

PJT명	개인별 관심 종목 관리	
단계	05 PJT	
진행일자	2025.10.02	
예상 구현 시간	필수기능	6H
	심화기능	2H

## 1. 목표

- 직전 프로젝트(관심 종목 토론 데이터 수집)를 확장하여, **회원 기능** 추가한다.
  - 제공된 skeleton 코드: pjt04 정답 코드
  - Skeleton 코드를 활용하거나, 본인이 구현했던 코드를 활용합니다.
- Django 인증 시스템을 활용하여 회원가입, 로그인/로그아웃을 구현한다.
- 사용자별 관심 종목을 DB 에 저장하고 관리할 수 있도록 모델 및 페이지를 구성한다.
- 프로필 페이지에서 관심 종목 관리(추가, 삭제, 조회)를 할 수 있도록 구현한다.
- 관심 종목 클릭 시 기존 크롤링 기능을 연동하여 댓글 데이터를 보여준다.
- (도전) Bootstrap 과 생성형 AI 를 활용해 UI/UX 를 개선한다.

## 2. 준비사항

- 개발언어 및 툴
  - Python
  - Django
  - Bootstrap5
  - Visual Studio Code

### 3. 작업 순서

- 제공된 프로젝트(04\_pjt)를 기반으로 페어와 함께 분석한다.
  - 본인의 PJT04의 코드를 활용하여도 무관함
- 페어와 함께 개발 시 역할을 분담한다.
  - 팀장/네비게이터 등 역할 분담
- [선택] 추가적인 아이디어를 적용하여 구현하고 프로젝트를 완성한다.
- 코드 및 산출물을 정리하여 Git에 반영한다.

### 4. 요구사항

본 프로젝트는 직전 단계에서 구현한 **관심 종목 토론 데이터 수집 프로젝트**를 확장하여, 사용자별 **회원 기능과 관심 종목 관리 기능**을 추가하는 것을 목표로 한다. Django의 인증 시스템을 활용하여 회원가입, 로그인·로그아웃 기능을 구현하고, 각 사용자 계정에 관심 종목을 저장·조회·삭제할 수 있는 모델과 페이지를 설계한다.

이를 통해 학습자는 단순한 데이터 크롤링을 넘어, **사용자 인증 기반의 데이터 관리**라는 웹 서비스의 핵심 구조를 직접 경험하게 된다. 특히 프로필 페이지를 통해 관심 종목을 관리하고, 특정 종목을 클릭했을 때는 기존의 크롤링 기능과 연동하여 댓글 데이터를 불러오도록 구현함으로써 **실제 서비스와 유사한 데이터 흐름**을 설계할 수 있다.

필수 요구사항을 모두 구현한 이후에는 Bootstrap 및 생성형 AI를 활용하여 화면 디자인을 개선하고, 금융 상품 비교 애플리케이션에 어울리는 UI/UX를 직접 구상해 적용할 수 있다. 이를 통해 웹 애플리케이션 개발에서 요구되는 **인증·데이터 관리·UI 설계**에 이르는 전 과정을 실습하고, 실무와 가까운 프로젝트 경험을 쌓게 된다.

## ● 요구사항 명세

- 아래에 제시된 기능 명세에 맞는 기능을 구현한다.
- 필수 기능은 반드시 구현해야 하며, 심화 기능은 팀 내 기획에 따라 선택적으로 구현한다.

번호	분류	요구사항명	요구사항 상세	우선순위
기능적 요구사항				
F02	회원 관리	회원가입 구현	Django User 모델 기반 회원가입 (username, password1, password2) 가입 성공 시 자동 로그인	필수
F03	회원 관리	로그인/로그아웃	Django auth 기능 활용, 로그인/로그아웃 구현	필수
F04	관심 종목	관심 종목 저장 모델 구현	사용자별 관심 종목 저장, 동일 사용자 중복 저장 방지(UniqueConstraint), 생성일자 포함	필수
F05	관심 종목	프로필 페이지 - 관심 종목 관리	관심 종목 조회/추가/삭제 기능, 종목이 없을 시 메시지 출력	필수
F06	관심 종목	프로필 페이지 - 종목 클릭 시 크롤링	클릭 시 해당 종목 댓글 출력 (DB에 있으면 DB 사용, 없으면 크롤링 수행)	필수
F07	UI/UX	디자인 개선	Bootstrap + 생성형 AI를 활용한 UI/UX 개선	심화
...	...	...	...	...
비 기능적 요구사항				
NF01	문서화	README 작성	분석 흐름과 학습 내용, 어려웠던 점을 README.md에 기록하여 제출	필수
NF02	Git	환경변수 관리	gitignore 작성하여 환경 변수 관리	필수
...	...	...	...	...

## 1) 기본(필수) 기능

본 프로젝트는 **관심 종목 토론 데이터 수집 프로젝트를 확장**하여, 사용자별 관심 종목을 관리할 수 있는 기능을 구현하는 것을 목표로 한다. Django의 회원가입·로그인·로그아웃 기능을 기반으로 사용자를 구분하고, 각 사용자 계정에 관심 종목을 저장·조회·삭제할 수 있는 구조를 설계한다.

또한, 프로필 페이지를 통해 관심 종목을 관리할 수 있도록 하고, 특정 종목명을 클릭했을 때는 기존의 크롤링 기능과 연동하여 댓글 데이터를 확인할 수 있도록 한다. 이를 통해 단순한 데이터 출력에서 한 단계 나아가, **사용자 인증과 데이터 관리가 결합된 실제 서비스 구조**를 경험한다.

### A. 네비게이션 바 구현 (요구사항 F01)

- 상단에 아래 기능들을 포함한 네비게이션 바를 구현한다.
  - **비로그인 사용자**: 회원가입 페이지 이동, 로그인 페이지 이동
  - **로그인 사용자**: 로그아웃 기능, 프로필 페이지 이동
  - **공통 기능**: 주식 댓글 분석 페이지로 이동
- 결과물 예시
  - 비로그인 시 → 회원가입/로그인 메뉴 표시

주식 댓글 분석	회원가입	로그인
----------	------	-----

■

- 로그인 시 → 로그아웃/프로필 페이지 메뉴 표시

주식 댓글 분석	로그아웃	프로필
----------	------	-----

■

B. 회원가입 구현 (요구사항 F02)

- Django 기본 사용자 모델을 활용하여 회원가입 기능을 구현한다.
- 회원가입 폼에서 입력받는 필드:
  - Username
  - password1
  - password2
- 회원가입 성공 시 동작:
  - 자동 로그인 처리
  - 주식 댓글 분석 페이지로 이동
- 결과물 예시
  - 회원가입 완료 → 자동 로그인 후 메인 화면 전환

### 회원가입

Username:

Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.

Password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

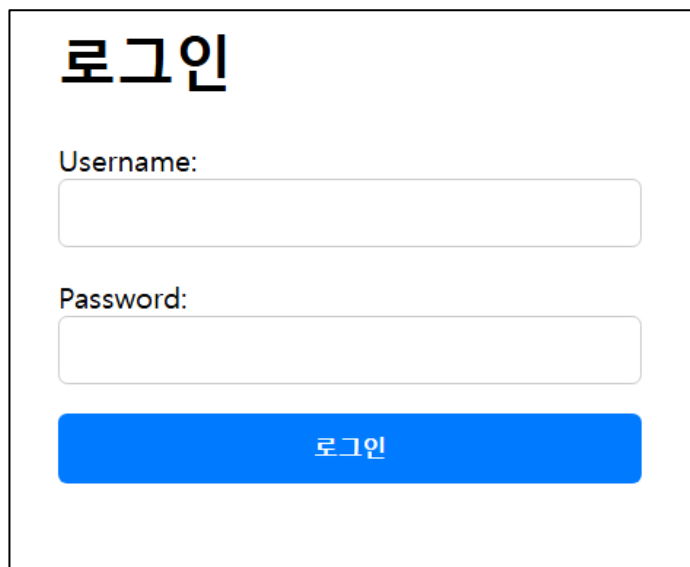
Password confirmation:

Enter the same password as before, for verification.

로그인

### C. 로그인 & 로그아웃 구현 (요구사항 F03)

- Django의 인증 시스템(django.contrib.auth)을 활용하여 로그인/로그아웃 기능을 구현한다.
- 로그인 페이지: 사용자 입력(username, password) → 인증 성공 시 메인 페이지 이동
- 로그아웃 기능: 네비게이션 바에서 로그아웃 버튼 클릭 시 즉시 로그아웃 처리
- 결과물 예시

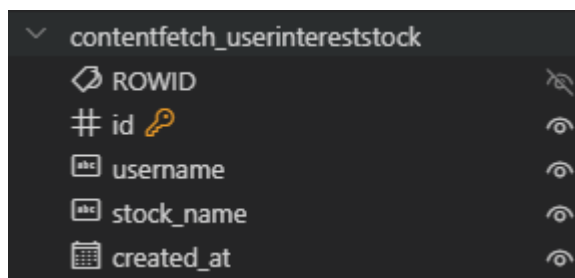


A login form UI mockup. At the top, the title '로그인' (Login) is displayed in a large, bold, black font. Below the title, there are two input fields. The first is labeled 'Username:' in a small, black font, followed by a white rectangular input box with a thin gray border. The second is labeled 'Password:' in a small, black font, followed by a white rectangular input box with a thin gray border. Below these two fields is a solid blue rectangular button with the text '로그인' (Login) in white, centered on the button.

- 
- 로그인 성공 → 사용자 이름 표시
- 로그아웃 성공 → 로그인/회원가입 버튼 표시

#### D. 관심 종목 저장 모델 구현 (요구사항 F04)

- 사용자별 관심 종목을 저장하기 위한 Django 모델을 설계한다.
- 저장 필드
  - 사용자 정보 (User 의 이름)
    - ✓ [검색 시 참고] 정석으로는 외래키를 써야 하지만, 이름으로 사용자를 구분함
  - 종목 이름 (문자열, 최대 255 자)
  - 생성일자 (자동 생성 DateTimeField)
- 제약 조건
  - 동일한 사용자가 같은 이름의 종목을 중복 저장할 수 없도록 UniqueConstraint 적용
- 결과물 예시



- 한 사용자가 "삼성전자"를 여러 번 등록 불가
  - ✓ [힌트] Meta 클래스의 UniqueConstraint 검색해보기

E. 프로필 페이지 – 관심 종목 관리 (요구사항 F05)

- 프로필 페이지에서 관심 종목을 추가·조회·삭제할 수 있는 기능을 구현한다.
- 관심 종목 추가
  - 입력폼을 통해 종목명 입력 후 저장
  - 실제 존재하지 않는 회사명도 우선 저장 가능
  - 결과물 예시

✓

주식 댓글 분석

로그아웃

프로필

test님의 관심 종목

종목명 입력

추가

- 관심 종목 조회
  - 저장된 관심 종목 리스트를 화면에 출력
  - 등록된 관심 종목이 없을 경우 메시지 출력 (예: “등록된 관심 종목이 없습니다.”)
- 관심 종목 삭제
  - 각 종목명 옆에 삭제 버튼 배치
  - 버튼 클릭 시 DB에서 해당 종목 삭제 후 화면 갱신
- 결과물 예시

■

test님의 관심 종목

종목명 입력

추가

삼성전자

삭제

삼성전기

삭제

멀티캠퍼스

삭제

싸피

삭제



F. 프로필 페이지 – 관심 종목 클릭 시 크롤링 (요구사항 F06)

- 프로필 페이지의 관심 종목 리스트에서 종목명을 클릭하면 댓글 분석 페이지로 이동한다.
- 데이터 출력 규칙
  - 해당 종목이 이미 크롤링되어 DB에 저장된 경우 → DB 데이터 출력
  - DB에 데이터가 없는 경우 → 크롤링 실행 후 데이터 저장 및 출력
- 크롤링 로직은 직전 프로젝트(04\_pjt)에서 구현한 기능을 재사용한다.
- 결과물 예시
  - “삼성전자” 클릭 시 → 커뮤니티 댓글 데이터 화면 표시

주식에 대한, 투자자들의 목소리

## 주식 댓글 분석기

회사 이름을 입력하세요:

 “삼성전자” 데이터 수집 및 분석 중...

토스 증권에서 댓글을 수집하고 'gpt-4o-mini' 모델을 이용하여 수집된 댓글을 분석하고 있습니다. 잠시만 기다려주세요.

### 주식 댓글 분석

주식에 대한, 투자자들의 목소리

## 주식 댓글 분석기

회사 이름을 입력하세요:

### 삼성전자 (005930)

수집된 댓글:

- 미국이야 뭔가 불확실성이 없어졌는데
- 이 나라는 더욱더 커지니.....
- 사실 산이유는 여친이 ds다니는데
- 맨날 야근하길래.. 고생하니깐 오르겠지 느낌으로 사봄
- 조정이라 어쩔수없음
- 56000 까지 올려있음..
- 사측생 각오로 임하랬더니 진짜...그래버리면

## 2) 심화 기능

본 심화 기능은 생성형 AI(OpenAI, Gemini 등)를 활용하여 금융 상품 관련 애플리케이션에서 자주 활용되는 UI/UX 구현을 목표로 한다. 회원가입, 로그인, 프로필 페이지, 관심 종목 관리 기능을 기반으로, AI를 통해 보다 직관적이고 사용자 친화적인 화면 설계 아이디어를 도출하고 이를 실제 페이지에 반영한다.

### A. 디자인 개선 (요구사항 F07)

- OpenAI, Gemini 등 생성형 AI에게 현재 디자인을 개선하거나, 금융 상품 비교 애플리케이션에 어울리는 프로필 페이지 요소들을 추가로 구성할 수 있도록 질문 프롬프트를 구성한다.

#### ■ 디자인 개선 프롬프트 예시

금융 상품 비교 애플리케이션을 구성하려고 해  
어떤 색감이 좋을까 ?

✓

#### ■ 프로필 페이지 요소 프롬프트 예시

금융 상품 비교 애플리케이션의 회원 프로필 페이지를 구현할 때 필수로 있어야 하는 구성 요소는 어떤 것들이 있을까 ?  
추후에 회원 정보를 토대로 상품을 추천할 거야

✓

## 5. 참고자료

- Django 공식 문서: <https://docs.djangoproject.com/>
- Django 인증 시스템 (django.contrib.auth)
- Selenium 공식 문서: <https://www.selenium.dev/documentation/>
- BeautifulSoup4 공식 문서: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
- Requests 공식 문서: <https://requests.readthedocs.io/en/latest/>
- Bootstrap 공식 문서: <https://getbootstrap.com/>
- OpenAI API 문서 (심화 기능용): <https://platform.openai.com/docs/>
- 웹 페이지 디자인 참고 사이트
  - 지디웹: <https://www.gdweb.co.kr/main/>
    - 좌측 WEB 부문 클릭

## 6. 결과

최종적으로 제출해야 할 항목은 아래와 같다

### 1. 구현 소스 코드

- Django 프로젝트 및 crawlings 앱 전체 소스코드

### 2. 실행 결과 캡처본

- 구현한 웹 페이지 캡처본

### 3. README.md

- 구현 기능 설명, 학습 내용, 느낀 점 포함

위 내용들을 모두 포함하여 GitLab에 업로드 한다.

GitLab 프로젝트 이름은 프로젝트 번호 + \_pjt 로 지정한다 (05\_pjt)