

PJT명	AI 기반 추천 서비스	
단계	[종합 서비스 개발]	
진행일자	2025.12.19	
예상 구현 시간	필수기능	2.5D
	추가기능	0.5D
	심화기능	1D

1. 목표

- 데이터를 기반으로 사용자에게 다음 선택을 추천하고, 다양한 기능을 제공하는 서비스를 구현할 수 있다.
- 작성된 요구사항 문서를 바탕으로 서비스를 구현할 수 있다.

2. 준비사항

1) 제공 파일

- 없음

2) 개발언어 및 툴

아래 기술 중 택 1

- Django & Vanilla JavaScript (또는 CDN 방식의 Vue)
- Django REST Framework & Vue SPA

3) 개발언어 및 툴

- Python 3.11
- HTML, CSS, JS
- Node.js LTS
- Visual Studio Code
- Google Chrome

4) 사용 라이브러리 / 오픈소스

- requests
- Django 5.2
- Bootstrap 5.3
- Vue 3
- Axios

3. 작업 순서

- 1) 팀원과 같이 필수 포함 요구사항을 확인하고, GitLab에 프로젝트를 생성한다.
 - 프로젝트 이름은 final-pjt로 지정한다.
 - 각 반 담당 강사님을 Maintainer로 설정한다.
- 2) 팀원과 합의하여 협업 방식을 결정하고, 요구사항을 구현한다.
- 3) 작성한 코드들을 정리하고, README를 작성한다.
 - .gitignore 파일을 활용하여 불필요한 파일 및 폴더는 제출하지 않는다.
- 4) README 작성이 완료되면 심화 학습을 진행한다.
- 5) 제출 기한에 맞춰 모든 산출물이 GitLab에 업로드 될 수 있도록 한다.

4. 요구사항

이전에 작성했던 프로젝트 기획과 설계 문서들을 바탕으로 최종 프로젝트를 구현해보자. 단, **아래에 작성된 필수 요구사항들은 포함한 뒤** 개별 설계에 따라 다양한 방향으로 서비스를 발전시켜야 한다.

- 요구사항 예시(참고용)
 - 아래의 내용을 참고하여 추가적인 아이디어에 대해 요구사항을 추가 또는 수정하여 기능을 구현한다. 단, **필수 기능은 구현해야 하며, 수정할 수 없다.**

번호	분류	요구사항명	요구사항 상세	우선순위
기능적 요구사항				
F01	기능 구현	사용자 추천	사용자에게 다음 확인할 정보, 행동 등을 추천하는 기능 포함	필수
F02	기능 구현	API 활용	다양한 API를 활용하여 직접 구현하기 힘든 기능을 포함	필수
비기능적 요구사항				
NF01	프로젝트 관리	Git 활용	개발자 간 작업 충돌이 일어나지 않게끔 Git을 활용하여 프로젝트 관리	필수
NF02	보안	API Key 관리	API Key를 비롯한 중요한 정보가 외부에 노출되지 않도록 관리	필수
NF03	서비스 품질	데이터셋 확보	양질의 서비스를 위한 충분한 양의 데이터 보유	필수
NF04	서비스 품질	페이지 다양성	사용자의 서비스 경험을 풍부히 할 수 있는 수준의 URL 및 페이지 구성	필수
NF05	설계	RESTful 원칙	RESTful 설계 원칙을 잘 따르도록 설계하여, 적합한 HTTP Method만 허용하고 상황에 맞는 응답을 하도록 구현	필수
NF06	배포	서비스 배포	AWS 등을 활용하여 인터넷을 통해 서비스에 접속할 수 있도록 배포	선택

1) 필수 요구사항

A. Git 활용

Git의 기능을 적극적으로 활용하여, 여러 개발자가 동시에 개발할 수 있는 환경을 조성한다.

- 요구사항 번호: NF01
- 팀원과 합의하여 branch 생성 원칙 정리하기
 - Gitflow, GitHub Flow 등을 참고하여 원칙을 정할 수 있음
 - branch 운영 중 문제상황이 발생할 경우 원인에 대한 분석을 꼭 진행할 것
- 팀원과 합의하여 commit 내역 기록 원칙 정리하기
 - 기능 개발, 버그 수정 등 그 목적이 명확히 드러날 수 있도록 commit 메시지를 작성할 것
 - 어느 시점에 commit을 남길지에 대하여 충분한 합의를 이룰 것
- 프로젝트 종료 시 초기에 세운 원칙이 잘 지켜졌는지 점검
 - README에 초기에 세운 원칙을 설명하여 정리할 것
 - 생성했던 branch 들에 대한 용도를 정리할 것
 - 작성된 commit들을 GitLab에서 확인하여 스크린샷으로 README에 포함시킬 것

B. API Key 관리

Django 프로젝트, Vue 프로젝트 등에서 API를 활용할 경우, API 키가 외부에 노출되지 않도록 주의하면서 프로젝트를 진행한다.

- 요구사항 번호: NF02
- Django의 경우 Python Dotenv 패키지 활용
- Vue SPA 프로젝트의 경우 .env의 변수를 노출시키는 Vite의 내장 기능을 활용

C. 데이터 확보

서비스의 원활한 운영을 위한 양질의 데이터를 확보할 수 있도록 한다.

- 요구사항 번호: NF03
- 다양한 공개 데이터셋, API를 활용하여 데이터 확보
 - 확보 가능하며 서비스 가능한 최대한의 데이터를 확보
 - load 가능한 fixture의 형태로 포함되어야 함
- 서비스 구성에 따라 주기적 업데이트 기능 포함 가능
- 수집하게 되는 데이터 형식에 주의할 것
 - 실제로 서비스에 활용되는 속성을 판단하고 진행
 - 제공하는 속성이 다른 API의 경우 그 성질을 비교하여 표준화를 진행할 것
- 수집한 데이터별 이용 허락 조건을 꼼꼼히 확인할 것

D. RESTful 원칙 준수

RESTful 설계 원칙을 준수할 수 있도록 노력한다.

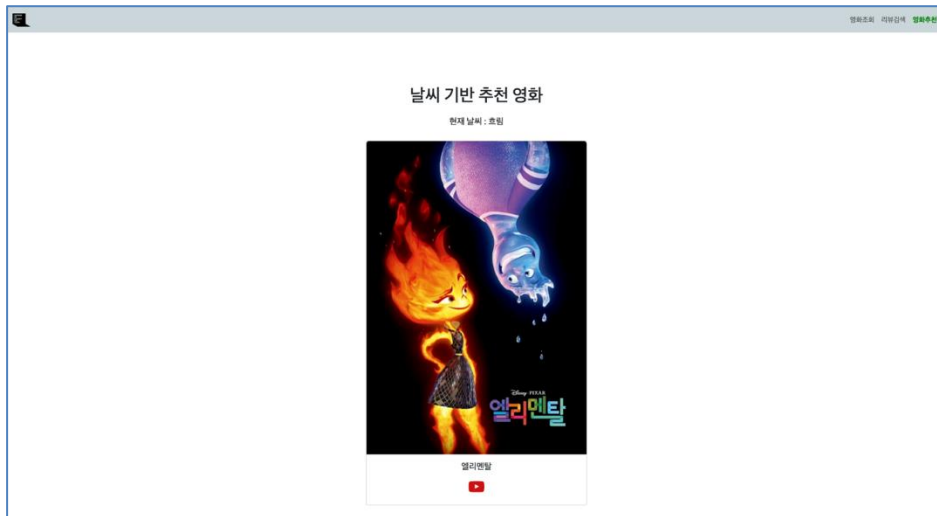
- 요구사항 번호: NF05
- 페이지를 보여주는 기능의 경우
 - URL이 사이트 구조를 잘 따르고 있는지
 - 페이지를 요청하는 URL이 제공되는 페이지의 내용과 맥락적 유사성을 가지고 있는지
- 동적으로 데이터를 주고받는 기능의 경우
 - URL이 데이터의 관계를 잘 표현하고 있는지
 - HTTP Method가 데이터베이스 변화와 연관성이 충분한지
 - Response Status Code가 서버의 처리 상태를 명확하게 표현하고 있는지
- 그 외의 RESTful 설계 원칙들을 찾아보고 적용
- 자신의 개발 환경에 따라 적용되는 내용이 다를 수 있음에 주의

E. 사용자 추천

사용자 정보와 수집한 데이터를 기반으로 사용자의 다음 행동을 권하는 기능을 포함하여 구현한다.

- 요구사항 번호: F01
- 특정 페이지 또는 상황에 따른 추천 기능이 동작해야 함
- 추천 방식은 자유롭게 구성 가능
 - 특정 페이지 이동시 추천
 - 현재 페이지와 연관된 데이터 추천
 - 문답을 통하여 추천
- 추천 알고리즘은 자유롭게 구현 가능
 - 단, 어떠한 방식으로 추천 시스템을 구현했는지 기술적으로 설명 가능해야 함

- 날씨 알고리즘을 바탕으로 다음 영화 추천 예시



- 팔로우한 유저가 추가한 도서를 바탕으로 다음 도서 추천 예시



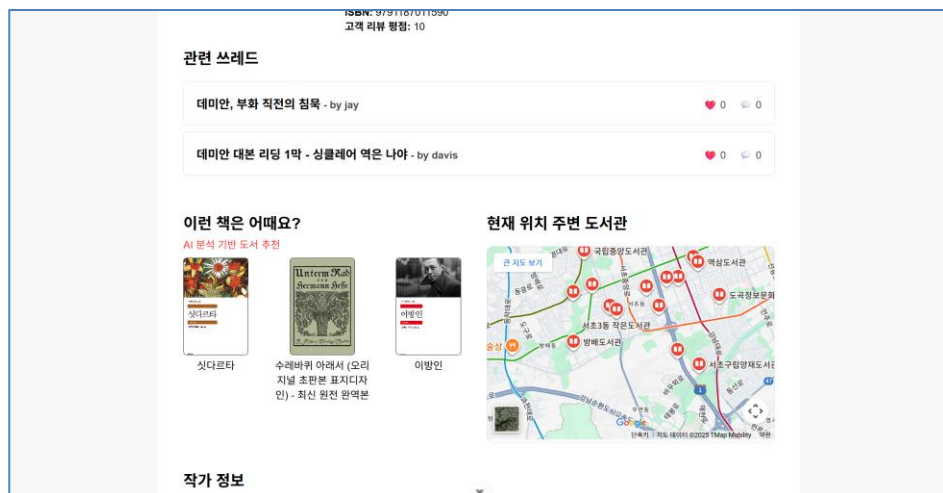
F. API 활용

다양한 API를 활용하여 직접 구현하기 힘든 기능을 포함해본다.

- 요구사항 번호: F02
- 데이터 수집, 영상 재생, 길찾기, 결제 등
- 단, 유료 API는 과금 주의
 - POC를 위한 테스트용 API 사용 가능
- 예시)
 - 예고편 재생을 위한 Youtube API



- 주변 도서관 정보 제공을 위한 지도 API



G. 페이지 및 기능 다양성

사용자의 서비스 체류 시간을 늘리기 위한 기능을 다양하게 구성하여 서비스에 포함시킨다.

- 요구사항 번호: NF04
- 사용자 경험을 위한 페이지 다양성 확보
 - 최소 5개 이상의 URL 및 페이지를 구성해야 함
 - load 가능한 fixture의 형태로 데이터 일부 포함

2) 선택 과제

서비스 배포를 진행해보자.

- 요구사항 번호: NF06
- 공용 문서를 참조하여 배포를 위한 다양한 서비스 확인
 - <https://abit.ly/pb-document>
 - 과금 정책에 주의
- 도메인을 추가하여 접근성 확보 가능
 - 도메인의 경우 일반적으로 연단위로 결제되므로 많은 주의를 요함

3) 생성형 AI 활용 관련

생성형 AI 도구를 활용하여 도전과제 요구사항을 해결해보자.

- 코드 생성, 아이디어 구상, 문제 해결 방법 탐색 등 다양한 방식으로 활용할 수 있다.
- GMS 시스템을 통해 제공된 Key를 활용하여 AI를 서비스에 적용할 수 있다.
 - 사용할 생성형 AI 모델은 자유롭게 선택한다.
 - 기능 구현에 의한 크레딧 사용량에 유의할 것
- 최종 결과물은 AI 생성 내용을 바탕으로 직접 수정 및 개선하여 적용해본다.

5. 참고자료

- Python dotenv
<https://pypi.org/project/python-dotenv/>
- Python OpenAI Library
<https://platform.openai.com/docs/libraries?language=python>
<https://github.com/openai/openai-python>
- MDN
<https://developer.mozilla.org/en-US/>
- Django
<https://www.djangoproject.com/start/overview/>
- Django REST Framework
<https://www.django-rest-framework.org/>
- dj-rest-auth
<https://dj-rest-auth.readthedocs.io/en/latest/>
- django-allauth
<https://docs.allauth.org/en/latest/>
- Vue 3
<https://vuejs.org/guide/introduction.html>
- Axios
<https://axios-http.com/kr/docs/intro>
- Bootstrap
<https://getbootstrap.com/>
- TMDB API
<https://developer.themoviedb.org/reference/intro/getting-started>
- Youtube Data API
<https://developers.google.com/youtube/v3/docs/search?hl=ko>

6. 결과

제출 기한은 진행일 18시까지이므로 제출 기한을 지킬 수 있도록 한다. 제출은 GitLab을 통해서 이뤄진다.

- 산출물

- 단계별로 구현 과정 중 학습한 내용, 어려웠던 부분, 새로 배운 것들 및 느낀 점을 상세히 기록한 README.md
- README.md에는 아래의 내용을 포함할 것
 - 팀원 정보 및 업무 분담 내역
 - 목표 서비스 및 실제 구현 정도
 - 데이터베이스 모델링 (ERD)
 - 추천 알고리즘에 대한 기술적 설명
 - 핵심 기능에 대한 설명
 - 생성형 AI를 활용한 부분
 - (배포했을 경우) 서비스 URL
 - 기타 포함시키고 싶은 내용
- 완성된 각 문제 별 소스코드 및 실행 화면 캡처본
- 이전까지 작성했던 기획 및 설계 문서 전체

- 제출

- 프로젝트 이름은 final-pjt로 지정 및 각자 제출
 - 한 명의 GitLab 계정에 Git 저장소 Push 및 작업
 - 나머지 인원은 제출 시 해당 저장소를 Fork 하여 제출
- 팀장의 경우 <https://edu.ssafy.com> 으로 기한을 맞춰서 제출
- 각 반 담당 강사님을 Maintainer로 설정

- 끝 -