에 볼 동영상 저장 및 삭제
- D. 내가 좋아하는 채널 저장 및 삭제 (도전과제)
- E. 생성형 AI 활용하기

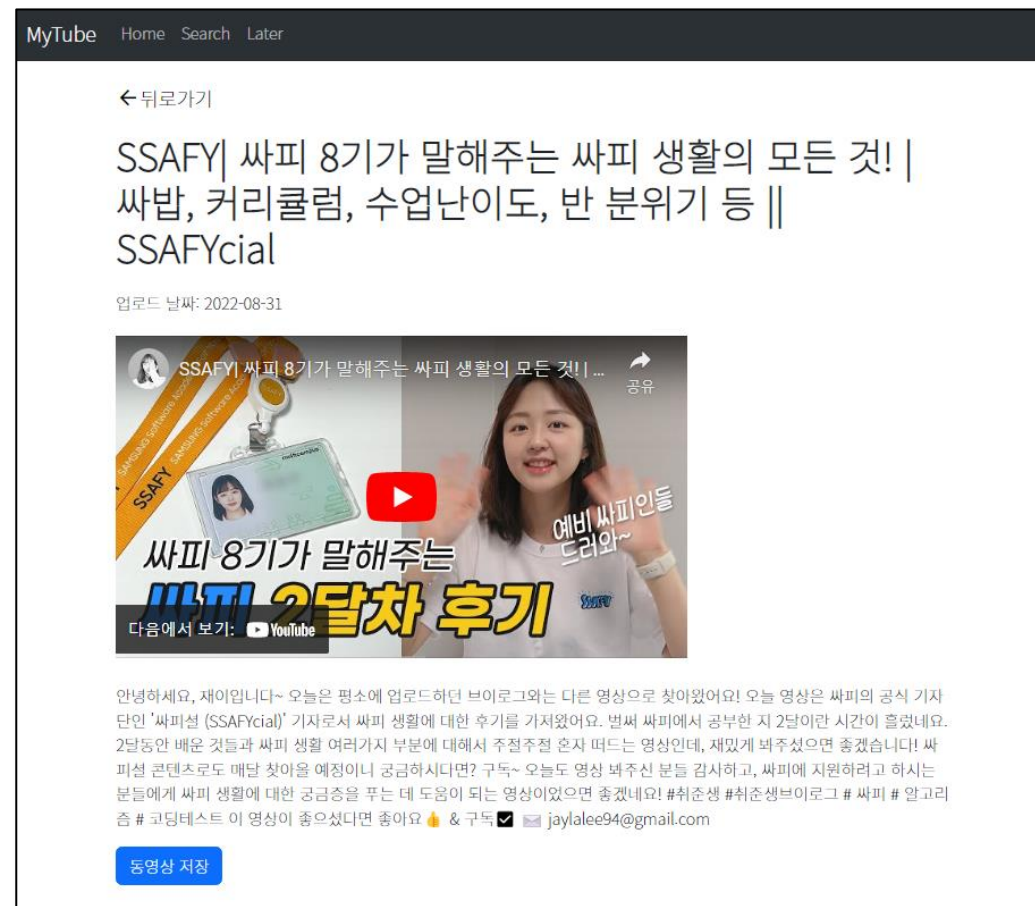
## A. 동영상 검색결과 출력

- 네비게이션 바에서 Search 링크 클릭
- 원하는 검색어 입력
- YouTube API 로부터 JSON 데이터 요청
- 예시와 같이 동영상 목록 출력



## B. 동영상 상세 정보 출력

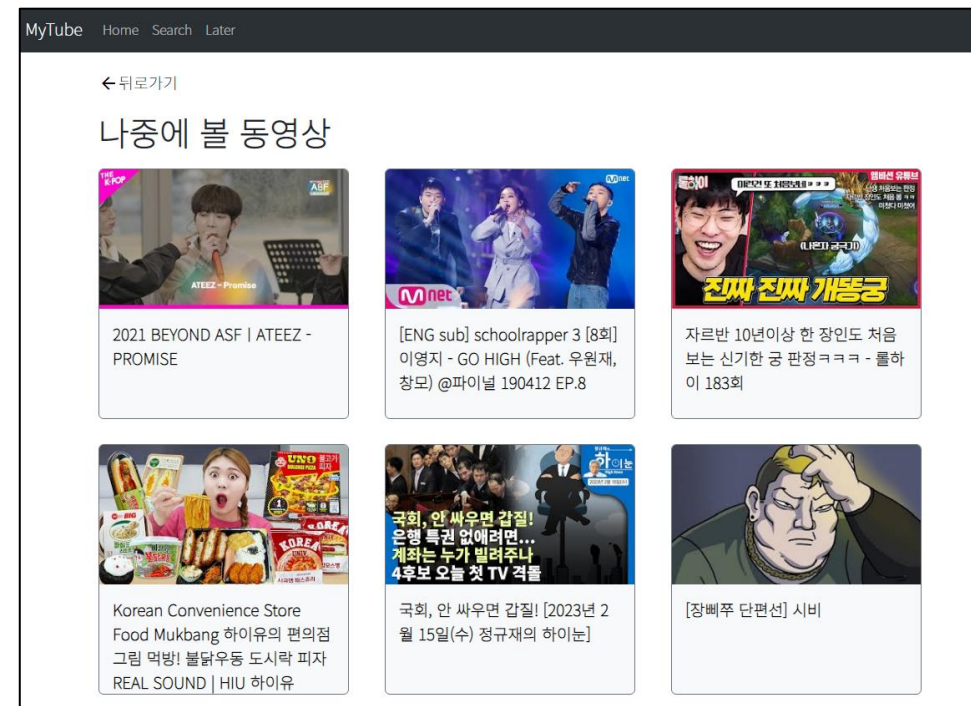
- 검색결과에서 특정 비디오 클릭
- 동영상에 대한 상세 정보 출력
- iframe 태그 활용해 동영상 재생
- 동영상 저장
  - Local Storage 활용
  - 저장 안 된 동영상 => 동영상 저장 버튼
  - 저장된 동영상 => 저장 취소 버튼





## C. 나중에 볼 동영상 저장 및 삭제

- 저장된 동영상 목록 확인
  - Local Storage 활용
  - 등록된 동영상 없을 경우 - 등록된 비디오 없음 출력



## | D. 내가 좋아하는 채널 저장 및 삭제 (도전과제)

- 상세 페이지에 채널 저장 기능 추가
- 클릭 시 Local Storage 활용해 채널 저장
- 채널 페이지 접속 시 저장된 채널 목록 출력

해서 주절주절 혼자 떠드는 영상인데, 재  
신 분들 감사하고, 싸피에 지원하려고 하  
로그 # 싸피 # 알고리즘 # 코딩테스트 0

동영상 저장

채널 저장

MyTube Home Search Later Channel

## | E. 생성형 AI 활용하기

- 생성형 AI 를 활용하여 적절한 아이디어를 도출합니다.
- 설계 단계
  - 효율적인 컴포넌트 구조 및 데이터 저장 구조 등에 대한 조언 및 아이디어 도출
- 구현 단계
  - 코드 리팩토링(ex. 라이프 사이클 사용 여부 등)을 통한 코드 성능 개선
- 각 프롬프트를 자유롭게 구성한 후 분석 과정 및 결과를 README.md 에 작성합니다.
  - 분석 과정은 대화 화면, 검색 결과 등 자유롭게 구성합니다.

# 제출

## | 제출 시 주의사항

- 제출기한은 금일 18시까지입니다. 제출기한을 지켜 주시기 바랍니다.
- 반드시 README.md 파일에 단계별로 구현 과정 중 학습한 내용, 어려웠던 부분, 새로 배운 것들 및 느낀 점 등을 상세히 기록하여 제출합니다.
  - 단순히 완성된 코드만을 나열하지 않습니다.
- 위에 명시된 요구사항은 최소 조건이며, 추가 개발을 자유롭게 진행할 수 있습니다.
- <https://lab.ssafy.com/> 에 프로젝트를 생성하고 제출합니다.
  - 프로젝트 이름은 '프로젝트 번호 + pjt' 로 지정합니다. (ex. 09\_pjt)
- 반드시 각 반 담당 교수님을 Maintainer 로 설정해야 합니다.