

# ESTube

 영상 추천 YOUTUBE CLONE PAGE

## Query Union

---

팀장: 조권호

권정연

유동혁

# 목 차

---

## 01 사전조사

## 02 프로젝트 요약 ————— 목표 및 배경 / 필수 구현 기능 / 추가 구현 기능

## 03 주요 개발 내용 ————— 조원별 주요 개발 기능 / 프로젝트 개발 기능 시연

## 04 개발 소감 ————— 인상 깊었던 기능 / 팀 커뮤니케이션 / 개발 중 장애물&극복 방법 / 추가·리서치 하고싶은 기능

## 05 우리 조의 MVP

# 사전 조사

---

Notion에 공유된 '유튜브 클론 코딩 샘플 프로젝트' Figma 디자인 시안을 참고하여 전반적인 UI 구조와 컴포넌트 배치 방식을 분석하고, 실제 YouTube 웹사이트를 벤치마킹해 페이지 레이아웃 및 주요 기능 동작 방식을 학습해 적용할 인사이트를 도출했습니다.



Figma



Youtube

# 프로젝트 요약

목표 및 배경 / 필수 구현 기능 / 추가 구현 기능

# 목표 및 배경

학습한 HTML·CSS·JavaScript 기술을 실전에 적용해 YouTube 클론 페이지를 구현함으로써 웹 프론트엔드 개발 역량을 실질적으로 향상시키고, 프로젝트 전반에 걸친 팀 협업 과정을 통해 효과적인 소통 및 문제 해결 능력을 강화하기 위해 진행했습니다.



**HTML**



**CSS**



**JS**

프로젝트 커리큘럼에 따라 진행하여 완성한 기능들을 소개합니다. 각 기능은 서비스의 기본 구조와 주요 동작 흐름을 구성하며, HTML/CSS 기반 UI부터 JavaScript 상호작용까지 핵심 요구사항을 반영합니다.

## 레이아웃 및 UI 구현

(HTML/CSS)

- 상단바·사이드바(3페이지 공통) 완성
- 메인페이지 카드·버튼 UI 완성
- 채널페이지 기본 구조 완성
- 비디오페이지 기본 구조 완성

## 인터랙티브 기능 구현

(JavaScript)

- 검색 기능 완성
- 사이드바(3페이지 공통) 동적 처리
- 구독 기능 구현
- 비디오 플레이어 구현·완성
- 댓글 기능 구현·완성

## AI 추천 목록 생성

(JavaScript)

- 현재 재생영상 관련한 추천 재생목록 생성

필수 기능을 넘어 UX 개선과 탐색 편의성 강화를 위해 자율적으로 설계·구현한 부가 기능들을 소개하며, 각 기능은 학습을 통해 익힌 기술을 최대한 적용하기 위해 노력하였습니다.

기능 이름	설명
반응형 미디어 쿼리	모바일·태블릿 대응 CSS 적용
카드 UI 애니메이션	마우스 포인터가 위치한 부분이 떠오르는 애니메이션
호버 자동 재생	썸네일에 마우스 올리면 영상 자동 재생
댓글 좋아요/싫어요	댓글에 좋아요·싫어요 버튼 구현
필터 기능	카테고리·정렬 필터링 UI 적용
정렬 기능	좋아요 개수 및 업로드 날짜, 조회수로 영상 정렬
미니 사이드바	상단바의 메뉴 버튼 클릭 시 기존 사이드바가 축소된 미니 사이드바로 전환
테마 변경	다크/라이트 모드 토글 구현
비디오 좋아요/싫어요 & 목록 페이지	좋아요 누른 영상 모아보기 페이지
구독 채널 영상 목록 페이지	구독 채널의 모든 영상 모아보기 페이지



# 주요 개발 내용

조원별 주요 개발 기능 / 프로젝트 개발 기능 시연

# 조원별 주요 개발 기능

주요 개발 내용

각 조원이 프로젝트에서 맡아 구현한 주요 기능을 한눈에 확인할 수 있도록 정리했습니다.

## HTML

- 상단바
- 메인페이지 일부(home.html, index.html )

## CSS

- 채널 페이지
- 메인 페이지 일부(content.css, filter.css)



조권호

## JavaScript

- 메인 페이지
- 탭바 및 사이드바 모듈화
- 현재 재생 영상 관련 추천 재생 목록 알고리즘

## 고도화 작업

- 메인 페이지 태그 필터 기능
- 메인 페이지 정렬 필터 기능

# 조원별 주요 개발 기능

주요 개발 내용

각 조원이 프로젝트에서 맡아 구현한 주요 기능을 한눈에 확인할 수 있도록 정리했습니다.

## HTML

- 메인 페이지 카드UI
- 채널 페이지

## CSS

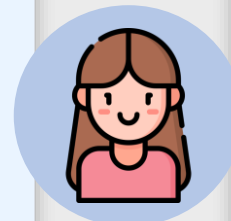
- 상단바
- 비디오 페이지
- 반응형 미디어 쿼리

## JavaScript

- 채널 페이지

## 고도화 작업

- 테마 변경 기능(다크모드/라이트모드)
- 비디오 페이지 댓글 기능
- 댓글 좋아요/싫어요 /수정/삭제 기능
- 채널 페이지 대표 비디오 배열 및 설명문 CSS 수정
- 채널 페이지 비디오 카드 CSS 수정
- 메인 페이지 썸네일 이미지 및 카드 배열 CSS 수정



권정연

# 조원별 주요 개발 기능

주요 개발 내용

각 조원이 프로젝트에서 맡아 구현한 주요 기능을 한눈에 확인할 수 있도록 정리했습니다.

## HTML

- 사이드바
- 비디오 페이지

## CSS

- 사이드바
- 메인페이지 카드UI

## JavaScript

- API호출 메서드
- UI호출 및 로딩
- 탭바 및 사이드바 버튼 기능
- 구독 기능
- 검색 기능
- 공통 유틸리티 메서드
- 비디오 페이지



유동혁

## 고도화 작업

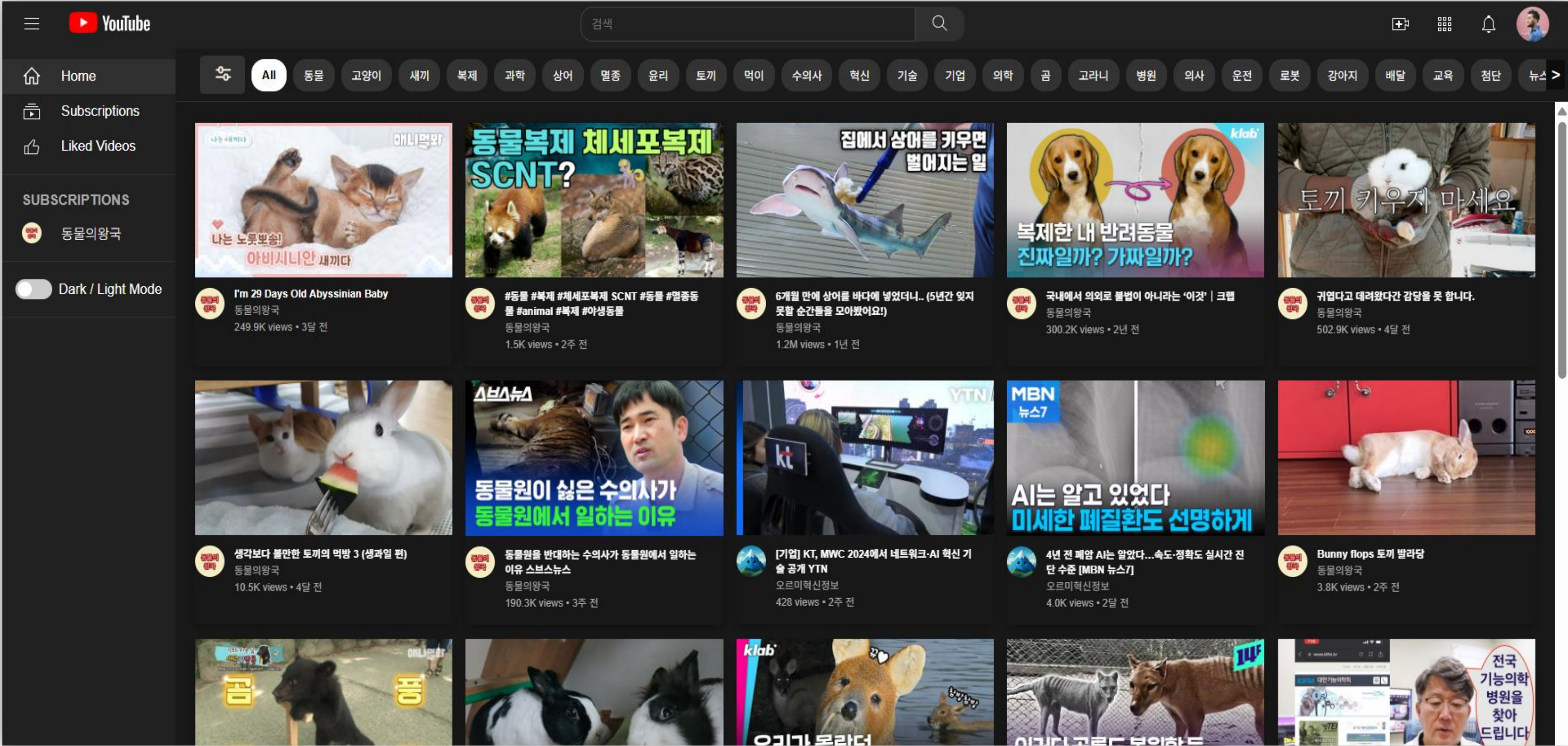
- 비디오 플레이어 추가
- 비디오 페이지 좋아요/싫어요 기능 및 애니메이션 기능
- 비디오 페이지 태그 검색 버튼 기능
- 카드 hover 스타일 수정
- 반응형 미디어 쿼리 수정



# 프로젝트 개발 기능 시연

주요 개발 내용

해당 시연을 통해 검색, 영상 재생, 추천 목록, 댓글 작성 및 구독 기능이 실제 UI 상에서 어떻게 작동하는지 확인할 수 있습니다



# 개발 소감

인상 깊었던 기능 / 팀 커뮤니케이션 / 개발 중 장애물&극복 방법 / 추가·리서치 하고싶은 기능



# 인상깊었던 기능

개발 소감

프로젝트 기간 동안 구현한 기능 중 개인적으로 가장 인상 깊었던 기능과 그 이유를 함께 공유합니다.



조권호

인상 깊었던 기능:

현재 재생 영상 관련 추천 재생 목록 알고리즘

추천 재생목록을 구현하면서 429 http status code (연속호출 허용 범위 초과) 오류코드 등을 해결하고 어떻게 하면 API요청을 최소화할 수 있을까 고민한 것이 인상깊었다.



권정연

인상 깊었던 기능:

테마 변경 토글 기능

버튼 하나로 CSS 변수를 통째로 바꿔 전체 페이지 색상을 전환하는 구조를 구현하면서, CSS와 JavaScript만으로도 스타일을 손쉽게 제어할 수 있다는 점이 신기했다. 특히 변수를 지정하는 것만으로 페이지 분위기를 간편하게 바꿀 수 있다는 점에서 코드의 효율성과 편리함을 직접 느꼈다.



유동혁

인상 깊었던 기능:

버튼 클릭 애니메이션, 커스텀 비디오 플레이어

둘 다 라이브러리를 사용하면 좀 더 편하게 만들 수 있다고 한다. 라이브러리 없이 바닐라 js 만으로 최대한 구현 하려다보니, 어려움도 많았지만 해결하는 재미를 느낄 수 있었다.

# 팀 커뮤니케이션

디스코드를 중심으로 진행한 업무 분담, 이슈 공유 및 피드백 등의 팀 커뮤니케이션 과정을 소개합니다.



Discord

업무 분담

회의·협의  
방식

문제 발생 시  
대응

성과공유 &  
피드백



개발 과정 중 마주한 문제점과, 각 문제를 해결하기 위해 적용한 구체적인 방안을 간결하게 정리했습니다.

## 문제 및 원인(1)

HTML에 data-video-id 속성을 주고 JavaScript에서 하이픈을 포함해 접근했더니 'undefined' 반환 됨

예시) <JS> element.dataset["video=id"]

'dataset'은 하이픈 속성을 자동으로 카멜케이스(videoId)로 바꿔 처리하기 때문에 원본 문자열(video-id)로 읽히지 않음

## 해결 방안(1)

다음과 같은 표기법으로 항상 카멜케이스로 접근하도록 통일하였음

1. 점 표기법 : element.dataset.videoId
2. 대괄호 표기법: element.dataset["videoId"]

```
const element1 = document.querySelector("[data-video-id]");
const videoId1 = element1.dataset["videoId"];
```

## 문제 및 원인(2)

검색 입력 필드에서 한글만 입력되고, 영문·숫자·특수 문자는 아예 입력되지 않는 현상 발생

'keypress' 이벤트 리스너에서 'event.preventDefault()'를 모든 키에 적용해 버려 일반 키 입력 자체가 차단되었기 때문

## 해결 방안(2)

'keypress' 핸들러를 다음과 같이 변경해 'Enter' 키일 때만 기본 동작을 막고 검색 로직을 실행하도록 분기 처리

```
searchInput.addEventListener("keypress", (event) => {
  if (event.key === "Enter") {
    event.preventDefault();
    triggerSearch();
  }
});
```

개발 과정 중 마주한 문제점과, 각 문제를 해결하기 위해 적용한 구체적인 방안을 간결하게 정리했습니다.

## 문제 및 원인(3)

테마 변경 토글 기능을 적용한 스위치를 작동해도 theme-toggle.js가 적용되지 않음

1. 스크립트 로딩 순서 문제: JS가 사이드바 로드 전에 실행되어 스위치 미탐지
2. 동적 로딩으로 인한 이벤트 미적용: 사이드바가 동적으로 로드될 때 JS의 이벤트 리스너가 이미 실행되어 나중에 추가되는 토글 스위치에 이벤트가 바인딩 되지 않음
3. 초기 테마 상태 동기화 문제: HTML문서에 'data-theme'속성 미설정 → 로컬 스토리지 값과 UI 불일치

## 해결 방안(3)

1. index.html에 <html lang="en" data-theme="dark"> 추가
2. <script type="module" src="/scripts/theme-toggle.js">를 끝에 배치
3. theme-toggle.js에 'DOMContentLoaded'와 'MutationObserver'로 사이드바 로드 시점 감지 및 토글 초기화를 동적으로 실행
4. loadUI.js에 사이드바 로드 완료 후 'CustomEvent('sidebarLoaded')' 발행  
→ 토글 초기화 트리거

## 3번 해결방안

```
// MutationObserver를 사용하여 사이드바 로드 감지
const observer = new MutationObserver((mutations) => {
  mutations.forEach((mutation) => {
    if (mutation.addedNodes.length) {
      const toggle = document.getElementById("theme-toggle-switch");
      if (toggle) {
        initThemeToggle();
        observer.disconnect();
      }
    }
  });
});

// 사이드바 컨테이너 감시 시작
const sidebarContainer = document.getElementById("side-bar-container");
if (sidebarContainer) {
  observer.observe(sidebarContainer, { childList: true, subtree: true });
}

// 페이지 로드 시 테마 초기화
document.addEventListener("DOMContentLoaded", () => {
  const htmlEl = document.documentElement;
  const toggle = document.getElementById("theme-toggle-switch");
```

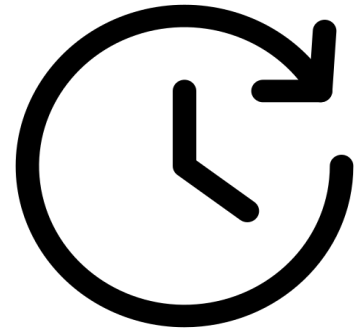
## 4번 해결방안

```
// 사이드바 로드 완료 후 테마 토글 초기화를 위한 이벤트 발생
const event = new CustomEvent("sidebarLoaded");
document.dispatchEvent(event);
```

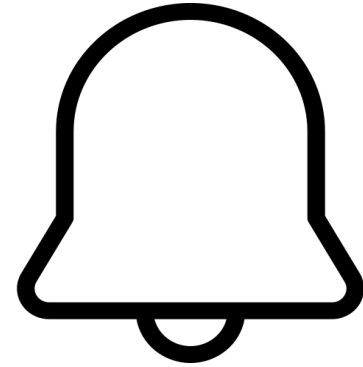
# 추가 / 리서치 하고 싶은 기능

개발소감

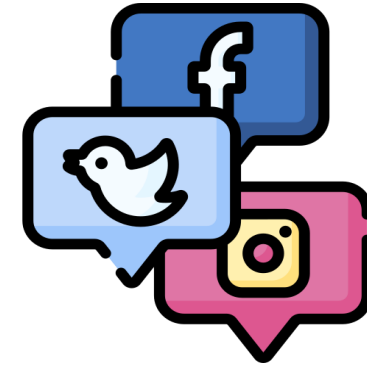
서비스 완성도와 사용자 만족도를 높이기 위해 향후 검토·연구하고자 하는 기능 아이디어를 모아봤습니다.



시청 기록



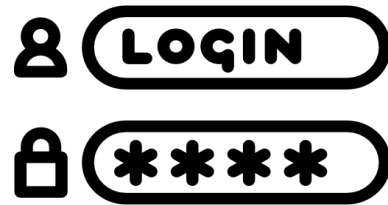
알림 기능



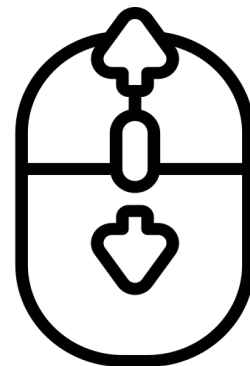
소셜 공유



더보기



로그인



무한 스크롤



이어보기

# 우리 조의 MVP

---

# 우리 조의 MVP는?

MVP 조원

GitHub 관리부터 프로젝트 진행 상황 작성 등 사소하지만 반드시 필요한 업무를 책임감 있게 전담하며, 프로젝트 일정 조율과 이슈 관리를 적극적으로 수행해 팀 전체가 원활히 작업할 수 있도록 지원하여 저희 팀의 MVP로 선정하였습니다.



# 조권호

# 감사합니다

Query Union

---

팀장: 조권호

권정연

유동혁