

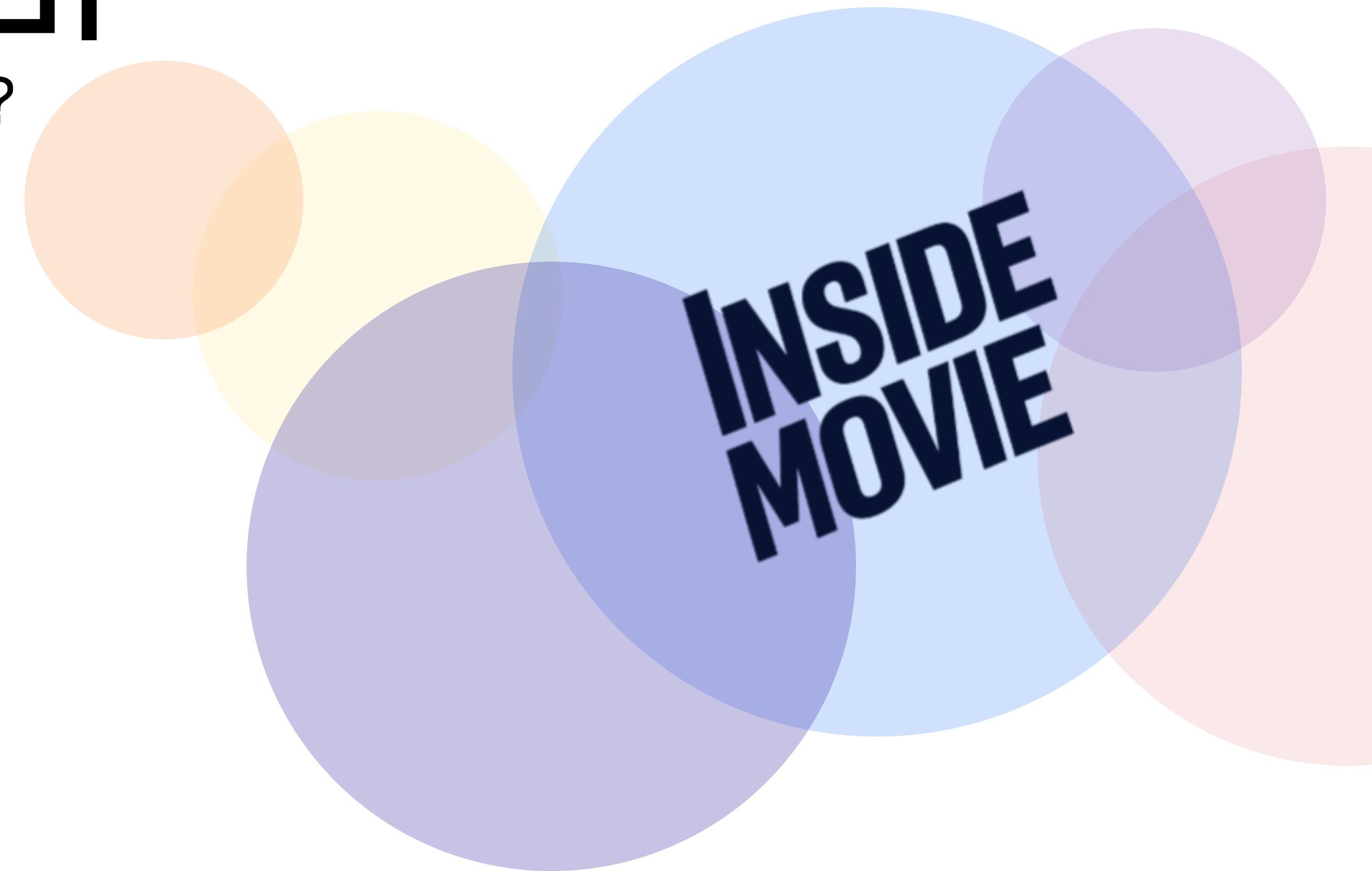
인사이트 무비

당신의 감정 취향은 무엇인가요?

현대오토에버 모빌리티 스쿨 웹/앱 2조

팀장 : 송창용

팀원 : 김소희, 김대환, 양지선, 이종진, 채상윤



발표 목차

01

프로젝트 개요

02

팀 구성 및 역할

03

수행 절차 및 방법

04

수행 경과

05

프로젝트 시연

06

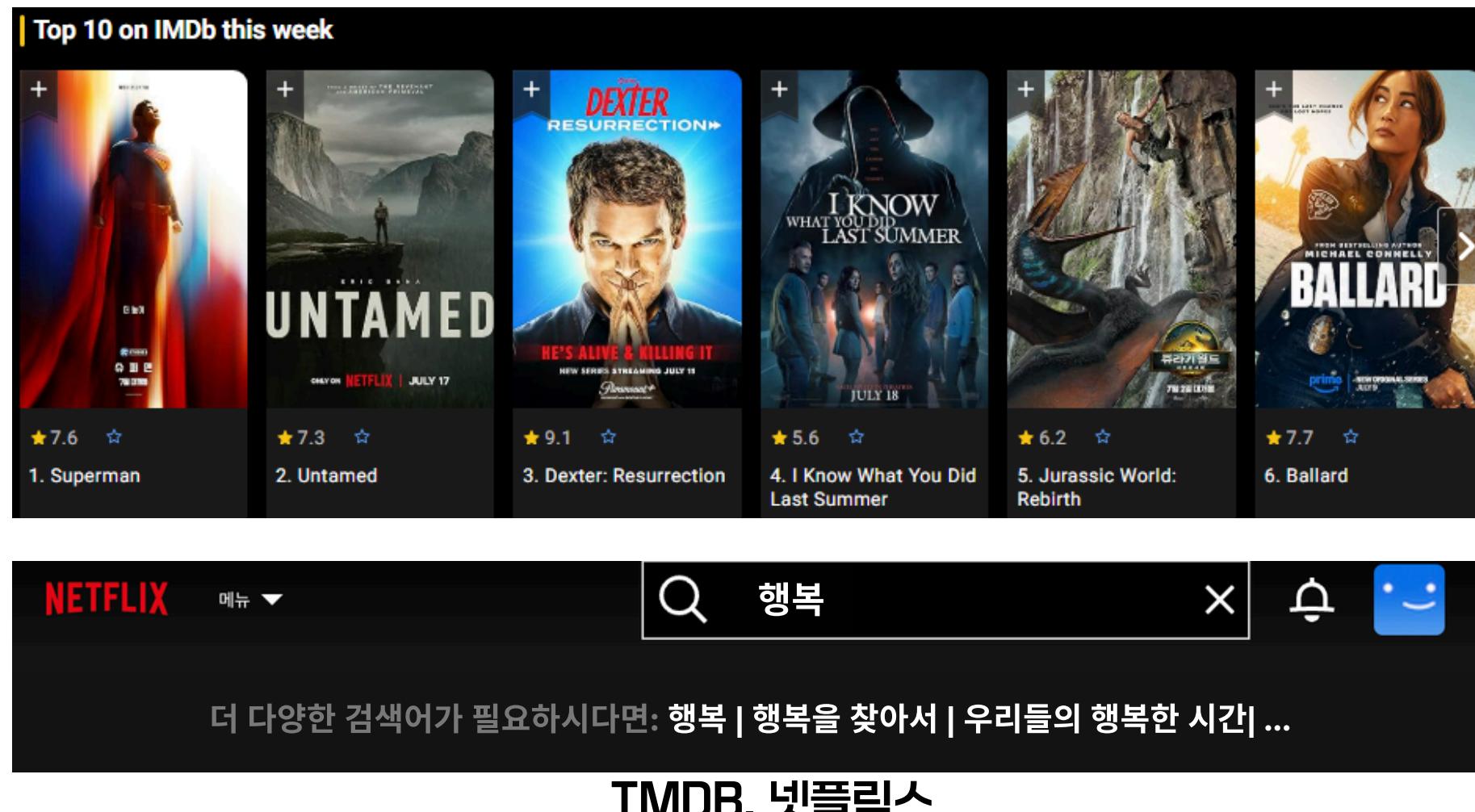
자체 평가 의견

프로젝트 개요

프로젝트 선정 배경 및 필요성

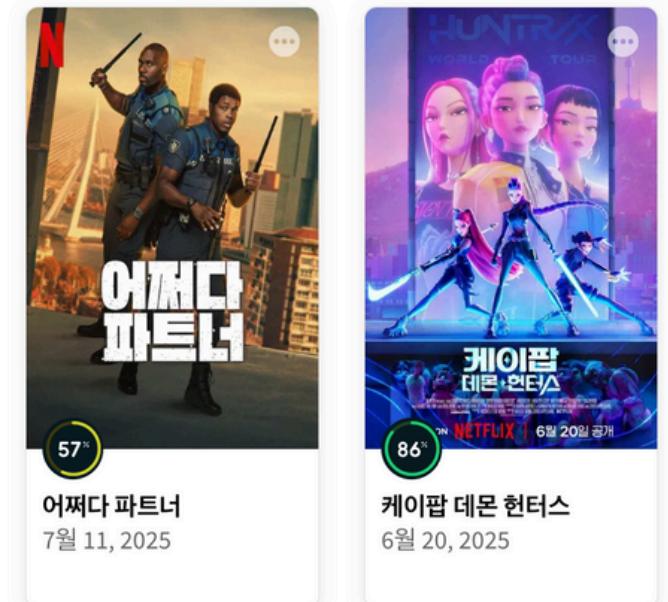
기존 영화 플랫폼의 한계

- 기존 영화 리뷰 서비스는 영화를 평점으로 평가
- 영화 추천은 사용자의 시청 기록에 의존



사용자 감정 기반의 맞춤 영화 서비스

- 실제 사용자의 감정을 복합적으로 단일 분류로는 사용자 공감형 콘텐츠 제안에 제약이 존재
- 사용자의 복합적인 감정을 고려하여 개인화 된 영화 추천 서비스



맞춤 영화 제안

프로젝트 개요

주요 기능

☑ 인공지능 기반의 복합 감정 리뷰 분석

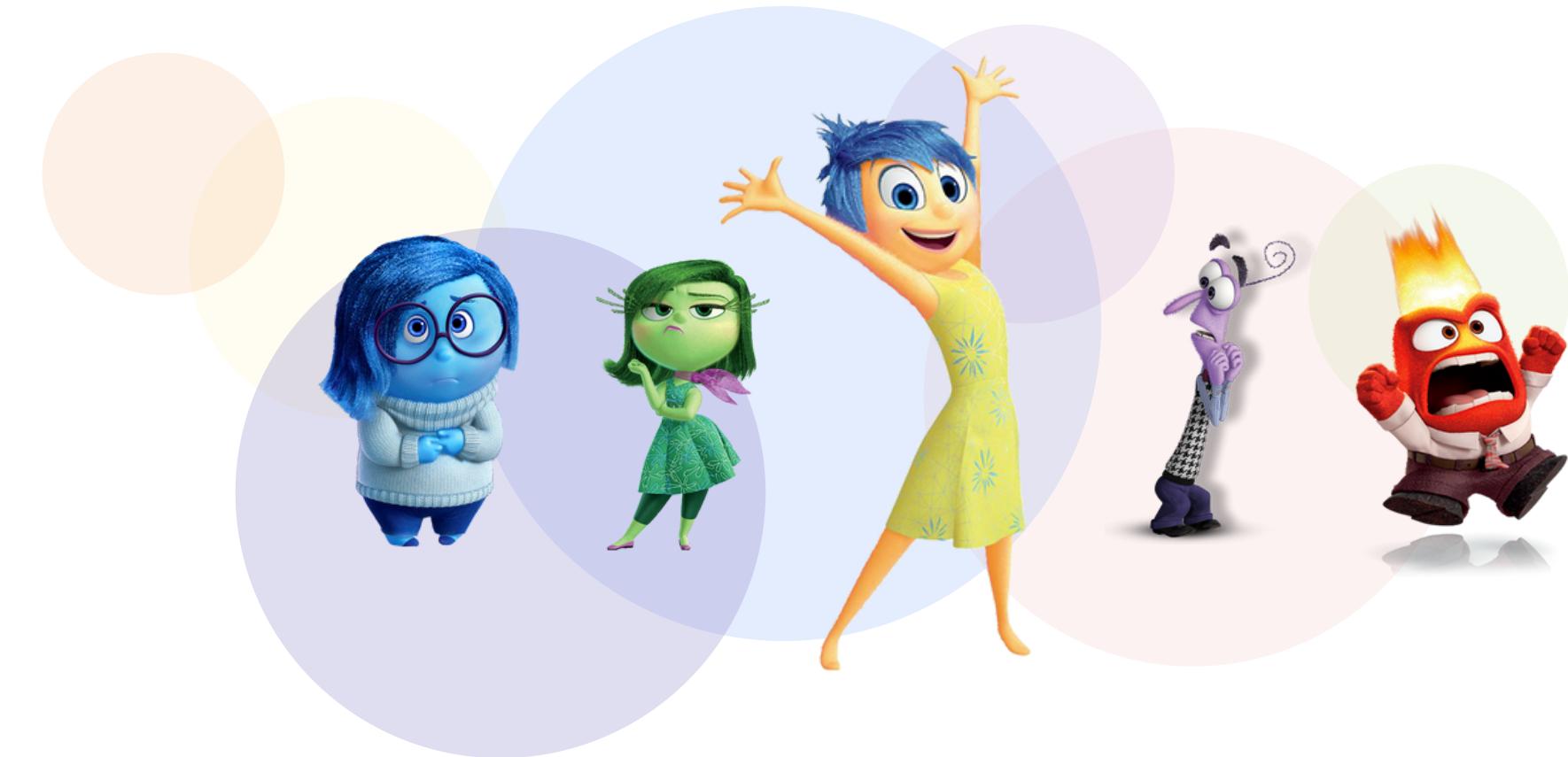
- KoBERT 모델을 이용하여 리뷰의 복합 감정 상태 분석
- 리뷰 집계를 통한 영화 감정 프로필 생성

☑ 사용자 감정 매핑 및 맞춤 추천

- 좋아요 한 영화를 기반으로 복합 감정 프로필 생성
- 사용자 감정 벡터와 영화 감정 벡터의 코사인 유사도 연산으로 관련도가 높은 영화 추천

☑ 주간 영화 대결 & 투표

- 스케줄러를 활용하여 일주일마다 자동으로 대결 생성
- 감정 상태별 대표 영화 세 가지 선정
- 역대 대결 내역 우승 영화 확인



팀 구성 및 역할



Backend
MSA / AI
CI / CD

송창용



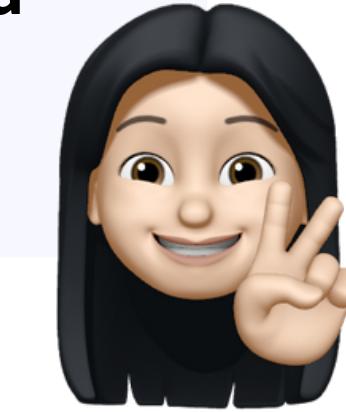
Frontend
Backend

채상윤



Backend

양지선



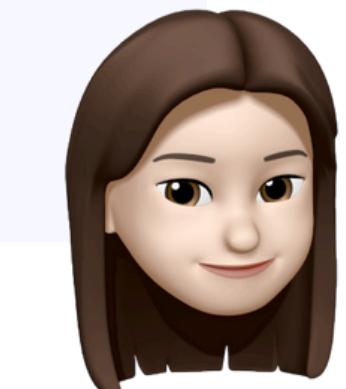
Backend
MSA / AI

김대환



Backend

김소희



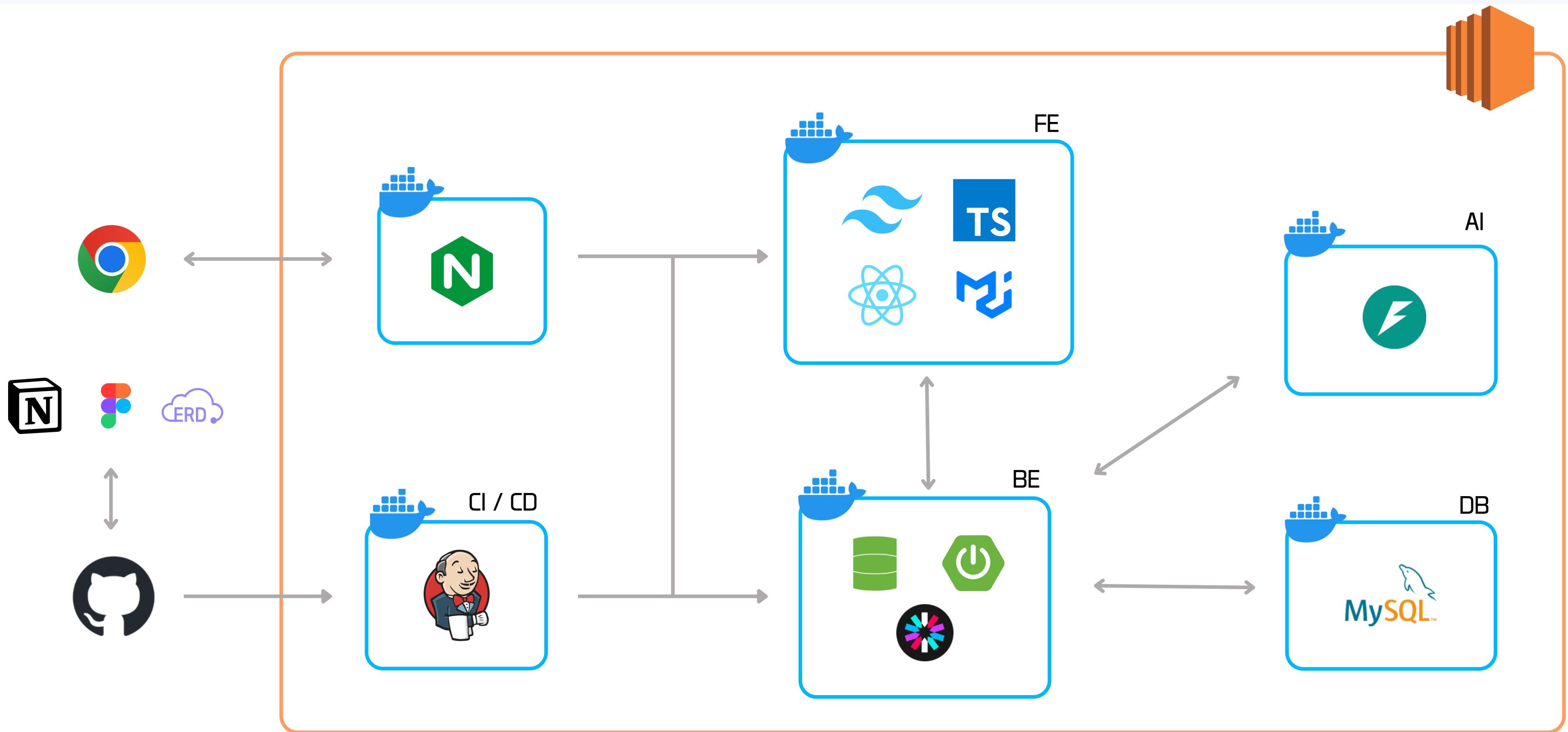
Frontend

이종진



프로젝트 개요

시스템 아키텍쳐

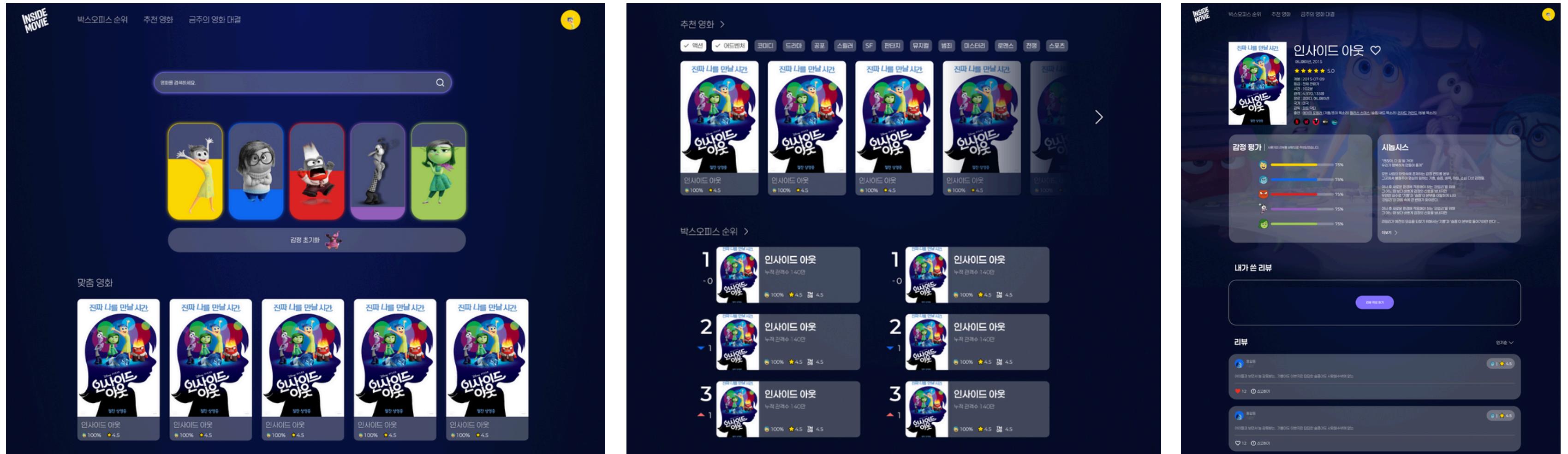


프로젝트 수행 경과

UI/UX

사용자 감정 프로필 기반 맞춤형 영화 추천 설계

- 피그마를 사용하여 UI/UX 설계 및 공유
- 5개의 감정(행복, 슬픔, 분노, 공포, 혐오) 캐릭터를 이용하여 맞춤 영화 제안
- 영화 상세 정보에서도 해당 영화의 감정 프로필 상태 확인



프로젝트 수행 경과

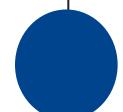
감정 분석

감정 분석 파이프라인



영화 리뷰 작성

사용자가 영화를 보고 느낀 감정을 리뷰로 작성



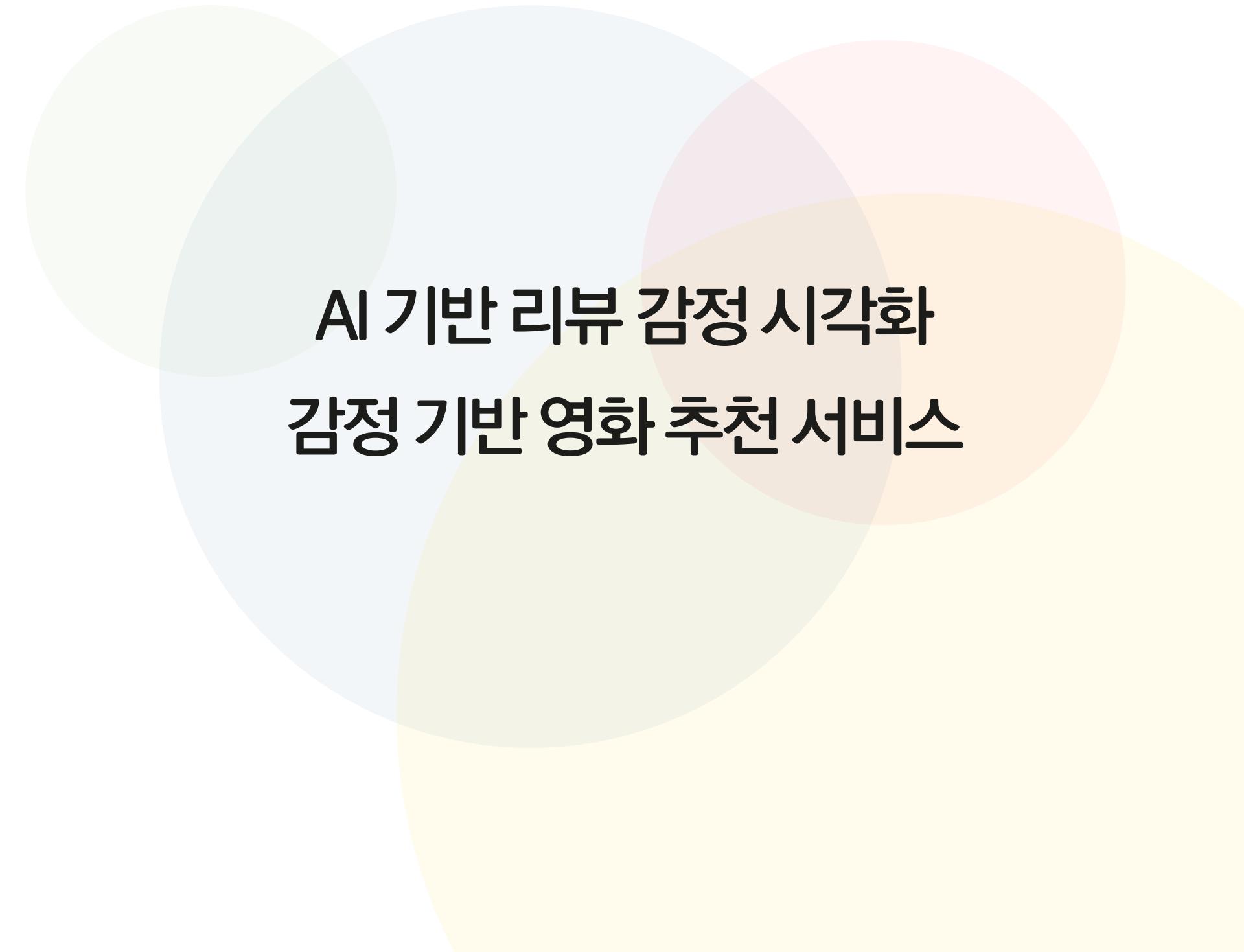
다차원 감정 프로필 생성

KoBERT를 이용하여 리뷰를 5가지 감정으로 분류
리뷰에 대한 5가지 감정 프로필을 생성



실시간 · 동적 감정 반영

리뷰의 감정을 바탕으로 영화의 감정 프로필 업데이트
영화를 좋아요 누른 사용자의 감정 프로필 업데이트



프로젝트 수행 경과

AI 모델 활용: KoBERT

KoBERT

- KoBERT는 SK 텔레콤이 한국어 위키 및 뉴스 말뭉치와 같은 대량의 한국어 데이터를 학습시킨 대규모 언어 모델입니다.
- 실제 응용 시에는 사용자의 의도에 맞게 파인튜닝 과정 필요
- 감정 분류를 위한 출력층을 추가, 데이터 세트에 맞게 모델을 재학습하여 성능 최적화



프로젝트 수행 경과

AI 모델 활용: KoBERT

모델 성능 개선

- 기본 모델의 다중 분류 테스트 결과 정확도가 54%로 실제 서비스 적용 어려움
- F1가 높은 4개 라벨(분노, 공포, 행복, 슬픔)을 제외하고 라벨을 하나씩 제거하며, 모델 학습 수행
- GPT 사용하여 실제 리뷰와 비슷한 1400개의 데이터 증강

7개 감정 라벨 학습 결과

	정밀도	재현율	F1 스코어	표본 수
분노	0.51	0.47	0.49	567
혐오	0.38	0.34	0.36	543
공포	0.57	0.58	0.57	547
행복	0.78	0.77	0.77	604
중립	0.41	0.39	0.4	483
슬픔	0.59	0.67	0.62	526
놀람	0.47	0.51	0.49	590

Test Accuracy: 0.5428

GPT 이용하여 데이터 증강

감정	데이터
분노	5,665
혐오	5,429
공포	5,468
행복	6,037
놀람	5,898
슬픔	5,267
중립	4,830
전체	38,594



감정	데이터
분노	5,915
혐오	5,429
공포	5,718
행복	6,287
놀람	6,148
슬픔	5,417
중립	5,080
전체	40,094

라벨을 변경하며 최종 모델 선택

기본 감정	추가 감정	정확도	평균 F1 스코어
분노, 공포, 행복, 슬픔	놀람, 중립	0.619	0.619
	중립	0.6877	0.6878
	혐오	0.6636	0.6532

모델 분류 정확도 12%p 개선

프로젝트 수행 경과

AI 모델 활용: KoBERT

리뷰 감정 분포 일반화

- 사용자 리뷰가 길어지면 감정 분류 모델이 사용자의 복합적인 감정을 고려하지 못함
- 문장 단위 평균화를 통해 특정 문장에 과도하게 편향된 감정 값을 완화
- 생성된 감정 프로필은 극단적인 한 문장으로 인한 노이즈를 줄이고, 사용자 전반의 감정 상태를 반영

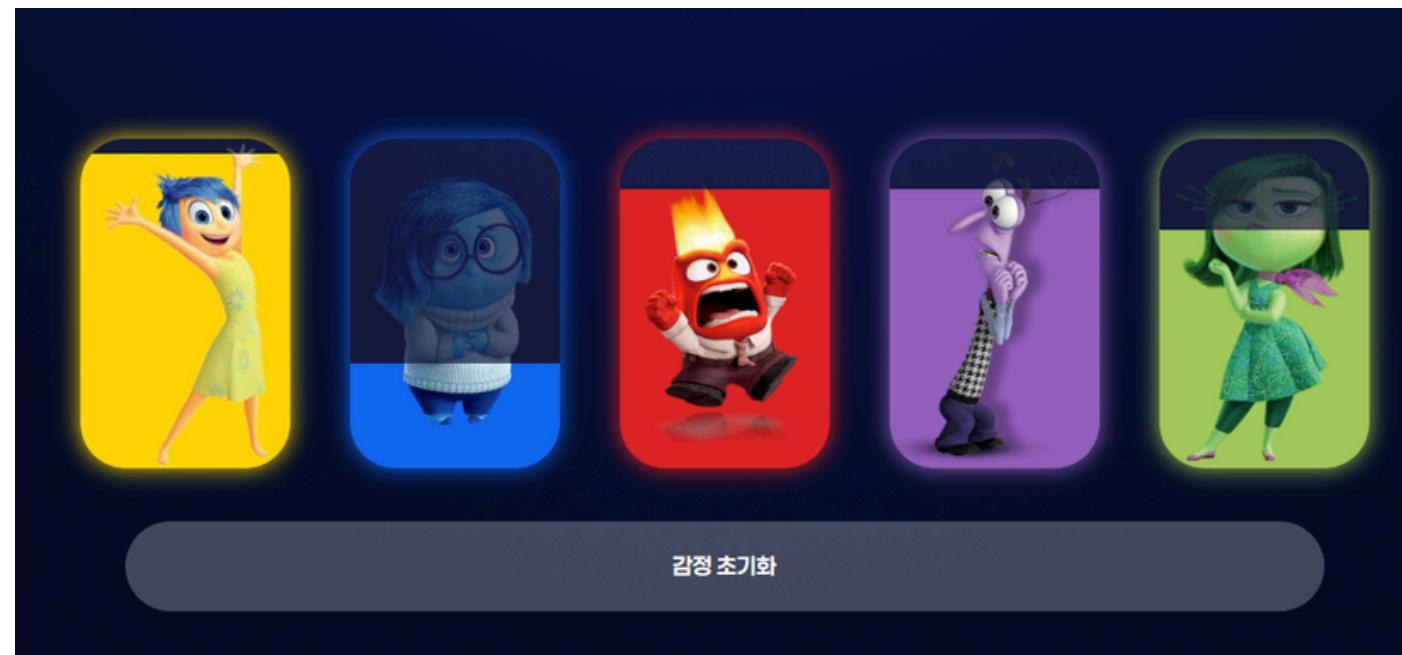
문장	분노	혐오	공포	행복	슬픔
(1) 와 종진이형 존잘이다. 종진이형의 눈물 연기는 슬픔 그 자체네. 종진이형 덕분에 하루가 행복해!!	0.09	0.16	0.07	98.05	1.64
(2) 와 종진이형 존잘이다	0.3	1.06	0.16	97.94	0.55
(3) 종진이형의 눈물 연기는 슬픔 그 자체네.	0.26	0.4	0.37	4.6	94.38
(4) 종진이형 덕분에 하루가 행복해!!	0.07	0.1	0.05	99.58	0.21
((1) + (2) + (3) + (4))의 평균값	0.24	0.58	0.19	68.42	30.57

프로젝트 수행 경과

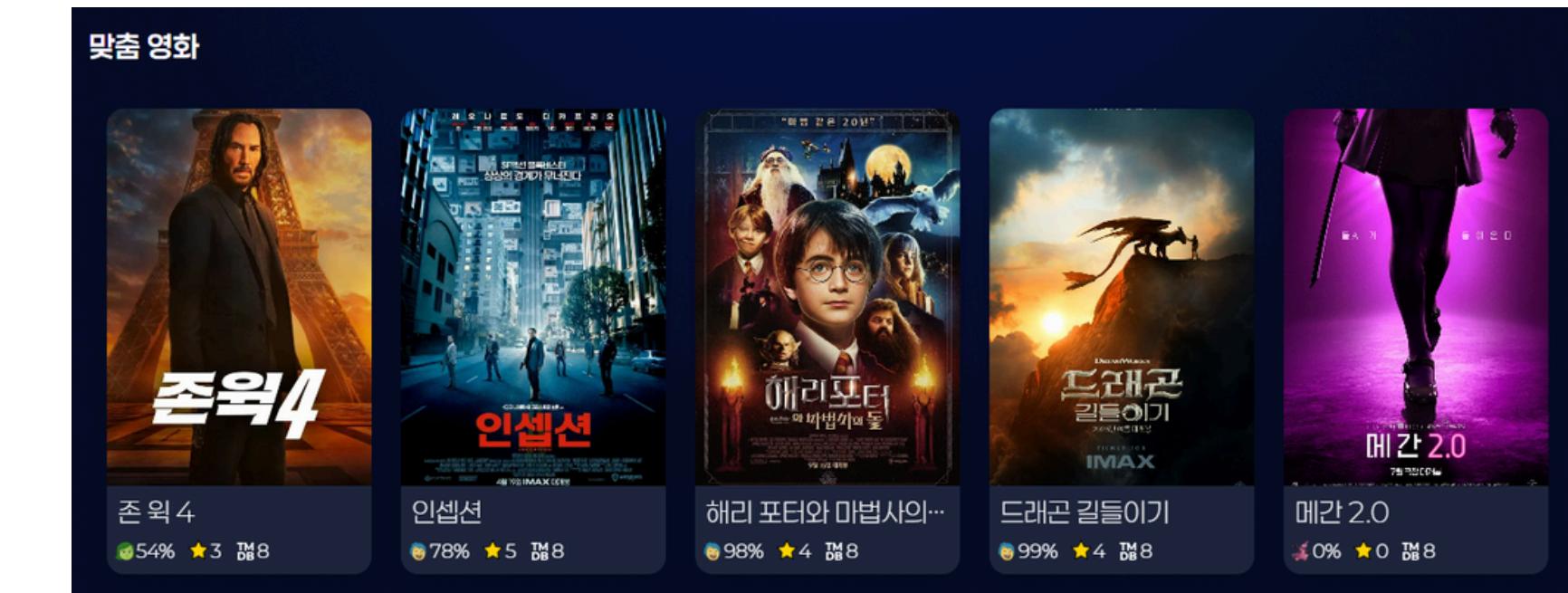
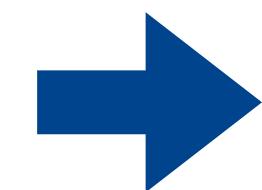
영화 추천

감정 상태 기반 맞춤 영화 추천

- 메인 화면의 감정 스크롤을 통해 감정 기반 영화 추천
- 사용자가 입력한 행복, 슬픔, 두려움, 혐오, 분노 값은 5차원 감정 프로필 벡터로 변환
- 사용자의 감정 상태와 영화의 감정 상태의 코사인 유사도를 비교하여 유사도 값이 높은 상위 10개의 맞춤 영화 제안

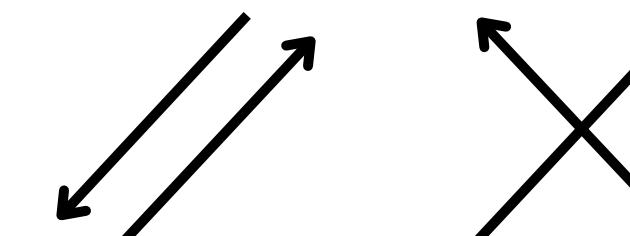


사용자의 5차원 감정 프로필

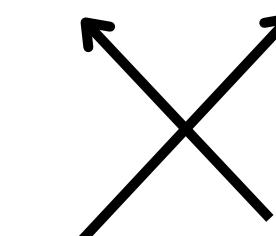


감정에 맞는 영화 추천

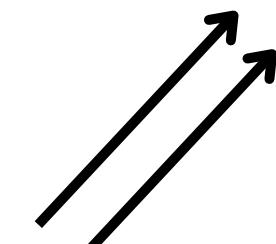
두 벡터간 코사인 값으로 유사한 정도 측정



$$\cos(\theta) = -1$$



$$\cos(\theta) = 0$$



$$\cos(\theta) = 1$$

코사인 유사도 예시

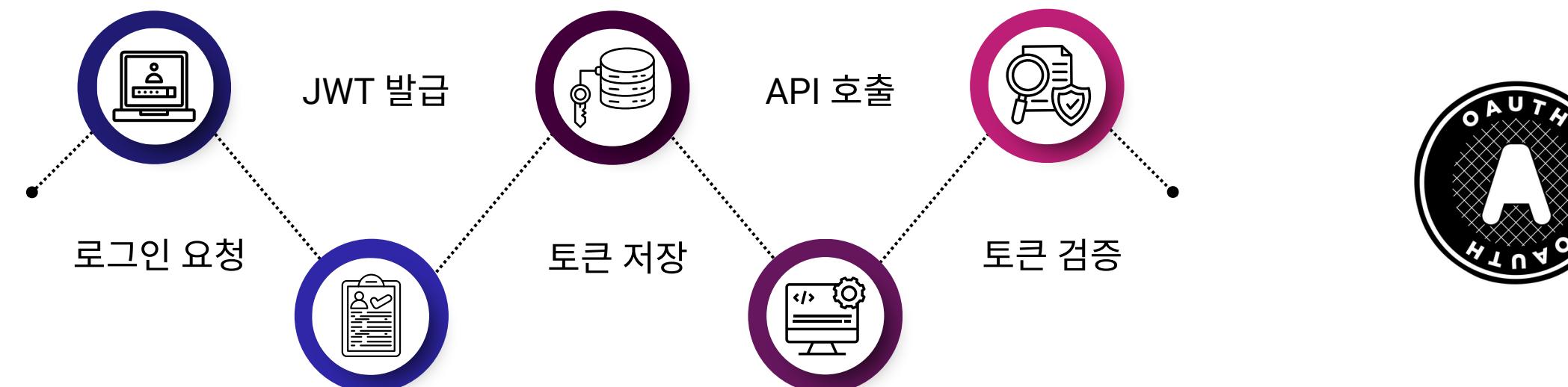
JWT 토큰을 이용한 인증 및 제 3자 인증

JWT (JSON WEB TOKEN)

- 토큰 기반 인증 방식으로 사용자 정보를 안전하게 전달
- 서버에서 JWT 토큰 발급 후, 클라이언트에서 이를 저장하여 API 요청 시 전달
- 서명된 JWT로 데이터의 무결성을 보장하고, 세션 관리와 보안성을 강화

OAuth 2.0 (제 3자 인증)

- KAKAO 계정을 통한 외부 인증으로 빠르고 간편한 로그인 제공
- 액세스 토큰을 발급하여 API 호출을 처리하고, 회원가입 절차 없이 사용자 편의성 및 보안성 향상



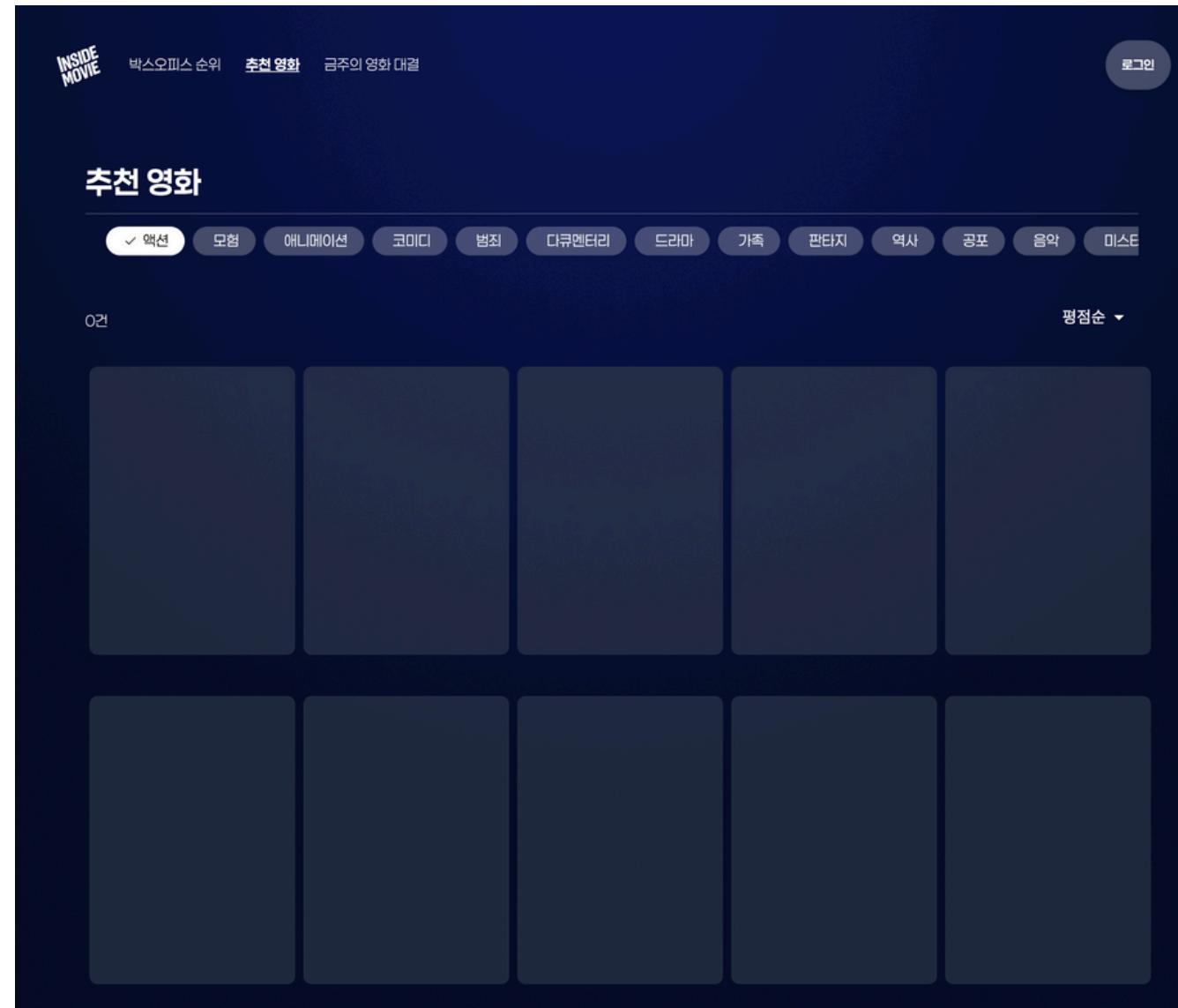
kakao

프로젝트 수행 경과

UI

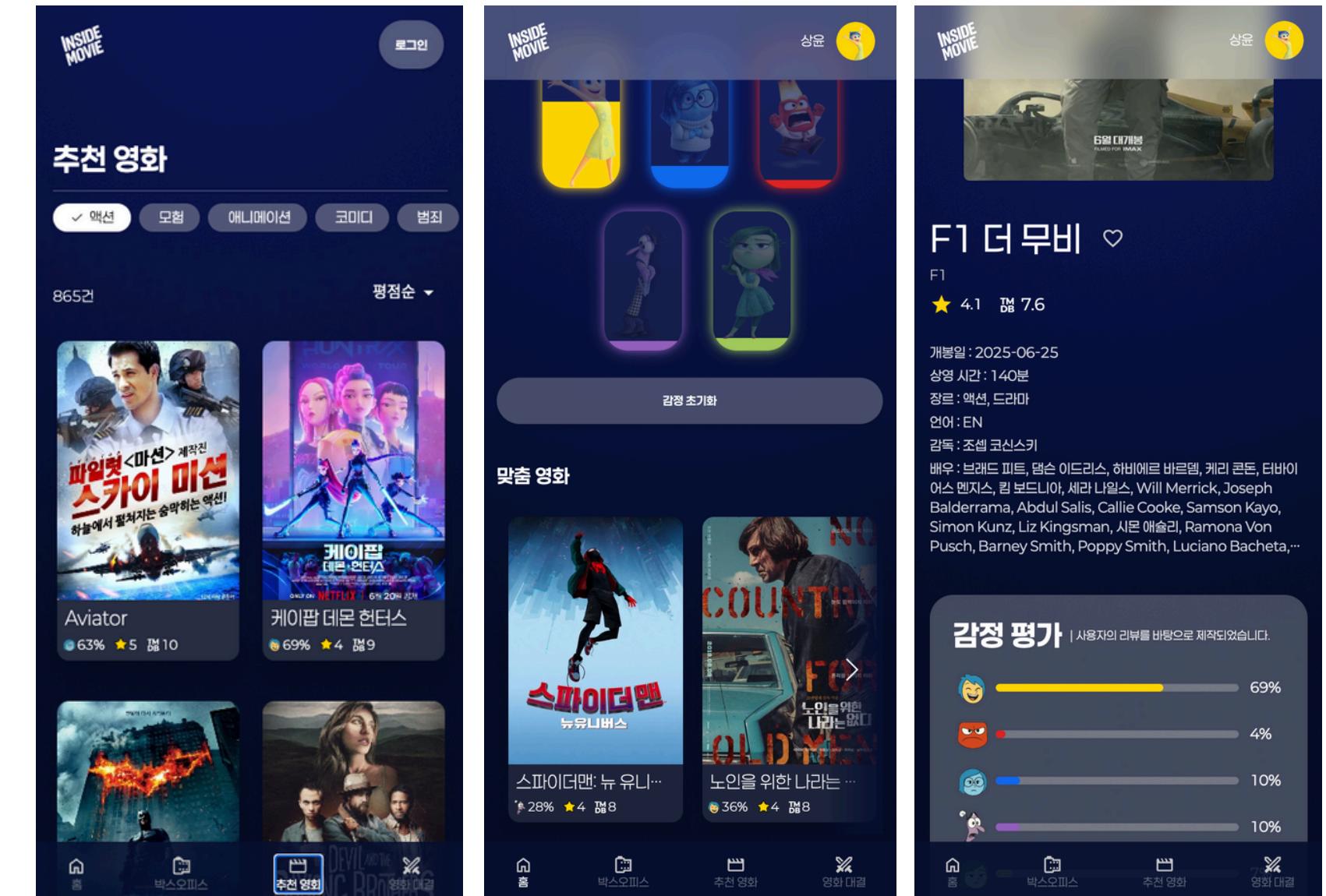
스켈레톤 UI

- 사용자 이탈률을 줄이기 위해 스켈레톤 UI 도입
- 스켈레톤 UI를 통해 애플리케이션이 대기 중임을 확인 가능



모바일 반응형 화면 구현

- 사용자의 일관된 경험을 지원하기 위해 모바일 반응형 화면을 구현

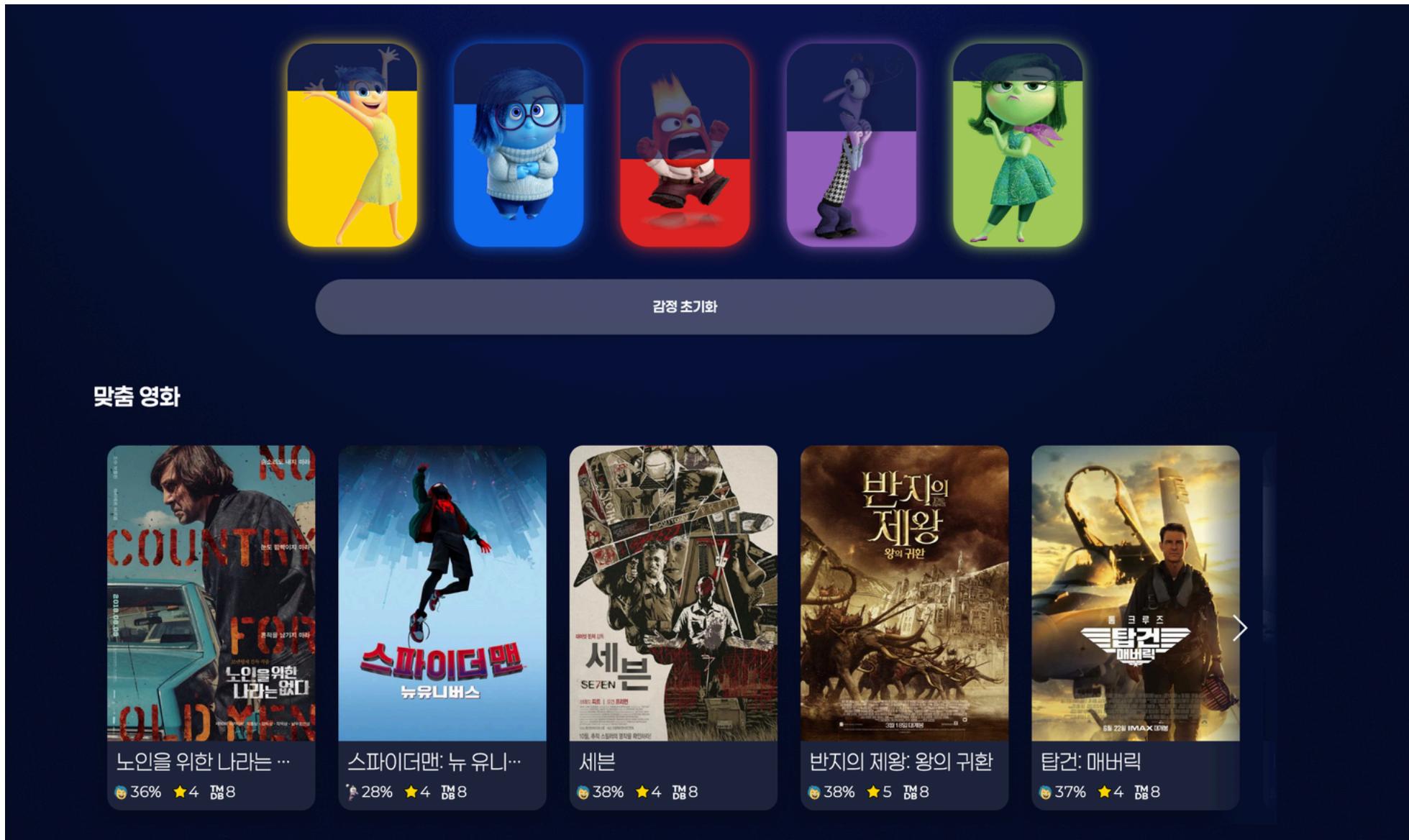


프로젝트 수행 경과

쓰로틀링

영화 감정 상태값 지연 전달 및 호출

- 감정 상태 값을 변경할 때 마다 API 요청으로 서버 부하 발생
- 감정 상태 값 변경 완료 시 맞춤 영화를 갱신하여 서버 부담 감소



MUI(Material UI Template)

MUI : React 기반 Material Design UI 컴포넌트 라이브러리

MUI 기본 제공 컴포넌트/훅

빠른 기능 구현 및 일관된 디자인 구현

ex

DataGrid

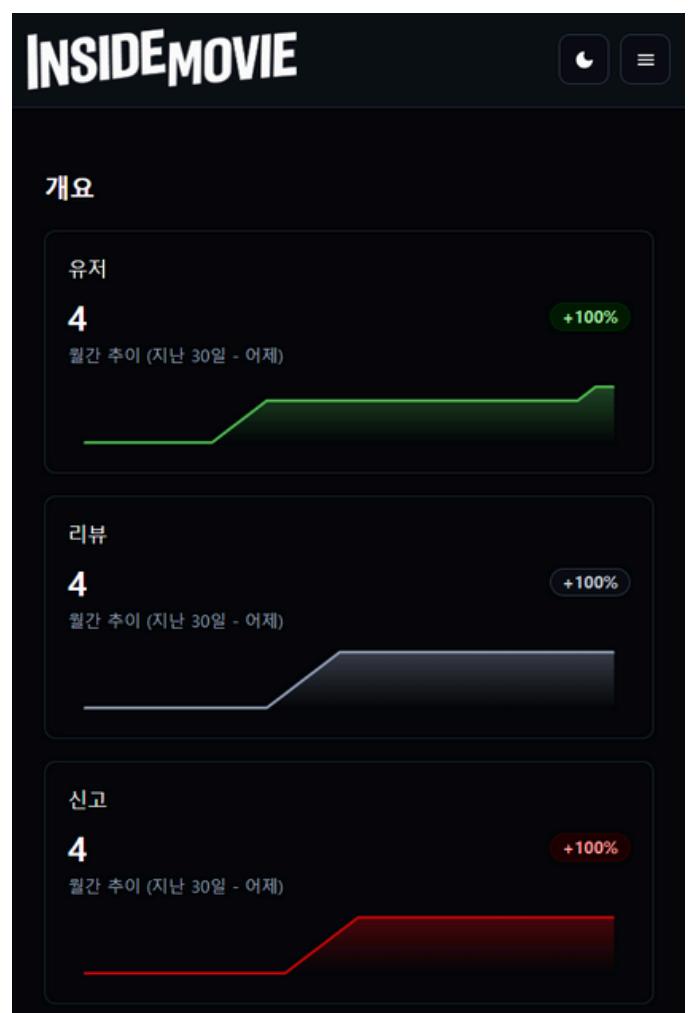
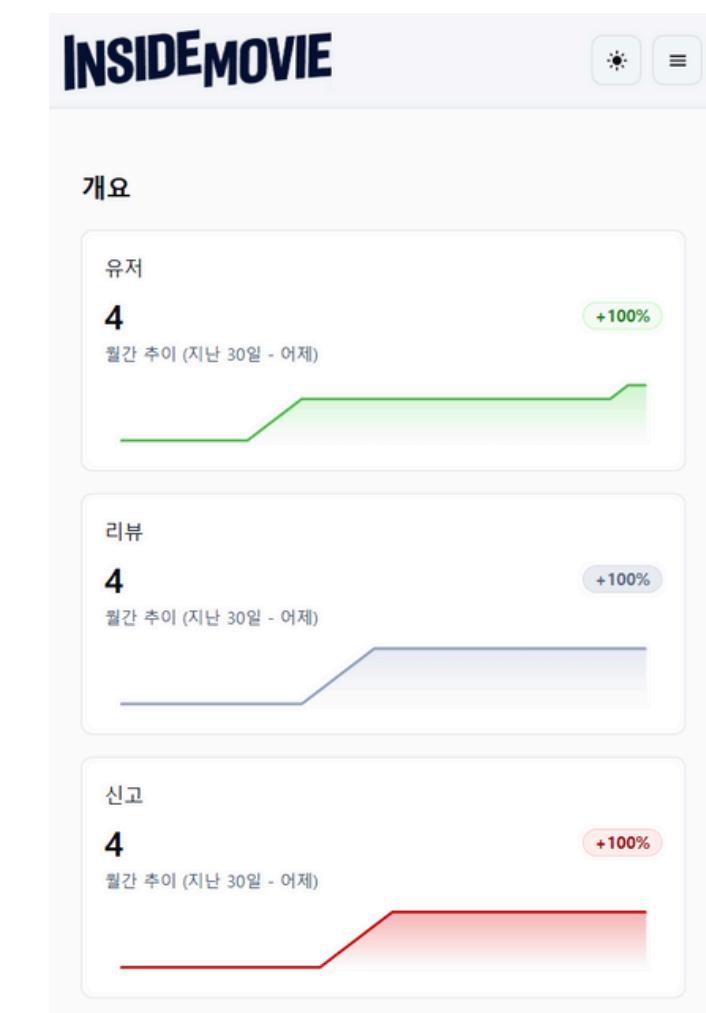
정렬, 필터, 페이징, 동적
클래스 등을 기본 지원하는 고
성능 표 컴포넌트

useColorScheme

현재의 색상 모드(light,
dark,system)를 읽거나
바꾸는 훅

▼ useColorScheme : system 색상 모드 반영

```
import { useColorScheme } from "@mui/material/styles";  
  
export default function ReportBoard() {  
  const { mode, systemMode } = useColorScheme();  
  const appliedMode = mode === "system" ? systemMode : mode;
```

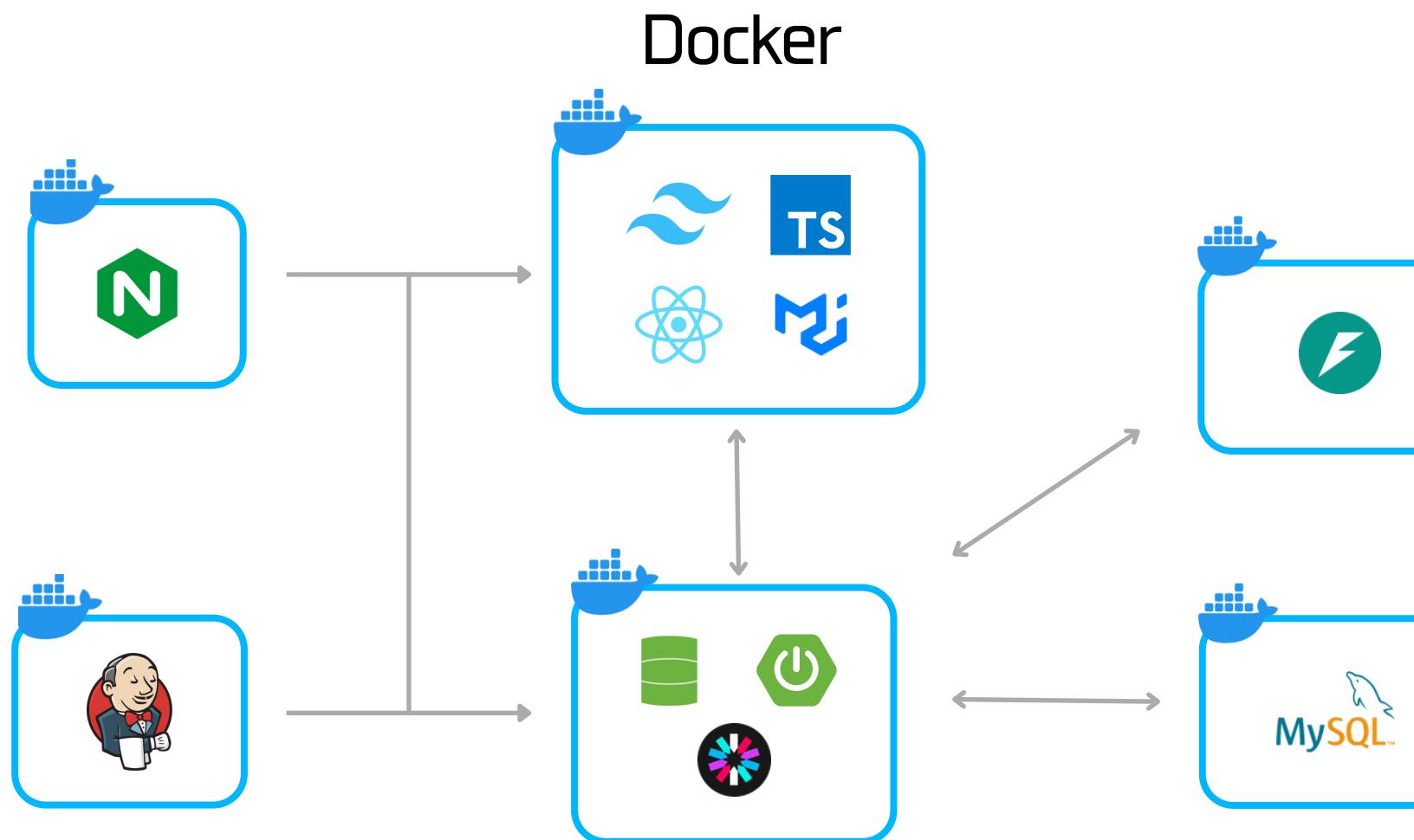


프로젝트 실행 경과

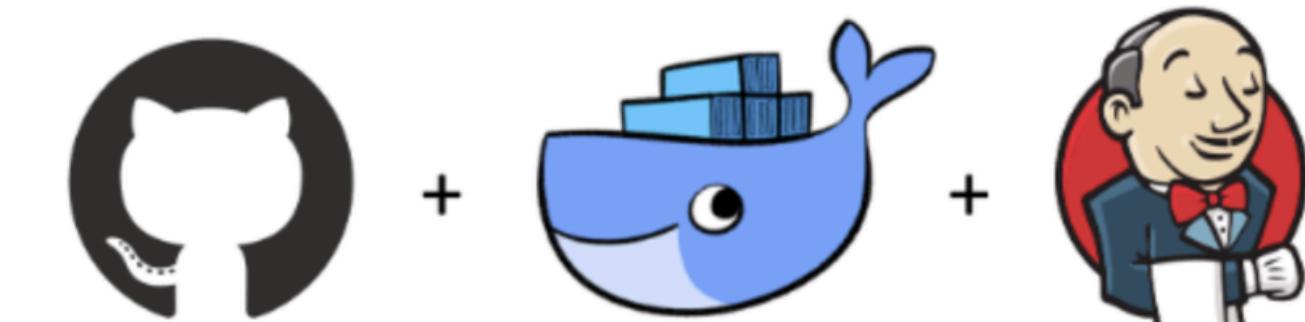
배포 자동화

Jenkins + Docker

- Jenkins 기반의 자동 배포 파이프라인을 구축하여, Git push가 발생했을 때 빌드 및 배포가 자동으로 수행
- Docker-Compose를 활용하여 BE, FE, AI 모듈을 컨테이너 단위로 독립 배포
- 환경 설정 관리와 재시작 이슈에 대응할 수 있는 안정적인 배포 구조를 설계
- 서버 운영 중 실시간 기능 반영 및 데이터 일관성 확보



Git - Docker - Jenkins



인공지능 모델 성능 향상

- 초기 모델(7가지 감정 기준)은 분류 정확도 51%, **F1 스코어** 53%로 저조한 성능을 기록
- F1 스코어를 기준으로 핵심 라벨을 선별**하여 분류 효율성을 개선
- 기존 데이터셋의 영화 리뷰가 부족하다는 점을 확인하고, **영화 리뷰 데이터를 증강**
- 근거있는 의사결정을 통해 지속적으로 모델을 최적화**
- 빠른 AI 서버 구성 및 안정적인 서비스 제공**

기본 감정	추가 감정	정확도	평균 F1 스코어
분노, 공포, 행복, 슬픔	놀람, 중립	0.619	0.619
	중립	0.6877	0.6878
	혐오	0.6636	0.6532

에러 핸들링

- HTTP 상태 코드 및 커스텀 에러 **코드를 표준화**하여 유지보수 효율성 향상
- Dialog 및 Snackbar 기반의 알림 UI를 통해 오류 원인과 해결 안내 구조화
- 프론트엔드와의 빠른 소통으로 오류 해결 시간 단축

```
{  
  "status": 200,  
  "success": true,  
  "message": "내가 관람한 영화 조회 성공",  
  "data": {  
    "content": [  
      {  
        "id": 1,  
        "posterPath": "https://image.tmdb.org/t/p/w500/8vywrRg1wr",  
        "title": "드래곤 길들이기",  
        "voteAverage": 8.08,  
        "mainEmotion": "NEUTRAL"  
      },  
      ...  
    ],  
    "pageNumber": 0,  
    "pageSize": 10,  
    "totalElements": 3,  
    "totalPages": 1,  
    "last": true  
  }  
}
```

응답 속도 개선(Faiss)

- 사용자 맞춤 영화 추천하기 위해 **사용자의 감정 벡터와 저장된 모든 영화와의 코사인 유사도 연산으로 서버 부하 발생**
- 배치 연산과 정규화를 통해 코사인 유사도 연산을 최적화 할 수 있는 Faiss(Facebook AI Similarity Search) 활용
- 기존 응답속도가 3~4s에서 0.5s~1s로 약 80% 개선**

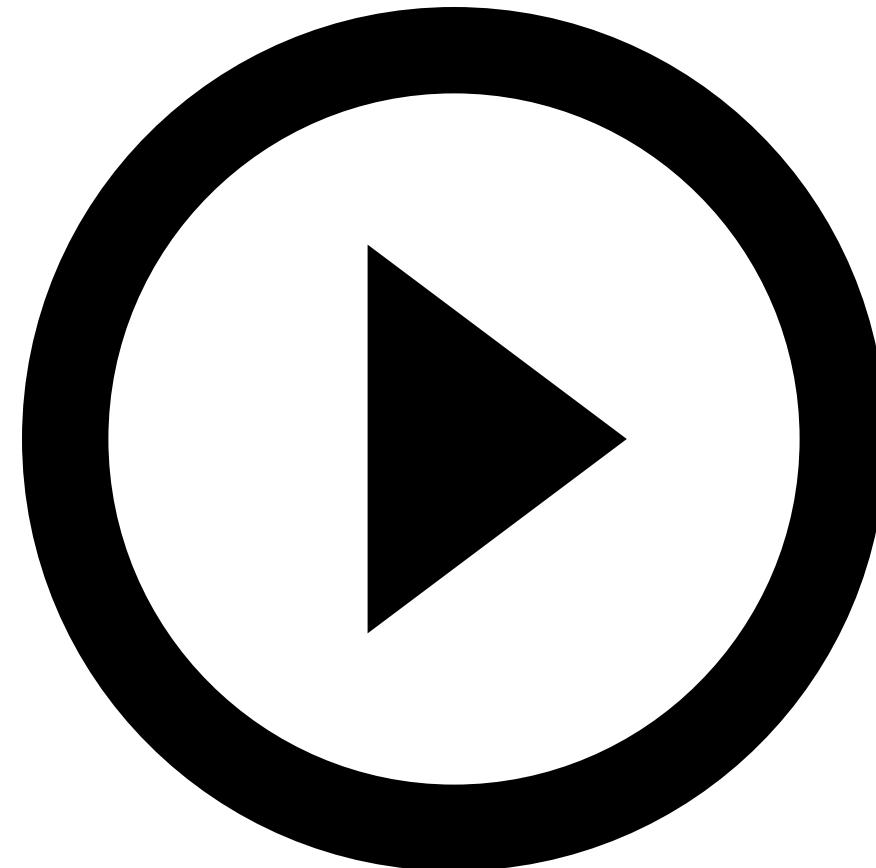


MUI

- MUI 컴포넌트 라이브러리와 사전 구성된 템플릿을 활용하여 공통 UI 요소 설계·구현·검증에 소요되는 반복 작업 단축
- 통일된 디자인 시스템 적용**을 통해 QA 및 유지보수 단계에서 발생하던 UI 비표준 이슈를 감소
- 색상·타이포그래피 등의 테마 관리로 협업자 간 디자인 편차를 제거하고 일관성을 확보



프로젝트 시연



자체 평가 의견

완성도 평가

- 맞춤 영화 제안 기능 구현 완료
 - KoBERT 기반 감정 분류 파이프라인 완성
 - 분류된 5차원 감정 프로필 간 코사인 유사도 계산으로 맞춤 영화 제안
- MSA 도입
 - FastAPI(인공지능)
 - Spring Boot(백엔드)
 - React + TypeScript(프론트엔드)
- 인증 · 인가 및 보안
 - JWT 토큰 기반 인증 체계 구현
 - OAuth2 소셜 로그인(카카오) 연동 완료

잘한 점

- 인공지능 모델 정확성 12% 향상을 통한 서비스 개선
- 에러 로그 관리 메시지를 통한 오류 해결 속도 개선
- 코사인 유사도 계산 성능 향상을 통한 응답 속도 개선

아쉬운 점

- 영화관/좌석 리뷰 및 친구 기능 미구현
- 영화 데이터 수집 시, 병렬 처리로 데이터 수집 속도 개선
- 다중 이벤트로 인한 동시성 문제

부족한점

- ▶ 영화 검색 시 영화 데이터베이스에 없는 영화에 대한 검색 기능 필요
- ▶ TMDB API의 초당 요청 제한을 초과하면 과부하 발생 가능성
- ▶ 동시 다발 리뷰 이벤트로 인한 감정 상태 데이터 요약 테이블 동시성 이슈 발생
- ▶ 감정 분류 모델 F1 스코어($\approx 61\%$)로 정확도 향상 필요

개선점

- ▶ 단위 시간동안 실행 횟수 제한할 수 있는 RateLimiter를 이용하여, 사용자가 동시에 데이터베이스에 없는 영화 검색 시 서비스 장애 발생하지 않도록 방지
- ▶ 비동기 이벤트 처리, 증분 업데이트 및 일괄 배치 집계를 활용하여 안정적인 감정 프로필 생성
- ▶ 다중 분류 모델 학습을 사용한 사용자 감정 복합 분석

감사합니다