

종합 프로젝트

1. 목표

- 영화 정보 기반 추천 서비스 구성
- 커뮤니티 서비스 구성
- HTML, CSS, JavaScript, Vue.js, Django, REST API, DataBase 등을 활용한 실제 서비스 설계
- 서비스 관리 및 유지보수

2. 개발환경

A. 언어

- i. Python 3.7+
- ii. Django 3.X
- iii. Node 14.15.X
- iv. Vue.js 2.6+

B. 도구

- i. vsCode
- ii. Chrome Browser

C. 아키텍처

- i. Django & Vanila JS
- ii. Django REST API 서버 & Vue.js

3. 서비스 개요

- A. 본 프로젝트는 '영화'를 주제로 진행되기 때문에, 영화 정보 데이터베이스를 필수적으로 가지고 있어야 합니다.

- i. 데이터를 수집하는 방법은 제한없이 자유롭게 진행합니다.
 - ii. 영화 데이터 수집 예시는 다음과 같습니다.
 - 1. 영화진흥위원회
 - 2. 네이버 검색 API(영화)
 - 3. The Movie Database(TMDB)
 - iii. 최초에 각 팀 별로 등록된 영화 레코드(record)는 최소 50개 이상을 유지해야 합니다.
- B. 모바일 대응을 위한 반응형 웹, Django REST API 서버 및 프론트엔드 프레임워크(Vue.js) 분리 등의 상세 구현 방식은 자유롭게 구성하되, 프로젝트 README.md 상단에 프로젝트 구조에 대한 설명을 반드시 명시해야 합니다.
- C. 영화 커뮤니티에 필요한 기능을 구성하여야 합니다.
- D. 사용자에게 제공되는 영화 추천 방식은 자유롭게 구성하되 해당 서비스를 이용하는 사용자는 반드시 영화를 추천 받을 수 있어야 합니다.
- E. 최소한의 HTML/CSS를 통해 웹 사이트를 디자인 해야합니다.

4. 요구사항

아래의 세부 요구사항은 필수 기능으로 프로젝트 내에 반드시 포함되어야 합니다. 이외의 추가적인 기능 및 디자인 등은 팀 별로 자유롭게 수행할 수 있습니다.

A. 관리자 뷰

- i. 관리자 권한의 유저만 영화 등록 / 수정 / 삭제 권한을 가집니다.
- ii. 관리자 권한의 유저만 유저 관리 권한을 가집니다.
- iii. 장고에서 기본적으로 제공하는 admin 기능을 이용하여 구현합니다.
- iv. Vue.js를 활용하는 경우에도 Django admin기능을 이용하여 구현할 수 있습니다.

B. 영화 정보

- i. 영화 정보는 Database Seeding을 활용하여 최소 50개 이상의 데이터가 존재하도록 구성해야 합니다.
- ii. 모든 로그인 된 유저는 영화에 대한 평점 등록 / 수정 / 삭제 등을 할 수 있어야 합니다.

C. 추천 알고리즘

- i. 평점을 등록한 유저는 해당 정보를 기반으로 영화를 추천 받을 수 있어야 합니다.
- ii. 추천 알고리즘의 지정된 형식은 없으나, 사용자는 반드시 최소 1개 이상의 방식으로 영화를 추천 받을 수 있어야 합니다.
- iii. 추천 방식은 각 팀별로 자유롭게 선택할 수 있으며 어떠한 방식으로 추천 시스템을 구성 했는지 설명할 수 있어야 합니다.

D. 커뮤니티

- i. 영화 정보와 관련된 대화를 할 수 있는 커뮤니티 기능을 구현해야 합니다.
- ii. 로그인한 사용자만 글을 조회 / 생성 할 수 있으며 작성자 본인만 글을 수정 / 삭제 할 수 있습니다.
- iii. 사용자는 작성된 게시글에 댓글을 작성할 수 있어야 하며 작성자 본인만 댓글을 삭제 할 수 있습니다.
- iv. 각 게시글 및 댓글은 생성 및 수정 시각 정보가 포함되어야 합니다.

E. 기타

- i. 최소한 5개 이상의 URL 및 페이지를 구성해야 합니다.
- ii. HTTP Method와 상태 코드는 상황에 맞게 적절하게 반환되어야 하며, 필요에 따라 메시지 프레임워크 등을 사용하여 에러 페이지를 구성해야 합니다.
- iii. 필요한 경우 Ajax를 활용한 비동기 요청을 통해 사용자 경험을 적절하게 향상 시켜야 합니다.

5. 결과

- A. 제출 기한은 **20.11.26(목) 18:00** 까지 입니다. 제출기한을 엄수하여 주시기 바랍니다.
- B. 사전 안내된 UCC, 슬라이드를 포함한 자료를 제출해야 합니다.
- C. 반드시 README.md 파일에 아래의 내용이 기록되어 있어야 합니다.
 - i. 팀원 정보 및 업무 분담 내역
 - ii. 목표 서비스 구현 및 실제 구현 정도
 - iii. 데이터베이스 모델링(ERD)
 - iv. 필수 기능에 대한 설명
 - v. 배포 서버 URL
 - vi. 기타(느낀점)
- D. 요구사항에 명시된 사항은 최소 조건이며 추가적인 정보를 수정할 수 있습니다.
- E. 각 팀의 팀장은 <https://lab.ssafy.com>에 final-pjt 프로젝트를 생성하고 해당 프로젝트에 업로드하여 제출합니다.
 - i. Django와 Vue.js를 같이 사용한 경우 final-pjt-front, final-pjt-back 프로젝트를 생성하여 각각 업로드 합니다.
- F. 각 반 담당 교수님과 팀원을 Maintainer로 설정해야 합니다.
- G. 추후 지역별 담당 교수의 안내에 따라 추가적인 자료 제출이 요구될 수 있습니다.