

PLAYLIST

플레이리스트 포트폴리오

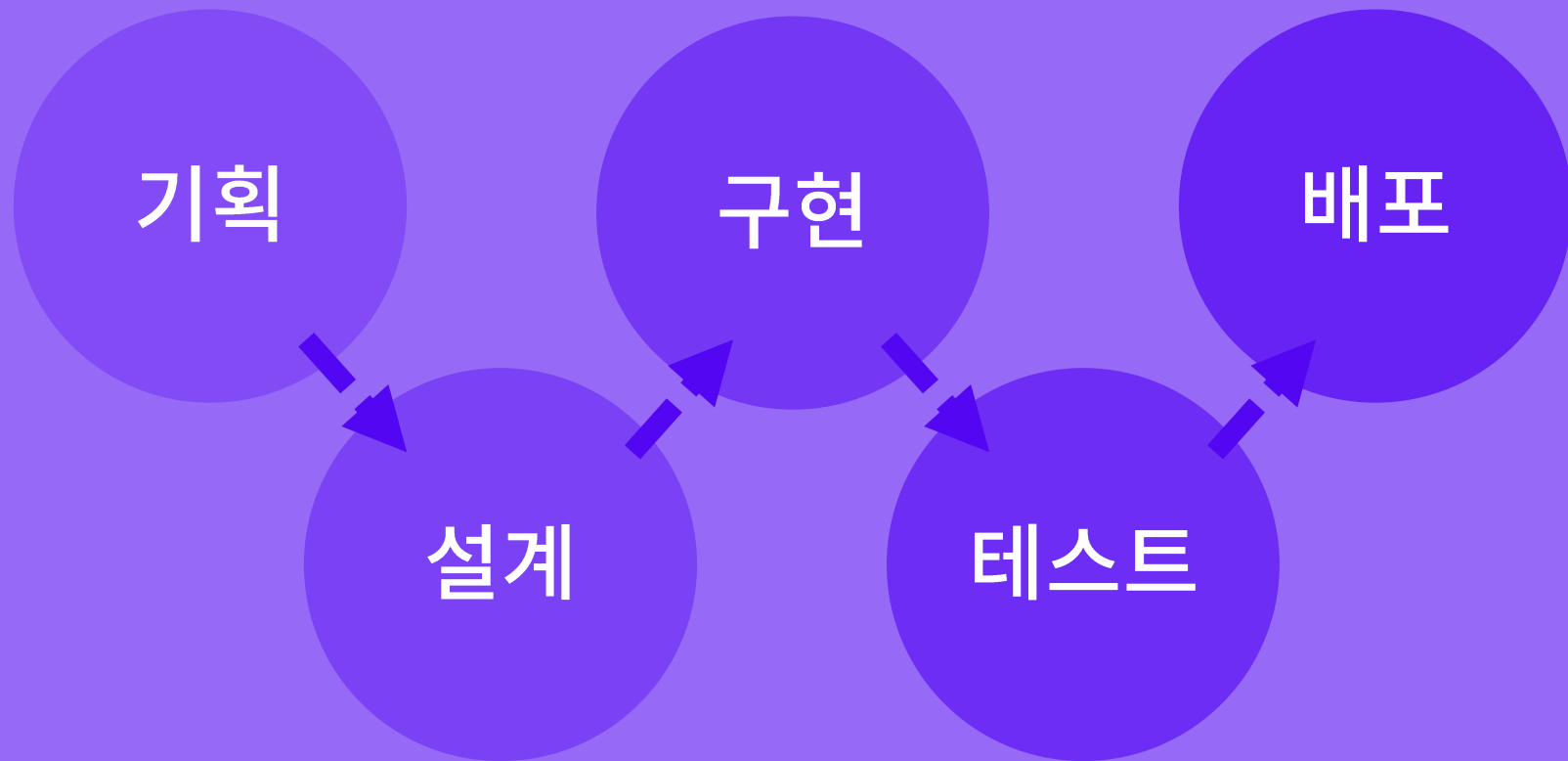
07.20 ~ 08.05

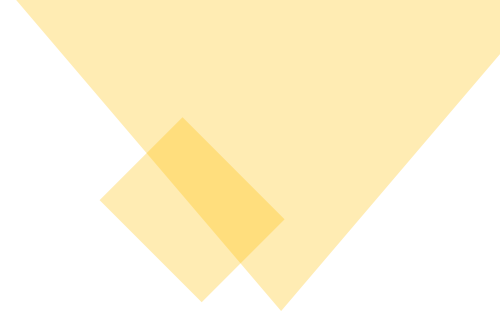
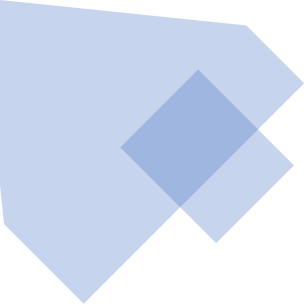
김예승

dev_webius@kakao.com

010-2164-7336

PROGRESS





PLAYLIST 기획



PURPOSE

떠오르는 동영상 시장,

발전하는 수많은 플랫폼

통합 플랫폼은 없을까?

MAIN POINT

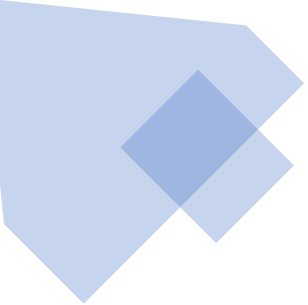
- ① 다양한 영상을 통합된 플랫폼에서 제공
- ② 접속 기기에 상관 없이 지원되는 반응형
- ③ 부트스트랩으로 디자인 개발 기간 단축
- ④ 꼼꼼한 오류 대응 설계

MEMBER



SCHEDULE

일정	7월												8월				
	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5
서버 환경 및 개발 툴 선정																	
대상 플랫폼 탐색																	
기본 개발 환경 구축																	
웹 서비스 개발 및 연동																	
서비스 가용성 테스트 및 오류 수정																	
프로그램 최종 수정 후 배포																	



PLAYLIST 설계



ENVIRONMENT

클라이언트

HTML5
CSS3
JavaScript
jQuery
Bootstrap
FontAwesome

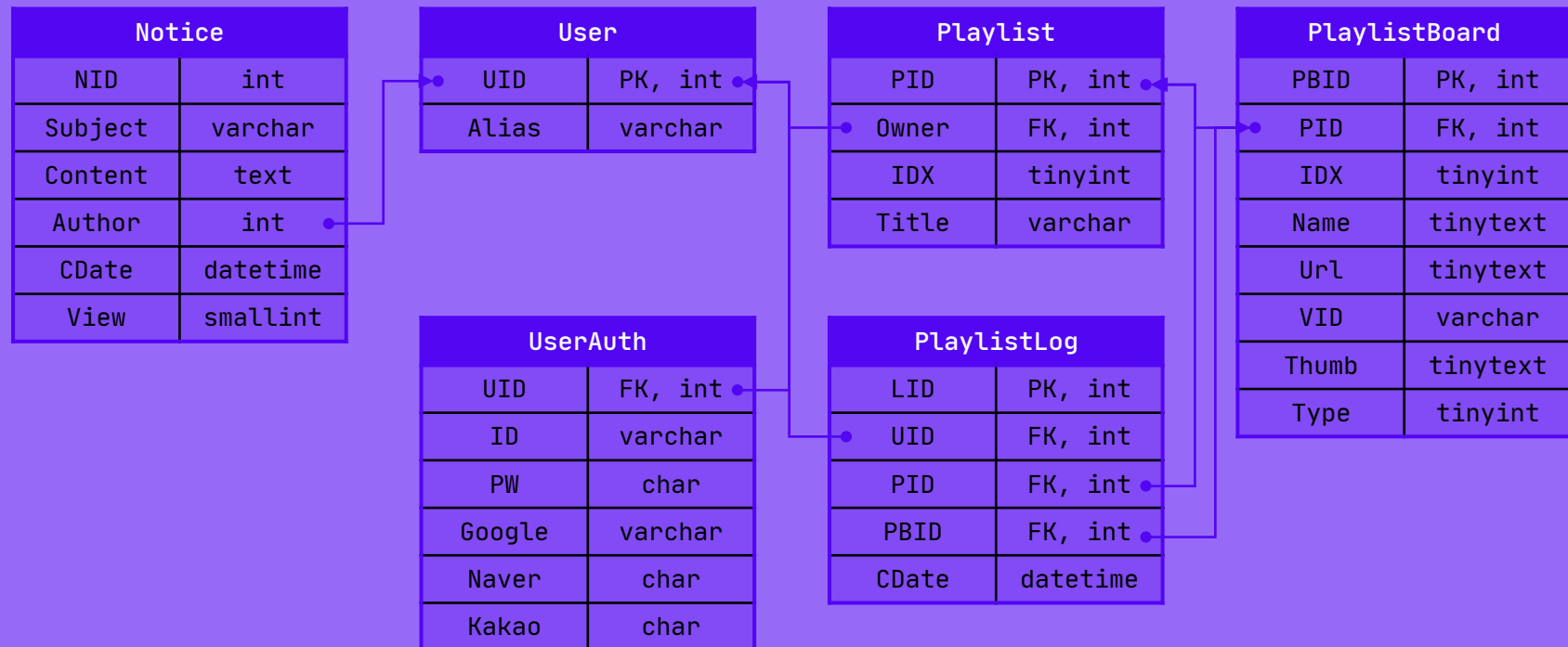
서비스

Nginx
Apache2
Tomcat9
MariaDB

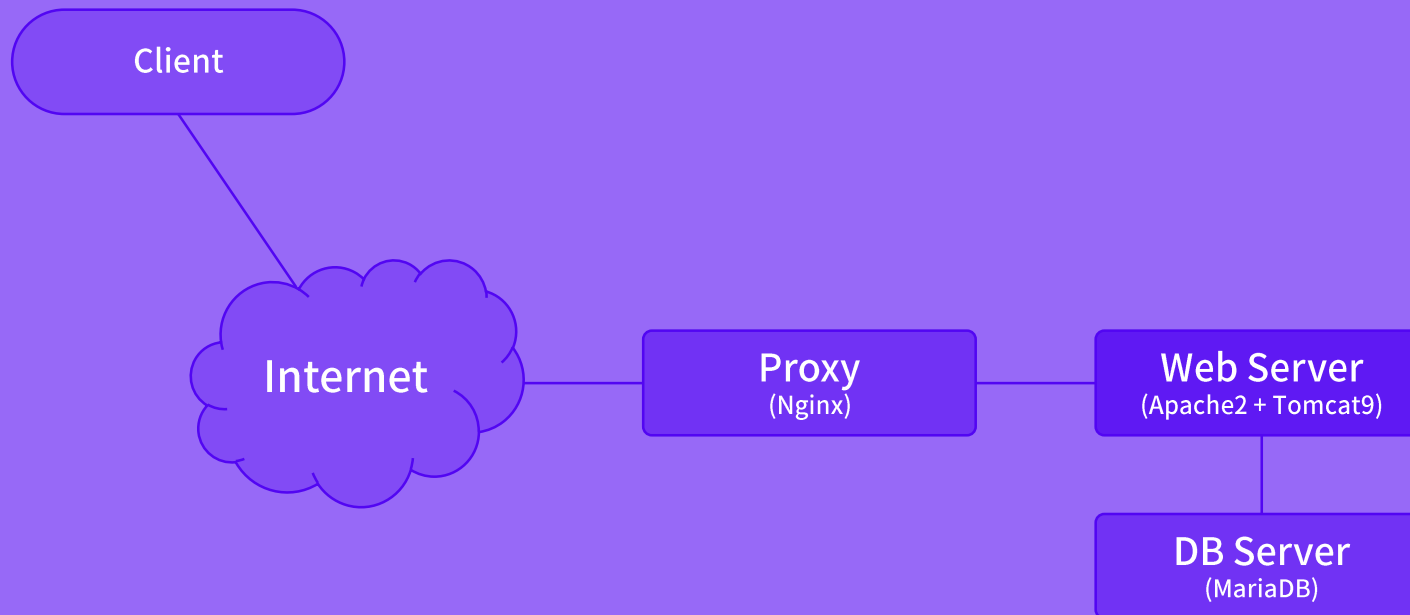
서버

Spring Framework
JSP
MyBatis
Lombok
Slf4j
Jsoup
Jackson JSON Parser

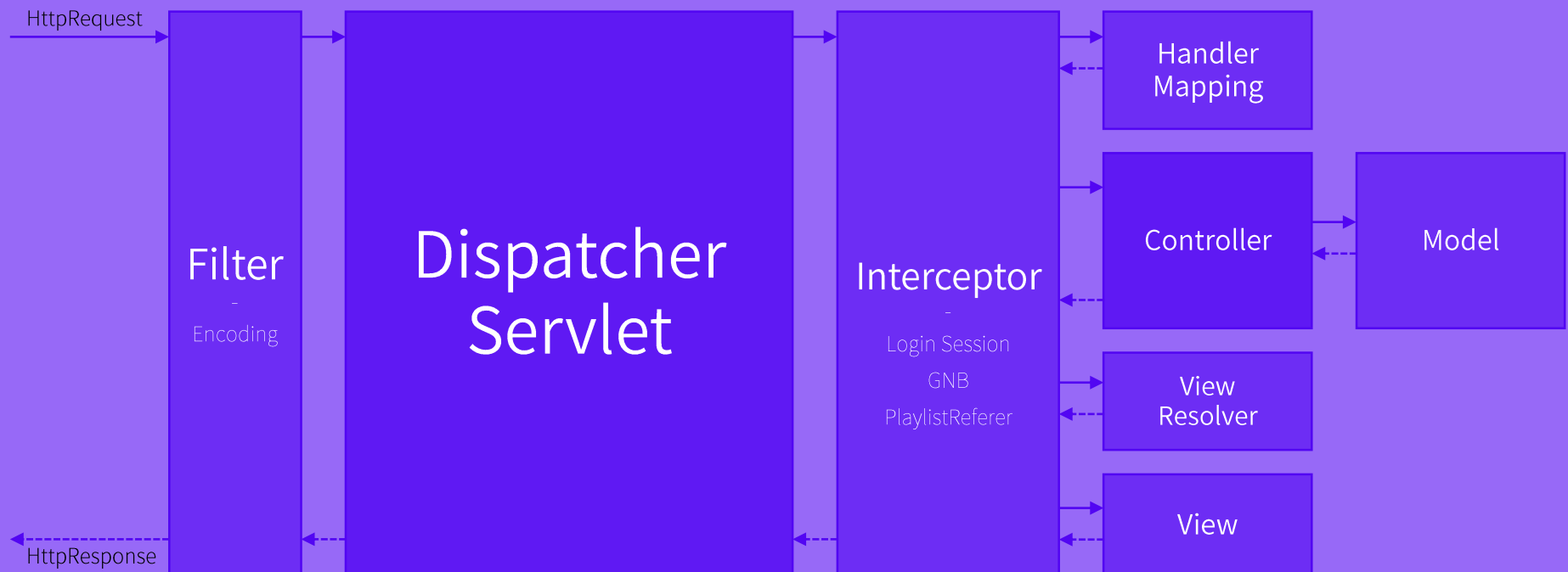
DB MODELING

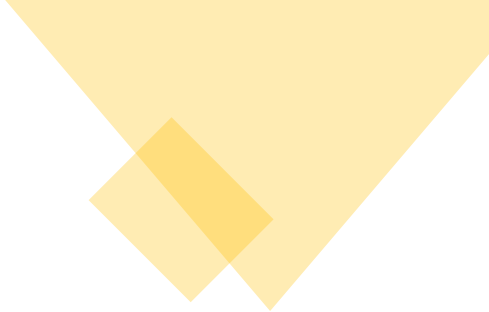
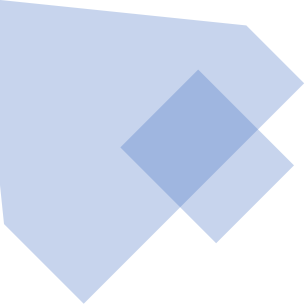


SERVICE NETWORK



SPRING MVC ARCHITECTURE

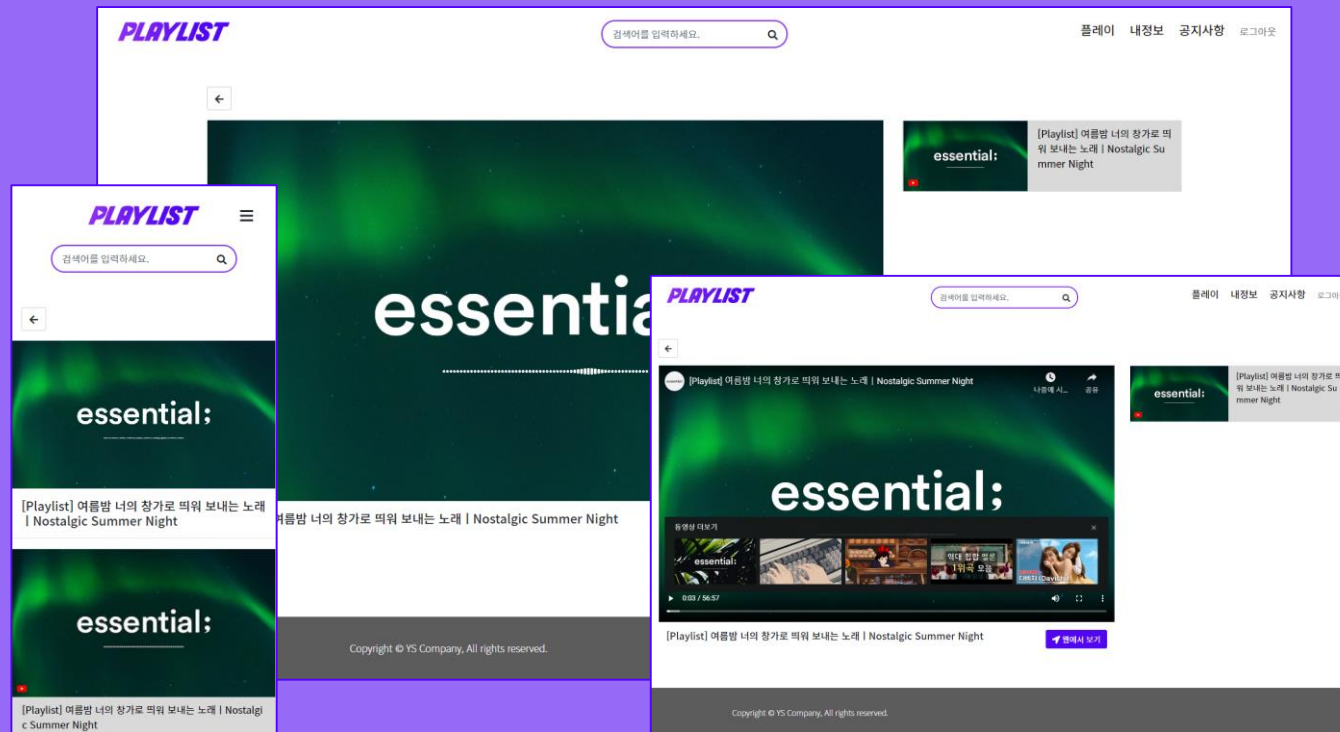




PLAYLIST 구현



#1. Responsive



Bootstrap을 이용하여 반응형 웹 페이지 작성

#1. Responsive

문제점

맥북, 태블릿 기기에서 일부 영역의 크기가 너무 작아
화면이 정상적으로 표시되지 않던 문제

해결 방안

서버에서 User-Agent 헤더 값을 가공하여 View로 전송하고
JSTL Core를 이용하여 맥북, 태블릿 기기에서 다른 크기로 표시하도록 작성

#2. SNS Login API

The image displays two side-by-side mobile app login screens for an application named 'PLAYLIST'. Both screens feature a white background with a blue header bar containing the 'PLAYLIST' logo and a hamburger menu icon. Below the header is a search bar with the placeholder text '검색어를 입력하세요.' and a magnifying glass icon. The main heading 'PLAYLIST' is followed by the text '당신의 실행 목록을 만드세요!'. The left screen shows a standard login form with fields for '아이디' (ID) and '암호' (Password), followed by a blue '로그인' (Login) button, and three social media login buttons: yellow for '카카오 로그인' (Kakao Login), green for '네이버 로그인' (Naver Login), and red for '구글 로그인' (Google Login). A white '회원가입' (Sign Up) button is at the bottom. The right screen shows a simplified login form with a single 'SNS 닉네임' (SNS Nickname) field and a blue '가입하기' (Join) button. Both screens have a dark grey footer with the text 'Copyright © YS Company, All rights reserved.'

기본 제공되는 사용자 ID&PW 인증 외 SNS 로그인 API 적용

#2. SNS Login API

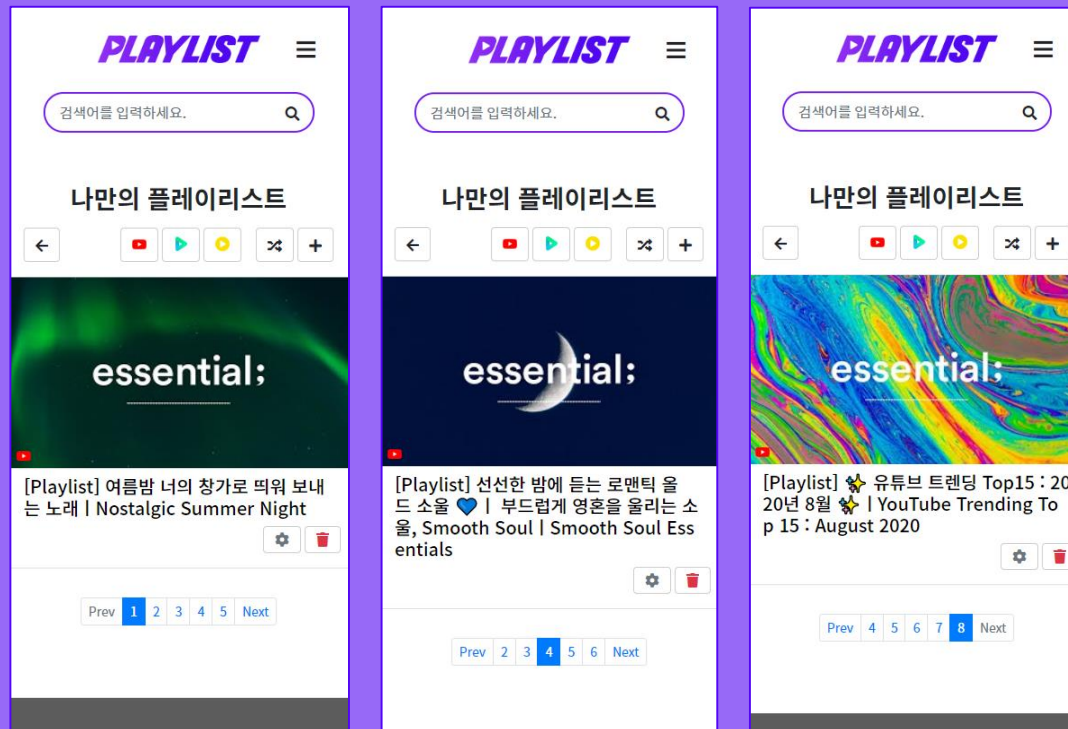
아쉬운 점

최초 가입 후 다른 인증 방식으로 연동할 수 없음

개선 방향

내 정보 페이지에서 가입 시 사용하지 않았던 다른 인증 방식에 대해
추가 연동을 진행할 수 있도록 개선 예정

#3. Pagination



페이지 길이에 따라 유동적으로 동작하는 페이지징

#4. Embed Player

Player의 핵심

- 1) 16 x 9 크기에 적절하게 동작
- 2) Stop 이벤트 핸들링 가능

#4. Embed Player

iframe 삽입에 대한 YouTube Player API 참조 문서

IFrame Player API를 통해 웹사이트에 YouTube 동영상 플레이어의 펌과 JavaScript를 사용하여 플레이어를 제어할 수 있습니다. 웹페이지에 Flash 개체를 삽입하는 것과 관련된 [Flash](#) 및 [JavaScript](#) Player API와 달리 IFrame API는 콘텐츠를 페이지의 `<iframe>` 태그에 게시합니다. 이 방법은 YouTube에서 Flash를 지원하지 않는 모바일 기기에 Flash 대신 HTML5 플레이어를 게재할 수 있도록 하므로 기존에 제공된 API보다 좀 더 유연성을 제공합니다.

API의 JavaScript 함수를 사용하여 재생을 위해 동영상을 대기열에 넣거나, 이러한 동영상을 재생, 일시중지 또는 중지하거나, 플레이어 볼륨을 조정하거나, 재생 중인 동영상에 대한 정보를 가져올 수 있습니다. 또한 플레이어 상태 변경 또는 동영상 재생 품질 변경과 같은 특정 플레이어 이벤트에 대한 응답으로 실행되는 이벤트 리스너를 추가할 수 있습니다.

이 가이드에서는 IFrame API를 사용하는 방법을 설명합니다. API에서 보낼 수 있는 이벤트의 여러 유형을 식별하고 이벤트 리스너를 작성하여 이러한 이벤트에 응답하는 방법을 설명합니다. 또한 동영상 플레이어를 제어하기 위해 호출할 수 있는 여러 JavaScript 함수 및 플레이어를 좀 더 맞춤설정하는 데 사용할 수 있는 플레이어 매개변수를 자세히 설명합니다.



Youtube 플랫폼의 경우 Embed API를 공식적으로 지원함

#4. Embed Player

- ① NaverTV와 KakaoTV 플랫폼의 경우
Embed Player API를 따로 제공하지 않음
- ② IFrame 사용 시 Cross Domain 정책에 의해 제어할 수 없음
따라서 Stop 이벤트 핸들링이 불가능함

#4. Embed Player

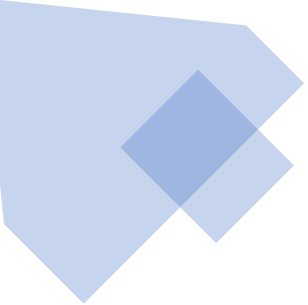
```
<bean id="platformUtil" class="net.webius.playlist.util.PlatformUtil">
  <property name="YOUTUBE_PATTERN">
    <list>
      <value>^(?:http(?:s)?://)?www.youtube.com/watch?v=([a-zA-Z0-9_-]+)</value>
      <value>^(?:http(?:s)?://)?music.youtube.com/watch?v=([a-zA-Z0-9_-]+)</value>
      <value>^(?:http(?:s)?://)?www.youtube.com/watch/([a-zA-Z0-9_-]+)</value>
      <value>^(?:http(?:s)?://)?youtu.be/([a-zA-Z0-9_-]+)</value>
      <value>^(?:http(?:s)?://)?www.youtube.com/embed/([a-zA-Z0-9_-]+)</value>
    </list>
  </property>
  <property name="NAVER_PATTERN">
    <list>
      <value>^(?:http(?:s)?://)?tv.naver.com/v/([0-9]+)</value>
      <value>^(?:http(?:s)?://)?tv.naver.com/embed/([0-9]+)</value>
    </list>
  </property>
  <property name="KAKAO_PATTERN">
    <list>
      <value>^(?:http(?:s)?://)?tv.kakao.com/v/([0-9]+)</value>
      <value>^(?:http(?:s)?://)?tv.kakao.com/channel/(?:[0-9]+)/cliplink/([0-9]+)</value>
      <value>^(?:http(?:s)?://)?tv.kakao.com/embed/player/cliplink/([0-9]+)</value>
      <value>^(?:http(?:s)?://)?play-tv.kakao.com/embed/player/cliplink/([0-9]+)</value>
    </list>
  </property>
</bean>
```

URL 입력 시 정규 표현식을 통해 플랫폼 인식

#4. Embed Player

```
<div>
  <div class="embed-responsive embed-responsive-16by9">
    <div class="embed-responsive-item">
      <c:choose>
        <c:when test="${play.platform == 'Youtube'}">
          <div id="player"></div>
          <%@include file="/WEB-INF/views/static/modules/api_youtube.jsp"%>
        </c:when>
        <c:when test="${play.platform == 'Naver'}">
          <div id="player"></div>
          <%@include file="/WEB-INF/views/static/modules/api_naver.jsp"%>
        </c:when>
        <c:when test="${play.platform == 'Kakao'}">
          <video id="player" autoplay controls...>
          <%@include file="/WEB-INF/views/static/modules/api_kakao.jsp"%>
        </c:when>
        <c:otherwise>
          <p>해당 플랫폼은 지원되지 않습니다.</p>
        </c:otherwise>
      </c:choose>
    </div>
  </div>
</div>
```

각 플랫폼 별로 플레이어가 연결되도록 구현

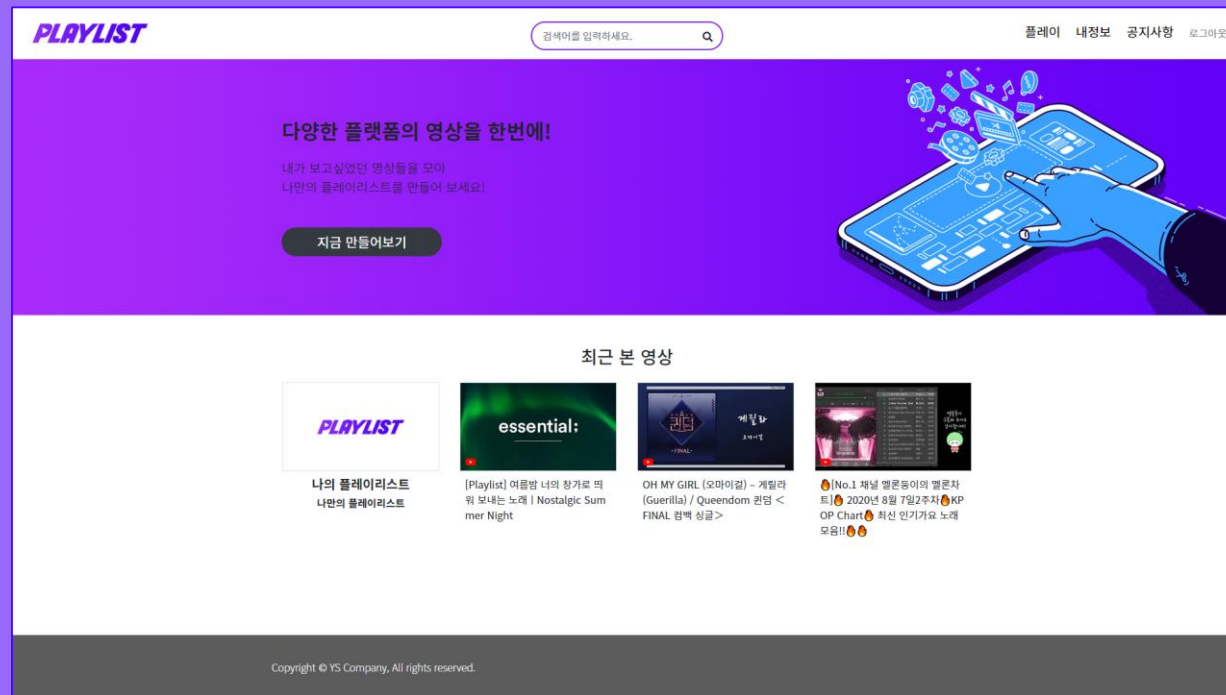


PLAYLIST

테스트



#1. Main



로그인 후 메인 페이지에서 최근 본 영상 제공
(최근 플레이리스트 1개, 영상 3개 수집)

#1. Main

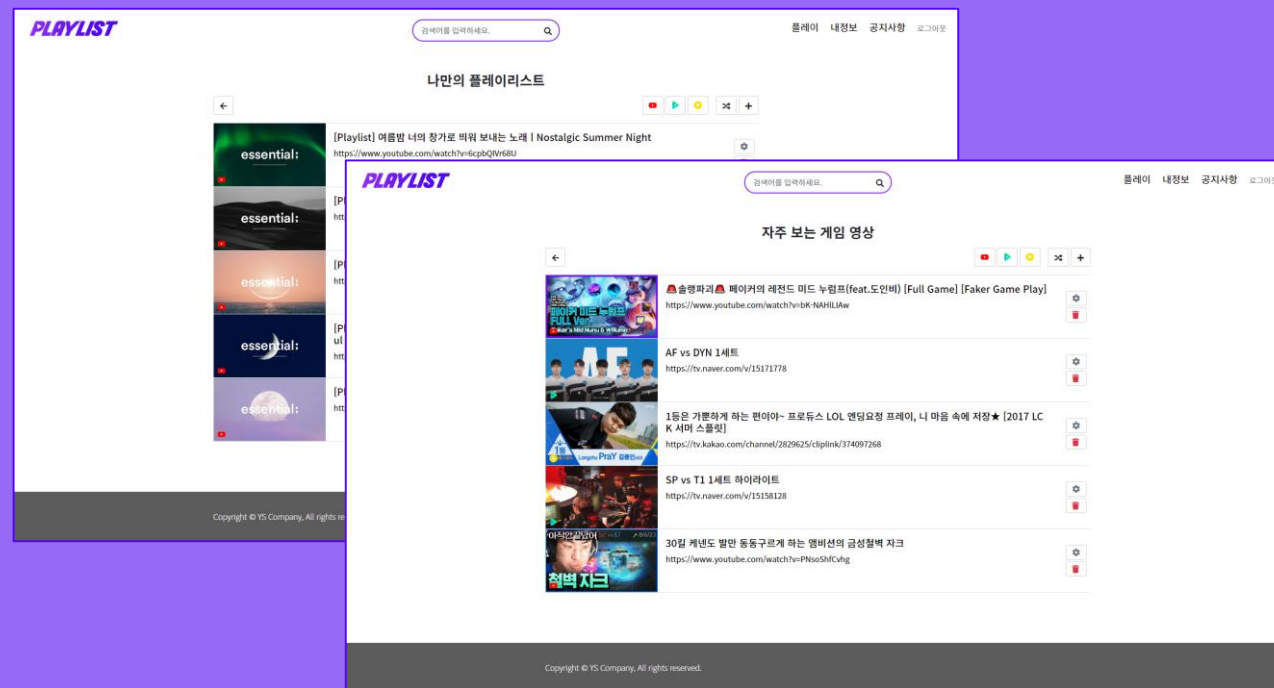
아쉬운 점

1. 플레이리스트를 대상으로 썸네일 정보가 없음
2. 추천 영상 정보가 없음

개선 방향

1. 플레이리스트의 가장 최상단 영상 썸네일을 수집하도록 개선 예정
2. 영상 해시 태그 기능을 추가하고 사용자 별 자주 시청하는 기록을 분석하여 추천 영상을 제공하도록 개선 예정

#2. Playlist List Page



영상 추가부터 셔플, 각 플랫폼 영상 재생 등 제공

#2. Playlist List Page

문제점

영상 URL 등록 시 요청 후 응답까지의 대기 시간동안
다시 등록 버튼을 누를 경우 같은 IDX 값으로 두 번 등록되는 문제 발생

해결 방안

홈페이지 내 모든 요청은 Submit 시 검증 시스템이 데이터를 검증되는데,
데이터 검증이 끝날 때 등록 버튼이 비활성화 되도록 수정

#2. Playlist List Page

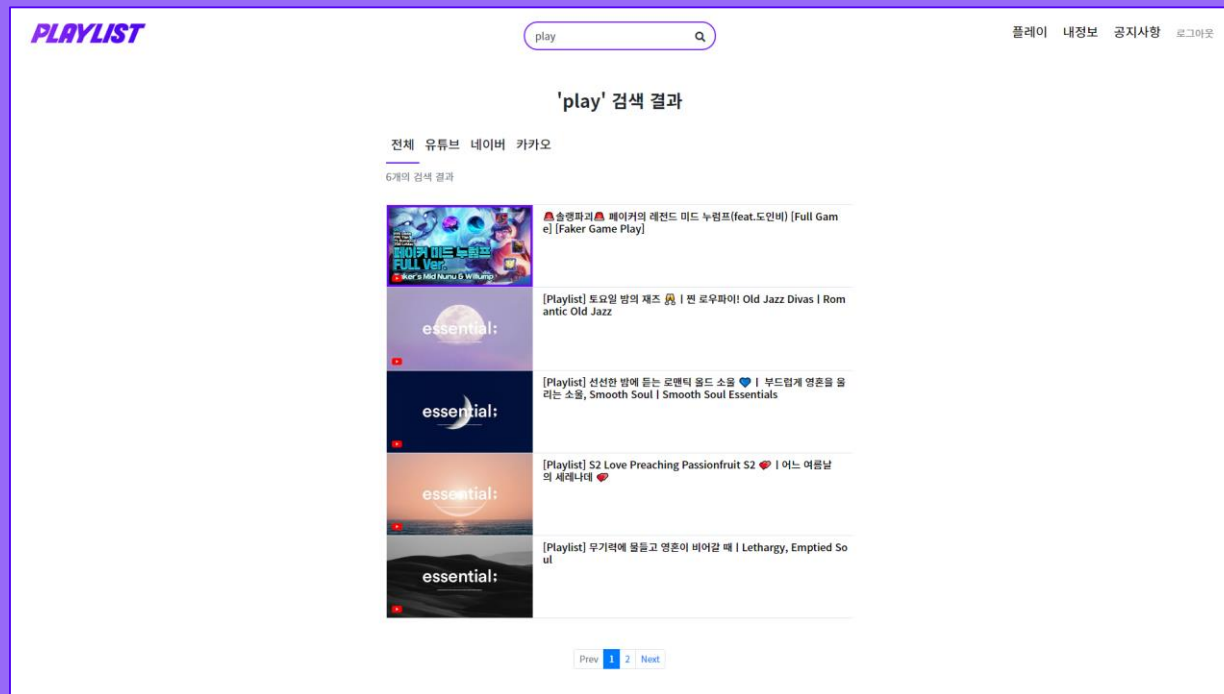
아쉬운 점

플레이리스트 영상을 정렬하는 기능이 없음

개선 방향

Drag & Drop을 통해 영상을 정렬할 수 있도록 개선 예정

#3. Search



나만의 플레이리스트에서 키워드&플랫폼을 통한 검색 기능

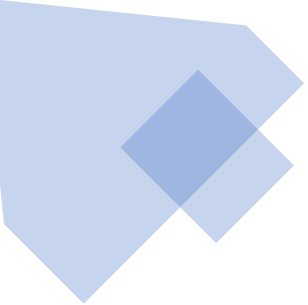
#3. Search

아쉬운 점

현재 나만의 플레이리스트에 추가된 영상에서 검색이 수행됨

개선 방향

다른 사용자의 영상에도 검색이 가능하도록하고,
각 플랫폼 사이트로부터 검색 결과를 받아와 나의 플레이리스트에 바로 추가할 수 있도록 개선 예정



PLAYLIST **배포**



DISTRIBUTE

배포 시작

2020-08-07 기준으로 URL 배포 시작

수많은 피드백과 향후 개선될 서비스 목록 수집

최적화 과정에서 발생했던 서비스 오류 수정

RELEASE

실 서비스 배포가 가능한지

현재 구현된 서비스에서 가장 중요한 부분인 플레이어는

Youtube 플랫폼만 공식적으로 지원하는 상태임

이외 플랫폼에서 공식으로 연동 가능한 API가 없다면 문제가 있음

RELEASE

플레이어 이외에는 문제가 없는지?

현재 빠르게 발전하는 동영상 시장에 따라 가장 중요한 문제로는

동영상 저작권에 관한 문제도 존재함

해당 영상을 배포함에 있어 각 플랫폼에서 허용하는 범위에 위배될 수 있음

REVIEW

첫 프로젝트를 마무리하며

디자이너와 컨택하고, 기능 구현을 검토하고

작업 중간마다 기획과 다르게 수정할 내용이 생기고

배포 테스트를 진행하는 과정에서 발생하는 문제들

다소 어려움은 있었지만 그만큼 배울 점이 많고 재밌던 프로젝트

AFTER

이번 프로젝트를 진행하면서 처음 사용해본 기술이

절반 이상이라고 생각되는데,

이후 프로젝트에선 Angular와 Spring Boot 등

새로운 기술에 도전하고

더욱 최적화하고 더욱 예외 처리에 신경을 쓰고자 함

PLAYLIST

THANK YOU

김예승
dev_webius@kakao.com
010-2164-7336