

# FINAL PJT

# 영화 추천 알고리즘 기반 커뮤니티 서비스

# INDEX

- 영화 추천 알고리즘 기반 커뮤니티 서비스
  - 목표
  - 개발환경
  - 필수 요구사항
  - 제출

# 목표

## 프로젝트 목표

- 영화 데이터 기반 추천 서비스 구성
- 영화 추천 알고리즘 구성
- 커뮤니티 서비스 구성
- 서비스 관리 및 유지보수

# 개발 환경

## | 필수 개발 환경

- Python 3.9.x
- Django 3.2.x
- Node.js 16.x
- Vue 2 ❖ Vue 3 및 Pinia는 사용할 수 없습니다.

## | 사용 아키텍처

- 1과 2중 택 1
  1. Django & Vanilla JavaScript (또는 Vue)
  2. Django REST Framework & Vue



# 필수 요구사항

## | 개요

- 필수 요구사항은 프로젝트 내에 반드시 포함되어야 합니다.
- 이 외의 추가적인 기능 및 반응형 디자인 등은 팀 별로 자유롭게 수행할 수 있습니다.
  - i. 영화 데이터
  - ii. 영화 추천 알고리즘
  - iii. API
  - iv. 커뮤니티
  - v. README
  - vi. 기타

## 1. 영화 데이터

- 본 프로젝트는 '영화'를 주제로 진행되기 때문에, 영화 정보 데이터를 필수적으로 가지고 있어야 합니다.
- 최소 50개 이상 존재해야 합니다.
- 데이터는 fixtures를 사용하여 언제든지 load 될 수 있도록 준비합니다.

## 2. 영화 추천 알고리즘

- 추천 방식은 자유롭게 구성하되 해당 서비스를 이용하는 사용자는 반드시 최소 1개 이상의 영화를 추천 받을 수 있어야 합니다.
- 어떠한 방식으로 추천 시스템을 구현 했는지 기술적으로 설명할 수 있어야 합니다.

## 3. API

- 사용하는 API에는 제한이 없습니다.  
(TMDB, Youtube, Naver API 등)
- 다만 유료 API 서비스는 주의가 필요합니다.

## 4. 커뮤니티

- 유저간 소통 할 수 있는 커뮤니티 기능을 구현해야 합니다.  
(게시글, 댓글, 좋아요, 팔로우 등)
- 소통 방식은 자유롭게 구성합니다.

## 5. README

- 반드시 아래의 내용이 작성되어 있어야 합니다.
  - i. 팀원 정보 및 업무 분담 내역
  - ii. 목표 서비스 구현 및 실제 구현 정도
  - iii. 데이터베이스 모델링 (ERD)
  - iv. 영화 추천 알고리즘에 대한 기술적 설명
  - v. 서비스 대표 기능에 대한 설명
  - vi. 배포 서버 URL (배포했을 경우)
  - vii. 기타 (느낀 점, 후기 등)

## 6. 기타

- 최소한 5개 이상의 URL 및 페이지를 구성해야 합니다.
- HTTP Method와 HTTP response status code는 상황에 맞게 적절하게 반환되어야 하며, 필요에 따라 적절한 에러 페이지를 구성해야 합니다.
- .gitignore 파일을 사용하여 불필요한 파일 및 폴더는 제출하지 않도록 합니다.
- 프로젝트명
  - **final-pjt-front**, **final-pjt-back** (두 서버를 모두 사용하는 경우)



## | 7. 선택 요구사항

- 배포는 선택적으로 진행합니다.
- 배포 진행은 공용 노선의 해당 문서를 참고합니다.
  1. 서버(Django) 배포
  2. 클라이언트(Vue) 배포

# 제출

## | 제출

- 제출 위치 ❖ 두 곳 모두 제출하며 팀장이 대표로 제출합니다.
  1. 학사 시스템
  2. <https://lab.ssafy.com/>
    - 프로젝트 이름 : **final-pjt**
    - 반드시 각 반 담당 교수님 Maintainer 설정
- 제출 기한
  - 학사시스템에 안내된 제출기한