

PJT명	Django 웹 프레임워크와 생성형 AI를 활용한 도서 데이터 정보 관리	
단계	[Django 개발]	
진행일자	2025.09.26	
예상 구현 시간	필수기능	5H
	추가기능	2H
	심화기능	1H

1. 목표

- 데이터를 생성, 조회, 수정, 삭제할 수 있는 Web application을 제작한다.
- Django Model과 ORM에 대하여 이해한다.
- Bootstrap 컴포넌트 및 Grid system을 활용하여 반응형 레이아웃을 구성할 수 있다.

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

- 1) 팀원과 같이 요구사항을 확인하고, GitLab에 프로젝트를 생성한다.
 - 프로젝트 이름은 04-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

AI 기반 도서 분석과 창작 지원 통합 솔루션 서비스를 구축하려고 한다. 다양한 장르와 형태의 도서 데이터를 수집 및 관리하고, 이를 기반으로 AI 기반 도서 분석 기능을 설계 및 구현한다. 또한 작가들에게 아이디어 발상 지원, 플롯 구성 제안, 문장 교정 및 스타일 개선 등 창작 전반에 걸친 다양한 AI 기반 창작 지원 기능을 제공한다. 또한 도서에 대한 사용자 리뷰 및 감상평 공유 커뮤니티 기능을 제공하여, 사용자들이 활발하게 소통하고 정보를 교환할 수 있는 기능을 제공한다. 사용자는 자신이 읽은 도서를 평가하고, 다른 사용자의 리뷰를 참고하여 다음 도서를 선택하는데 도움을 받을 수 있다. 나아가, 관심 도서 목록을 맞춤형으로 구성하는 등 다양한 편의 기능을 제공한다. 팀원과 상의하여 아래 요구사항을 만족할 수 있도록 요구 사항 명세서를 작성 및 구현해보자.

서비스의 화면 및 데이터를 구성하는 단계이다. 도서 데이터 생성, 조회, 수정, 삭제가 가능한 웹 애플리케이션을 완성하자.

- 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	프로젝트	프로젝트 구성	도서 커뮤니티 서비스 구현을 위한 Django 프로젝트 및 앱 생성	필수
F02	프로젝트	Model Class	도서 데이터를 데이터베이스에 저장할 수 있도록 Django Model 클래스 구현	필수

F03	view 함수	index	전체 도서 데이터를 조회하는 view 함수 구현	필수
F04	view 함수	new	도서 데이터를 입력할 수 있는 UI를 제공하는 view 함수 구현	필수
F05	view 함수	create	새로운 도서 데이터를 전달받은 도서 데이터를 바탕으로 저장하는 view 함수 구현	필수
F06	view 함수	detail	단일 도서 데이터를 조회하는 view 함수 구현	필수
F07	view 함수	edit	수정 대상 데이터가 미리 작성되어 있는, 도서 데이터를 입력할 수 있는 UI를 제공하는 view 함수 구현	필수
F08	view 함수	update	선택된 도서 데이터를 전달받은 데이터를 바탕으로 수정하는 view 함수 구현	필수
F09	view 함수	delete	단일 도서 데이터를 삭제하는 view 함수 구현	필수
F10	AI 활용	정보 생성 및 저장	생성형 AI API를 활용하여 작가 정보 검색 및 생성	도전
비기능적 요구사항				
NF01	유지보수	URL 구성	URL이 변경될 때 유지보수가 용이하도록 구현	필수
...

1) 프로젝트 및 앱 (필수)

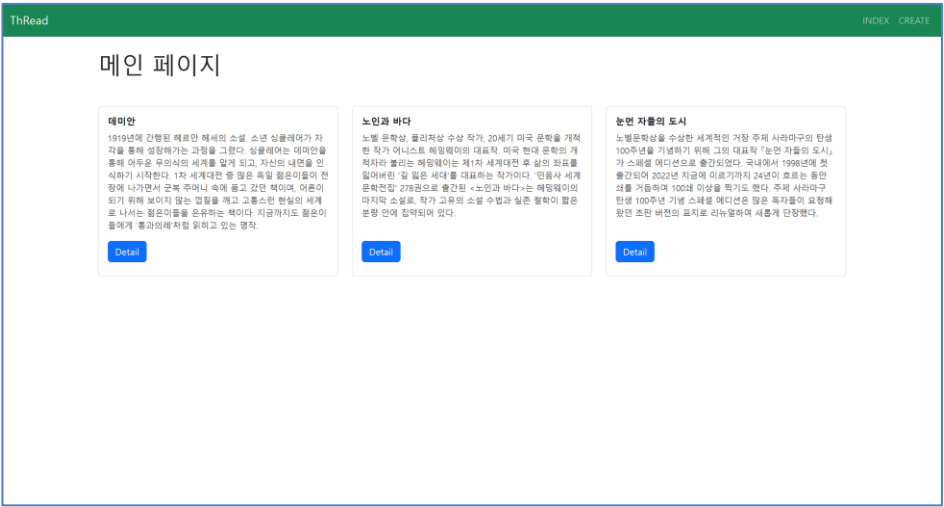
도서 데이터를 생성, 조회, 수정, 삭제할 수 있는 Django 프로젝트를 만든다. 명시된 요구사항 이외 서비스를 위해 필요하다고 생각되는 기능 등은 자유롭게 구현해도 무관하며, Bootstrap을 활용하여 자유롭게 스타일링 한다.

- 요구사항 번호: F01, F02, NF01
- 프로젝트 이름: mypjt
- 앱 이름: books
- Model 클래스 이름: Book
 - 도서 제목, 설명, 회원 리뷰 평점, 작가를 저장할 필드 4개 지정
- 각 세부 기능들을 위한 view 함수를 구성
 - index, create, detail, update, delete
 - URL이 변경될 가능성을 고려하여 유지보수에 유리한 형태로 구성
- 페이지간 이동 편의성을 갖추기 위한 Navbar 구성
 - books의 index, create 기능들에 접근하기 편하도록 링크를 제공
 - 다양한 페이지에서 도달할 수 있도록 여러 페이지에 동일하게 출력되도록 구현

A. index

전체 도서 데이터를 조회하기 위한 index view 함수를 구현한다.

- 요구사항 번호: F03
- 전체 도서 데이터가 조회되어야 함
- 특정 도서를 선택하여 상세 페이지로 이동할 수 있는 링크가 제공되어야 함
 - 링크의 문구와 UI는 자유롭게 구성
- 출력 예시



B. create

도서 데이터를 입력하기 위한 UI를 제공하는 new view 함수, 입력된 데이터를 저장하는 create view 함수를 구현한다.

- 요구사항 번호: F04, F05
- Book 모델의 데이터를 입력할 수 있는 UI 구현
 - 유효한 데이터의 경우 생성 후 상세 페이지로 리다이렉트
 - UI 디자인은 자유롭게 구성
- 출력 예시

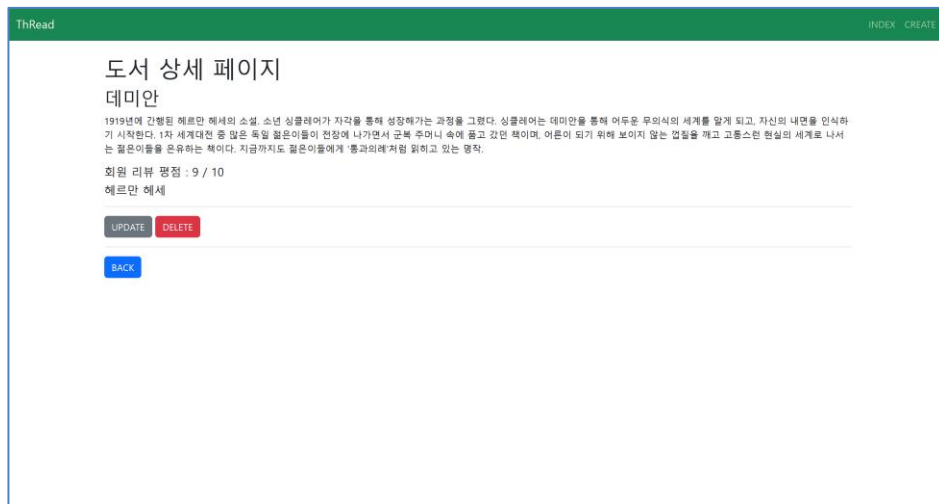


The screenshot displays a web browser window with a green header bar. On the left of the header is the text 'ThRead' and on the right are the links 'INDEX' and 'CREATE'. The main content area is titled '도서 등록 페이지' (Book Registration Page). It contains a form with the following elements: a 'Title:' label followed by a text input field; a 'Description:' label followed by a large text area; a 'Customer review rank:' label followed by a small text input field; and an 'Author:' label followed by a text input field. At the bottom left of the form is a button labeled '저장' (Save).

C. detail

단일 도서 데이터를 조회하는 detail view 함수를 구현한다.

- 요구사항 번호: F06
- index 페이지에서 도서를 선택했을 때 이동됨
 - 선택한 해당 도서의 정보를 바탕으로 내용이 출력되어야 함
- 도서 제목, 설명, 회원 리뷰 평점, 작가 데이터를 출력
 - 도서를 수정하기 위한 페이지로 이동하는 링크 구현
 - 도서를 삭제하는 버튼 구현
 - 이전 페이지(index)로 이동하는 링크 구현
 - UI 디자인은 자유롭게 구성
- 출력 예시



D. update

도서 데이터를 수정하기 위한 UI를 제공하는 edit view 함수, 입력된 데이터를 바탕으로 대상 도서 데이터를 수정하는 update view 함수를 구현한다.

- 요구사항 번호: F07, F08
- detail 페이지에서 수정하기 링크를 통해 이동됨
 - detail 페이지에서 제공되었던 도서 정보를 바탕으로 기능이 동작되도록 구현
- Book 모델의 데이터를 입력할 수 있는 UI 구현
 - 이전 도서의 데이터가 미리 입력되어 있도록 구현
 - 유효한 데이터의 경우 저장 후 상세 페이지로 리다이렉트
 - UI 디자인은 자유롭게 구성
- 출력 예시

The screenshot shows a web application interface for editing a book. The header is green with 'ThRead' on the left and 'INDEX CREATE' on the right. The main content area is white and titled '도서 수정 페이지'. It contains a form with the following elements:

- Title:** A text input field containing '데미안'.
- Description:** A text area containing two paragraphs of Korean text. The first paragraph describes Hermann Hesse's 'Demian' as a novel about a young man's journey to self-discovery. The second paragraph mentions the book's popularity and its status as a classic.
- Customer review rank:** A text input field containing the number '9'.
- Author:** A text input field containing '헤르만 헤세'.
- Save:** A button labeled '저장' (Save).

E. delete

도서 데이터를 삭제하는 기능을 제공하는 delete view 함수를 구현한다.

- 요구사항 번호: F09
- detail 페이지에서 삭제하기 버튼을 통해 호출됨
 - detail 페이지에서 제공되었던 도서 정보를 바탕으로 기능이 동작되도록 구현
- 삭제가 된 이후에는 index 페이지로 리다이렉트

2) 도전 과제

생성형 AI 도구를 활용하여 도전과제 요구사항을 해결해보자.

- 코드 생성, 아이디어 구상, 문제 해결 방법 탐색 등 다양한 방식으로 활용할 수 있다.
- 사용할 생성형 AI 서비스는 자유롭게 선택한다.
 - 제공되는 GPT API Key를 활용할 경우, 모델은 반드시 gpt-5-nano 모델을 사용한다.
- 최종 결과물은 AI 생성 내용을 바탕으로 직접 수정 및 개선하여 적용해본다.

보조 수단으로 활용하되, 능동적인 자세로 학습에 임할 것.

- 최종적인 이해와 적용은 자기 주도적 학습을 통해 이루어지며, 배운 내용을 스스로 기록하고 정리하며 학습 효과를 높일 것.

A. 생성형 AI API를 활용한 작가 정보 생성 및 저장

생성형 AI API를 활용하여 작가 정보를 생성하고 저장해보자.

- 요구사항 번호: F10
- 도서를 등록할 때 입력 받은 작가 이름을 위키피디아 API의 쿼리로 전달해 작가 정보, 프로필 사진 검색
- 사용자 입력 데이터와 위키피디아 검색 정보를 바탕으로, GPT API를 활용하기
 - 작가 정보와 대표 작품 목록 요청
 - JSON 형태로 응답하도록 프롬프트 작성
- Book 모델에 필드를 추가하여 저장
 - 작가 프로필 사진: author_img_url
 - 작가 정보: author_info
 - 대표 작품 목록: author_works
 - 프로필 사진의 경우 이미지 링크를 저장
- 추가한 필드를 도서 상세 페이지에 출력
 - 작가 정보 영역 아래에 "작가 대표작과 작가 정보는 AI가 생성한 정보로, 정확하지 않을 수 있습니다."라는 안내 문구 추가
- 출력 예시



5. 참고자료

- Django 5.2
<https://www.djangoproject.com/start/overview/>
- Python OpenAI Library
<https://platform.openai.com/docs/libraries?language=python>
<https://github.com/openai/openai-python>

6. 결과

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

- 산출물과 제출
 - 단계별로 구현 과정 중 학습한 내용, 어려웠던 부분, 새로 배운 것들 및 느낀 점을 상세히 기록한 README.md
 - 완성된 각 문제 별 소스코드 및 실행 화면 캡처본
 - 프로젝트 이름은 04_pjt로 지정, 각자의 계정에 생성할 것
 - 각 반 담당 강사님을 Maintainer로 설정

- 끝 -