

# 종합 프로젝트 안내

---

## 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 배포 및 관리

## 2. 개발 환경

---

### 1. Python Web Framework

- A. Django 2.1.15
- B. Python 3.7+

### 2. Vue 개발 환경

- A. Node 12.18.X
- B. Vue 2.6+

### 3. 개발 아키텍처

아래의 옵션 중 하나를 선택하여 구성하시오.

- A. Django & Vanila JS
- B. Django REST API 서버 & Vue.js

### 4. 서비스 배포 환경

아래의 옵션 중 하나를 선택하여 구성하시오.

- A. 서버: Ubuntu / Amazon Linux 등
- B. DB: MySQL / SQLite 등

### 3. 프로젝트 수행 정보

---

1. 기본적으로 팀은 2인으로 구성되며, 부득이한 경우 담당 교수의 안내를 받아 3인으로 구성하여 진행합니다.
2. 프로젝트 수행은 **6월 18일(목)** 까지이며, UCC 제출 기한은 **06월 17일(수) 23:59**, 프로젝트 발표는 **06월 19일(금)**에 운영 프로 및 담당 교수의 안내에 따라 진행합니다.
  - 최종 프로젝트 기한을 넘어가는 경우 불이익이 있으니 반드시 마감 기한을 지켜주세요.
3. 프로젝트를 진행하는 모든 팀은 09:30 ~ 18:00까지 반드시 프로젝트를 진행해야 합니다.
  - 09:30~10:00와 17:30~18:00은 프로젝트 진행 내용을 확인하는 팀 내부 회의 시간입니다.
  - 10:00 ~ 17:30까지 점심 및 휴게시간을 제외하고는 모두 팀별 개발을 진행합니다.
  - 부득이한 사정으로 프로젝트를 수행하지 못하는 경우 반드시 지역별 담당 프로 및 교수에게 사전 공지를 해야 합니다.
4. 해당 일정은 내부 사정에 따라 일부 변경될 수 있습니다.

### 4. 서비스 개요

---

1. 본 프로젝트는 '영화'를 주제로 진행되기 때문에, 영화 데이터베이스를 필수적으로 가지고 있어야 합니다.
  - 데이터를 수집하는 방법은 각 팀별로 제한없이 자유롭게 진행합니다.
  - 영화 데이터 수집 예시는 다음과 같습니다.
    - 영화진흥위원회
    - 네이버 검색 API(영화)
    - The Movie Database(TMDb)
  - 최초에 각 팀 별로 등록된 영화 레코드(record)는 **최소 50개 이상을 유지**해야 합니다.
2. 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 명시**해야 합니다.
3. 영화 커뮤니티에 필요한 기능을 구성하여야 합니다.
4. 사용자에게 제공되는 영화 추천 방식은 자유롭게 구성하되 해당 서비스를 이용하는 사용자는 반드시 영화를 추천 받을 수 있어야 합니다.
  - 예시 - 추천 알고리즘 활용, 다양한 추가 데이터를 활용한 형식의 추천, 날씨에 따른 장르 추천 등
5. 최소한의 HTML/CSS를 통한 웹 사이트 디자인을 해야합니다.
6. 완성된 서비스를 배포하고, 유지 보수를 진행합니다.

## 5. 서비스 필수 기능

아래 제시된 기능은 필수 기능으로 프로젝트 내에 반드시 포함되어야 합니다.

이 외의 추가적인 기능 및 디자인 등은 팀 별로 자유롭게 수행할 수 있습니다.

### 관리자 뷰

- 관리자 권한의 유저만 영화 등록 / 수정 / 삭제 권한을 가집니다.
- 관리자 권한의 유저만 유저 관리 권한을 가집니다.

### 영화 정보

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

### 추천 알고리즘

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

### 커뮤니티

- 영화 정보와 관련된 대화를 할 수 있는 커뮤니티 기능을 구현해야 합니다.
- 로그인한 사용자만 글을 조회 / 생성 할 수 있으며 작성자 본인만 글을 수정 / 삭제 할 수 있습니다.
- 사용자는 작성된 게시글에 댓글을 작성할 수 있어야 하며 작성자 본인만 댓글을 삭제 할 수 있습니다.
- 각 게시글 및 댓글은 생성 및 수정 시각 정보가 포함되어야 합니다.
- 아래의 심화 내용을 제외한 다양한 추가 기능을 구성할 수 있습니다.
  - (심화) 게시글 pagination 활용
  - (심화) 복수의 기능 게시판 구성
  - (심화) 권한을 나누어 유저 관리 (예 - 관리자, 스태프 등)

### 기타

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

이외의 추가적인 기능은 자유롭게 구성합니다.

## 6. 제출 방식

---

### edussafy 업로드

1. 사전 안내된 UCC, 슬라이드를 포함한 자료를 수요일 23:59까지 제출합니다.
2. 자세한 안내는 edussafy를 참고 바랍니다.

### 소스코드

1. 완성된 소스코드의 Gitlab 주소를 담당 교수님께 제출합니다.
2. 해당 Gitlab 저장소의 최상단에는 반드시 `README.md` 이 있어야 하며 아래의 내용이 기록되어 있어야 합니다.
  - 팀원 정보 및 업무 분담 내역
  - 목표 서비스 구현 및 실제 구현 정도
  - 데이터베이스 모델링(ERD)
  - 필수 기능
  - 배포 서버 URL
  - 기타(느낀점)
3. 추후 지역별 담당 교수의 안내에 따라 추가적인 자료 제출이 요구될 수 있습니다.
4. 매일 정해진 시간을 활용하여, 팀별 개발 진행 사항을 논의합니다.

## 7. 평가 기준

---

1. 완성도, 편리성, 발표, 팀웍 총 4개의 기준으로 평가가 진행됩니다.
2. 평가는 지역별 담당 교수님의 점수와 학생 상호 평가로 이루어집니다.