

PJT명	프로젝트 기획과 협업	
단계	06 PJT	
진행일자	2025.11.07	
예상 구현 시간	필수기능	5H
	추가기능	1H
	심화기능	2H

1. 목표

- 데이터를 생성, 조회, 수정, 삭제할 수 있는 Web application을 제작한다.
- Django Model과 ORM에 대하여 이해한다.
- Django Form에 대하여 이해한다.
- Django Authentication System에 대하여 이해한다.
- Many to one relationship(N:1)에 대하여 이해한다.

2. 준비사항

- 1) 제공 파일
 - 없음
- 2) 개발언어 및 툴
 - Python 3.11
 - HTML, CSS
 - Visual Studio Code
 - Google Chrome
- 3) 필수 라이브러리 / 오픈소스
 - Django 5.2
 - Bootstrap 5.3

3. 작업 순서

- 팀원과 같이 요구사항을 확인하고, GitLab에 프로젝트를 생성한다.
 - 프로젝트 이름은 06-pjt로 지정한다.
 - 각 반 담당 강사님을 Maintainer로 설정한다.
- 팀원과 합의하여 협업 방식을 결정하고, 요구사항을 구현한다.
- 작성한 코드들을 정리하고, README를 작성한다.
 - .gitignore 파일을 활용하여 불필요한 파일 및 폴더는 제출하지 않는다.
- README 작성이 완료되면 심화 학습을 진행한다.
- 제출 기한에 맞춰 모든 산출물이 GitLab에 업로드 될 수 있도록 한다.

4. 요구사항

AI 기반 금융 데이터 분석 및 커뮤니티 서비스를 구축하려고 한다.

다양한 금융 상품(예금, 적금, 펀드 등) 데이터를 수집 및 관리하고, 이를 기반으로 사용자 맞춤형 금융 상품 추천 및 투자 성향 분석 기능을 설계·구현한다.

또한 사용자들이 금융 상품과 투자 전략에 대해 의견을 공유할 수 있는 커뮤니티 기능을 제공하여, 실제 투자자 간의 정보 교환과 소통이 이루어지도록 한다.

사용자는 회원가입 및 로그인을 통해 개인 프로필을 생성하고, 본인이 작성한 게시글 및 댓글을 프로필 페이지에서 모두 확인할 수 있다.

게시판에서는 다양한 금융 주제(상품 비교, 투자 팁, 금리 이슈 등)에 대해 자유롭게 글을 작성, 조회, 수정, 삭제할 수 있다.

팀원과 상의하여 아래 요구사항을 만족할 수 있도록 요구사항 명세서를 작성 및 구현해보자. 서비스의 인증 시스템을 구성하는 단계이다.

금융 게시판 데이터의 CRUD(생성, 조회, 수정, 삭제)가 가능하며, 로그인, 로그아웃, 회원가입, 회원정보수정, 비밀번호 변경 기능이 포함된 애플리케이션을 완성한다.

● 요구사항 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	프로젝트	프로젝트 구성	금융 커뮤니티 서비스 구현을 위한 Django 프로젝트 및 앱(accounts, posts, finances) 생성	필수
F02	accounts	Model Class	사용자 데이터를 데이터베이스에 저장할 수 있도록 AbstractUser를 상속받는 Django User 모델 클래스 구현	필수
F03	accounts	Form Class	사용자 입력 데이터 검증을 위해 Django 기본 Form 클래스(UserCreationForm, AuthenticationForm)를 커스터마이징하여 구현	필수
F04	posts	Model Class	금융 게시글 데이터를 데이터베이스에 저장할 수 있도록 Django Post 모델 클래스 구현 (제목, 내용, 작성자, 작성일 등)	필수
F05	posts	Model Class	댓글 데이터를 데이터베이스에 저장할 수 있도록 Django Comment 모델 클래스 구현 (게시글, 작성자, 내용, 작성일)	필수
F06	posts	Form Class	게시글 및 댓글 작성 시 입력 데이터 검증을 위한 Django ModelForm 클래스 구현	필수
F07	accounts	view 함수 login	로그인 UI 출력, 유효성 검증 및 세션 데이터 저장 기능 구현	필수
F08	accounts	view 함수 logout	현재 인증된 사용자의 세션 정보를 삭제하는 로그아웃 함수 구현	필수
F09	accounts	view 함수 signup	회원가입 UI 출력, 유효성 검증 및 회원 데이터 저장 기능 구현	필수
F10	accounts	view 함수 update	회원정보 수정 UI 출력, 유효성 검증 및 회원 데이터 수정 기능 구현	필수
F11	accounts	view 함수 delete	현재 인증된 사용자의 계정을 삭제하는 view 함수 구현	필수
F12	accounts	view 함수 change_password	비밀번호 변경 UI 출력, 유효성 검증 및 비밀번호 변경 기능 구현	필수
F13	posts	view 함수 index	전체 금융 게시글 목록을 조회 및 출력하는 view 함수 구현	필수

F14	posts	view 함수 create	게시글 입력 UI를 제공하고, 입력된 데이터를 저장하는 view 함수 구현	필수
F15	posts	view 함수 detail	단일 게시글 데이터를 조회하고 댓글을 함께 출력하는 view 함수 구현	필수
F16	posts	view 함수 update	게시글 수정 UI를 제공하고, 수정된 데이터를 저장하는 view 함수 구현	필수
F17	posts	view 함수 delete	단일 게시글을 삭제하는 view 함수 구현	필수
F18	posts	view 함수 comment_create	댓글 데이터를 입력 받아 저장하는 view 함수 구현	필수
F19	posts	view 함수 comment_delete	댓글을 삭제하는 view 함수 구현	필수
F20	profiles	view 함수 profile	로그인한 사용자의 프로필 페이지 구현 사용자가 작성한 게시글과 댓글을 모두 확인 가능	추가
F21	AI 활용	금융 데이터 요약	OpenAI API를 활용하여 게시글 내용을 요약하거나 감정 분석하는 기능 도전	심화
F22	AI 활용	금융 상품 추천	생성형 AI를 활용하여 사용자의 관심 카테고리에 맞는 금융 상품 추천 기능 도전	심화
F23	AI 활용	커버 이미지 생성	게시글 작성 시 금융 테마에 맞는 커버 이미지를 생성형 AI로 자동 생성	심화
F24	AI 활용	User 모델 확장	사용자 투자 성향, 자산 규모 등 추가 정보를 저장·관리할 수 있는 커스텀 필드 구현	심화
비기능적 요구사항				
NF01	프로젝트 관리	Git 활용	개발자 간 충돌이 없도록 Git을 활용하여 버전 관리 수행	필수
NF02	유지보수	URL 구성	URL 구조를 일관성 있게 구성하여 유지보수가 용이하도록 설계	필수
NF03	보안	HTTP Method 허용	허용된 HTTP Method(GET, POST, DELETE, PUT)만 접근 가능하도록 설정	필수
NF04	문서화	README 작성	구현 기능 설명, 학습 내용, 느낀 점을 포함한 README.md 작성	필수

1) 프로젝트 관리 (필수)

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

2) 프로젝트 기획

이번 단계에서는 관통 프로젝트(PJT)의 전체 설계 방향을 청사진 수준으로 기획하는 것을 목표로 한다.

지금까지 학습한 기술 스택(Django, DB, HTML/CSS, API, AI 활용 등)을 종합하여 팀별로 아이디어를 구체화하고, 실제 서비스로 발전 가능한 형태로 정리한다.

향후 진행할 관통 프로젝트의 데모를 확인하고, 이를 기반으로 서비스 아이디어를 도출한 뒤, **팀별 기획서(docx)를 작성하여 제출**해야 한다.

본 단계의 핵심은 단순한 코드 구현이 아니라,

- "어떤 문제를 해결하는 서비스를 만들 것인가?"
- "우리 팀의 서비스가 어떤 AI 기능과 결합될 수 있을까?"

를 명확히 정의하는 것이다.

제공되는 팀별 기획서 양식을 요약하면 다음과 같다.

단계	수행 내용	비고
1	관통 PJT 데모를 확인하고, 전체 서비스 흐름을 이해한다.	데모 사이트 참고
2	팀별로 어떤 서비스를 개발할지 기획한다. 기존 관통 PJT 주제를 응용하거나 확장하여, 우리 팀만의 차별화된 기능을 제안한다.	신규 기능 제안 가능
3	서비스에 적용할 AI 기능을 구상한다. 예를 들어, 추천 시스템, 자동 분석, 요약, 생성형 응답 등 다양한 방향을 검토한다.	OpenAI / Gemini / HuggingFace / 공공데이터 활용 가능
4	관련된 데이터셋을 탐색한다. (전자정부 데이터, 공공 API, 크롤링 등 활용 가능)	데이터 수집 방식 기획
5	유사 서비스나 레퍼런스를 조사하여 벤치마킹한다.	URL, 주요 기능 요약 포함
6	모든 내용을 '팀별 기획서.docx' 파일에 정리하고 GitLab에 업로드한다.	파일명: 팀번호_기획서.docx