

06 PJT

회원제 커뮤니티 게시판 구현

챕터의 포인트

- [도전] 회원제 커뮤니티 게시판 구현
- 제출

회원제 게시판 구현하기

| 공통 요구사항 (1/2)

- 추가 구현할 회원 관리 앱의 이름은 **accounts** 로 지정합니다.
- **.gitignore** 파일을 추가하여 불필요한 파일 및 폴더는 제출하지 않도록 합니다
- 명시된 요구사항 이외에는 자유롭게 작성해도 무관합니다.


| 공통 요구사항 (2/2)

- 커뮤니티 웹 서비스에 회원 기능을 추가합니다.
 - 게시글 생성, 조회, 수정, 삭제 및 댓글 생성, 삭제 기능을 제공합니다.
 - 로그인, 로그아웃, 회원가입이 가능한 애플리케이션을 완성합니다.
 - 로그인 여부에 따라 다른 동작이 가능한 애플리케이션을 완성합니다.
- 메서드 구현 시 `@require_http_methods` 를 활용하여 필요한 HTTP 메서드만 허용하도록 구현합니다.
- **제공된 코드**와 프로젝트 예시를 참고하여, 직접 회원 애플리케이션을 설계하고 완성합니다.

| Model

- 수정할 모델 클래스 목록
 - A. Board
 - B. Comment
- 새로 정의할 모델 클래스 목록
 - C. User

A. Board

- Board 모델에 다음과 같은 정보를 추가합니다.
-  표시된 부분을 추가하고 수정한 내용을 데이터베이스에 반영합니다.


필드명	데이터 유형	역할
author	integer	외래 키(User 클래스 참조)
title	varchar(100)	게시글 제목
content	text	게시글 내용
created_at	datetime	게시글 생성일
updated_at	datetime	게시글 수정일

B. Comment

- Comment 모델에 다음과 같은 정보를 추가합니다.
-  표시된 부분을 추가하고 수정한 내용을 데이터베이스에 반영합니다.

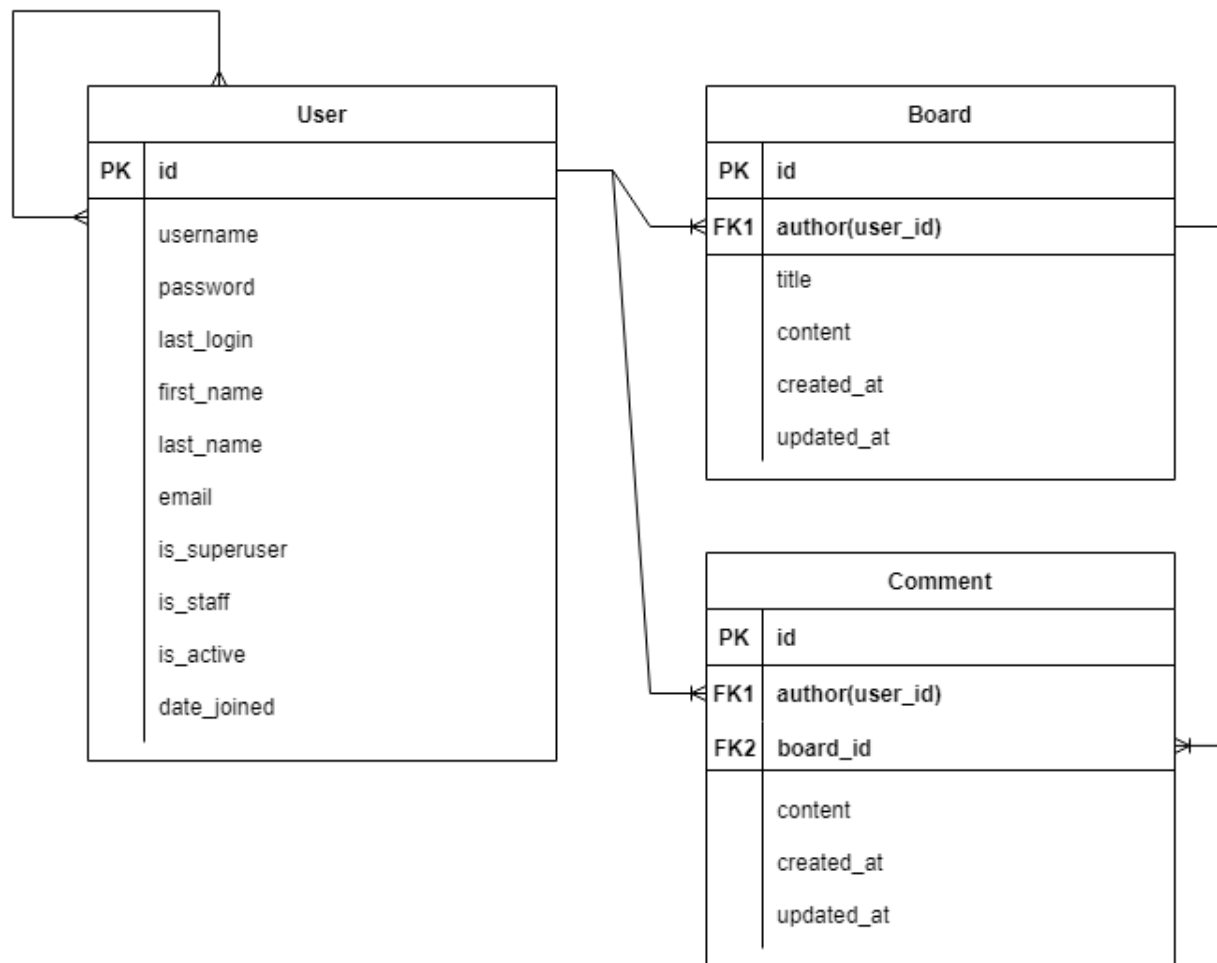
필드명	데이터 유형	역할
author	integer	외래 키(User 클래스 참조)
board	integer	외래 키(Board 클래스 참조)
content	text	댓글 내용
created_at	datetime	댓글 생성일
updated_at	datetime	댓글 수정일

| C. User

- Django 에서 제공하는 AbstractUser 를 상속받는 User 클래스를 추가합니다.
- 수정할 모델 클래스 이름은 User 이며, 아래 필드를 추가합니다.
 -  표시된 부분을 추가하고 수정한 내용을 데이터베이스에 반영합니다.

필드명	역할
followings	M:N 중개 테이블 생성 (User 클래스 비대칭 참조)
나머지 필드들은 기존에 있던 필드들을 그대로 활용합니다.	

ERD (Entity-Relationship Diagram)



| URL (1/2)

- boards 앱은 다음 URL 요청에 맞는 역할을 가집니다.

URL 패턴	역할
/boards/	전체 게시글 목록 페이지 조회
/boards/create/	게시글 생성 페이지 조회 & 게시글 작성 및 저장
/boards/<board_pk>/	단일 게시글 상세 페이지 조회 & 댓글 작성 및 저장
/boards/<board_pk>/update/	단일 게시글 수정 페이지 조회 & 게시글 수정 및 저장
/boards/<board_pk>/comment/	댓글 저장
/boards/<board_pk>/comment/<comment_pk>/	댓글 삭제

| URL (2/2)

- accounts 앱은 다음 URL 요청에 맞는 역할을 가집니다.

URL 패턴	역할
/accounts/login/	로그인 페이지 조회 & 세션 데이터 생성 및 저장(로그인)
/accounts/logout/	세션 데이터 삭제(로그아웃)
/accounts/signup/	회원 생성 페이지 조회 & 회원 데이터 작성 및 저장(회원가입)
/accounts/<user_pk>/follow/	팔로우 데이터 저장 및 팔로우 데이터 삭제

| View (1/2)

- boards 앱은 다음 역할을 가지는 view 함수를 가집니다.

View Method	역할	허용 HTTP Method
index	전체 게시글 조회 및 index.html 렌더링	GET
create	create.html 렌더링 유효성 검증 및 게시글 저장 후 detail.html 리다이렉트	GET, POST
detail	게시글 상세 데이터 조회 및 detail.html 렌더링 게시글 삭제	GET, POST
update	수정 대상 게시글 조회 및 update.html 렌더링 유효성 검증 및 게시글 수정 후 detail.html 리다이렉트	GET, POST
comment	유효성 검증 및 댓글 저장 후 detail.html 리다이렉트	POST
comment_detail	댓글 삭제 후 detail.html 리다이렉트	POST

| View (2/2)

- accounts 앱은 다음 역할을 가지는 view 함수를 가집니다.

View Method	역할	허용 HTTP Method
login	login.html 렌더링 및 회원 로그인	GET, POST
logout	DB 와 클라이언트 쿠키에서 세션 데이터 삭제	POST
signup	signup.html 렌더링 유효성 검증 및 회원 데이터 저장 후 index.html 리다이렉트	GET, POST
follow	회원 팔로우 데이터 저장 및 팔로우 데이터 삭제	POST

| Admin

- 모델 Board, Comment, User를 Admin site에 등록합니다.
- Admin site에서 데이터의 생성, 조회, 수정, 삭제가 가능해야 합니다.

| Form

- Board 모델과 Comment 모델의 데이터 검증, 저장, 에러 메시지, HTML을 모두 관리하기 위해 적절한 **ModelForm**을 사용합니다.
- User 모델의 데이터 검증, 저장, 에러 메시지, HTML을 모두 관리하기 위해 적절한 Form 과 **ModelForm** 그리고 커스텀 **ModelForm**을 사용합니다.

| 세부 요구사항 - (1/2)

A. 회원 관리 기능

1. 회원 가입
2. 로그인 & 로그아웃

B. Boards 앱 기능

1. 게시글 CRUD(생성, 조회, 수정, 삭제) 기능
2. 댓글 생성, 삭제 기능

| 세부 요구사항 - (2/2)

C. 프로필 페이지

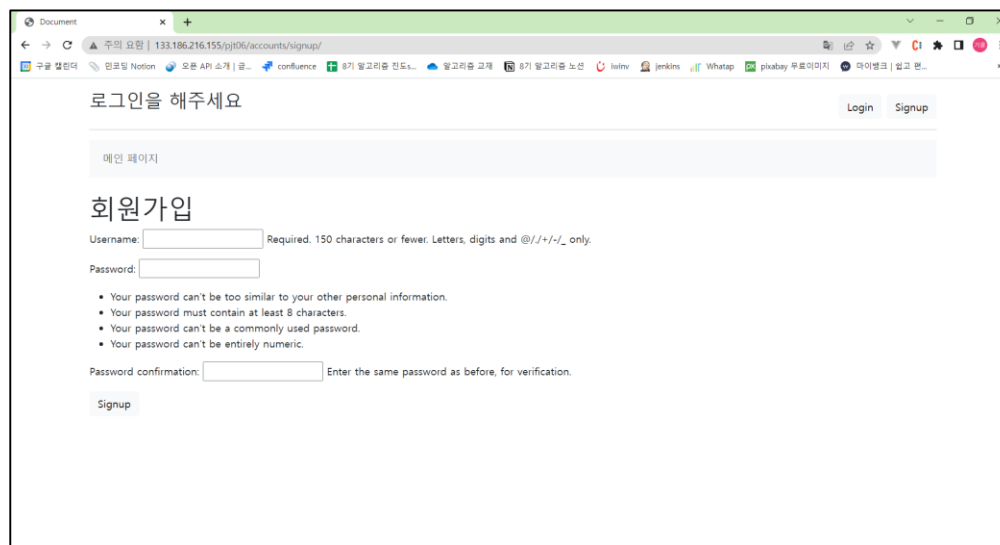
D. 유저 팔로우 기능

E. 게시물 좋아요 기능 (선택 사항)

A. 유저 기능

1. 회원 가입

- Django의 UserCreationForm 을 적절히 수정하여 커스텀 ModelForm을 구현합니다.
- 가입 정보를 입력할 수 있는 적절한 UI를 제공합니다.
- 회원 가입 버튼 클릭 시 입력한 회원 정보로 DB에 저장될 수 있도록 구현합니다.

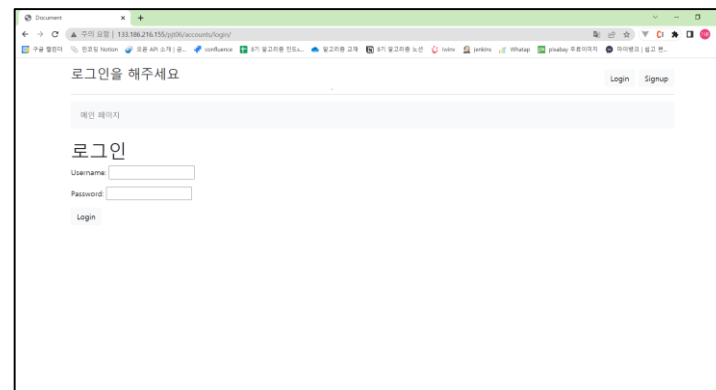


The screenshot shows a web browser window with a URL bar indicating a local development environment. The page title is "로그인을 해주세요" (Please log in). Below the title, there are "Login" and "Signup" buttons. A link for "메인 페이지" (Main page) is also visible. The "회원가입" (Sign up) section contains a "Username:" label, a text input field, and a note: "Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only." Below this is a "Password:" label, a text input field, and a list of password requirements: "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.", and "Your password can't be entirely numeric." There is also a "Password confirmation:" label, a text input field, and a note: "Enter the same password as before, for verification." At the bottom of the form is a "Signup" button.

A. 유저 기능

2. 로그인 & 로그아웃

- 비로그인 시 로그인 버튼을, 로그인 된 사용자에게는 로그아웃 버튼을 제공합니다.
- 로그인 버튼 클릭 시 로그인을 할 수 있는 페이지를 제공합니다.
 - Django가 제공하는 로그인 Form 과 메서드를 활용합니다.
- 로그아웃 버튼 클릭 시 DB와 클라이언트 쿠키에서 세션 데이터를 삭제할 수 있도록 구현합니다.
 - Django 가 제공하는 로그아웃 메서드를 활용합니다.



| B. Boards 앱 기능

1. 게시글 CRUD(생성, 조회, 수정, 삭제) 기능

- 생성, 조회 기능
 - 사용자의 인증 여부와 관계 없이 전체 영화 및 단일 영화 조회 페이지에서 적절한 UI를 활용하여 영화 목록을 제공합니다.
- 수정, 삭제 기능
 - **본인이 작성한 게시글만** 수정 및 삭제가 가능하도록 UI와 로직을 구현합니다.
 - 수정 페이지에서 내용 작성 및 수정이 가능하도록 적절한 UI를 제공합니다.

| B. Boards 앱 기능

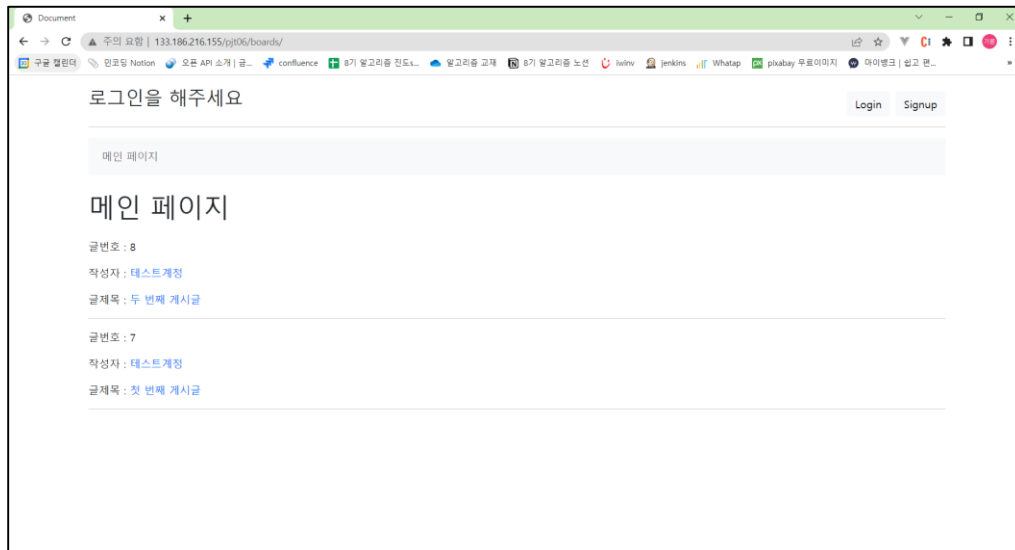
2. 댓글 생성, 삭제 기능

- 생성 기능
 - 인증된 사용자에게만 댓글 작성이 가능하도록 UI와 로직을 구현합니다.
- 삭제 기능
 - 본인이 작성한 댓글만 삭제가 가능하도록 UI와 로직을 구현합니다.

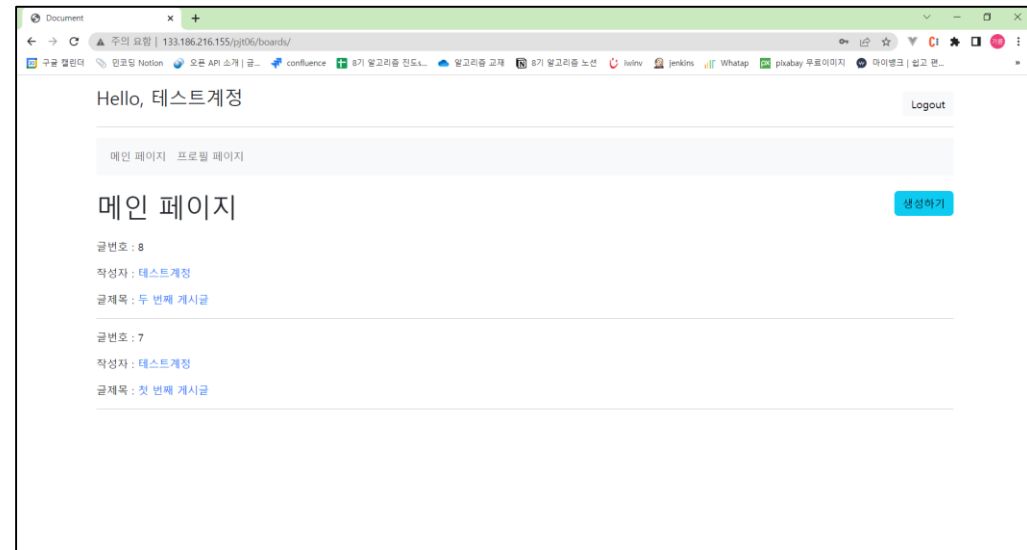
B. Boards 앱 기능 예시 화면

- 전체 게시글 조회 페이지

로그인 전



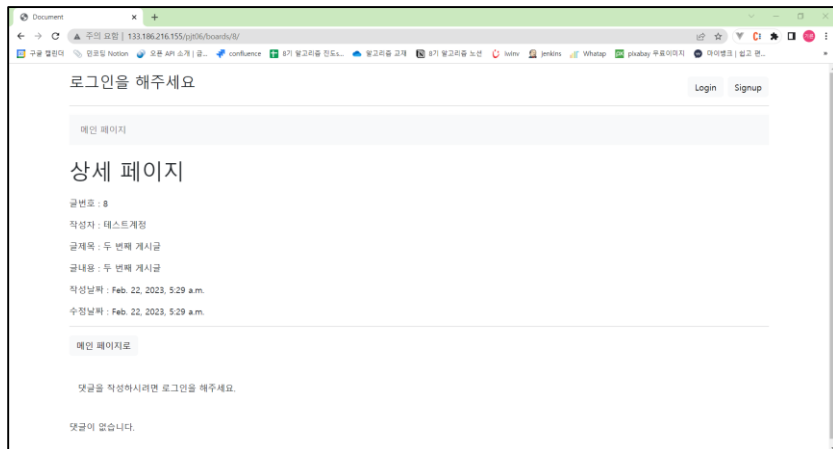
로그인 후



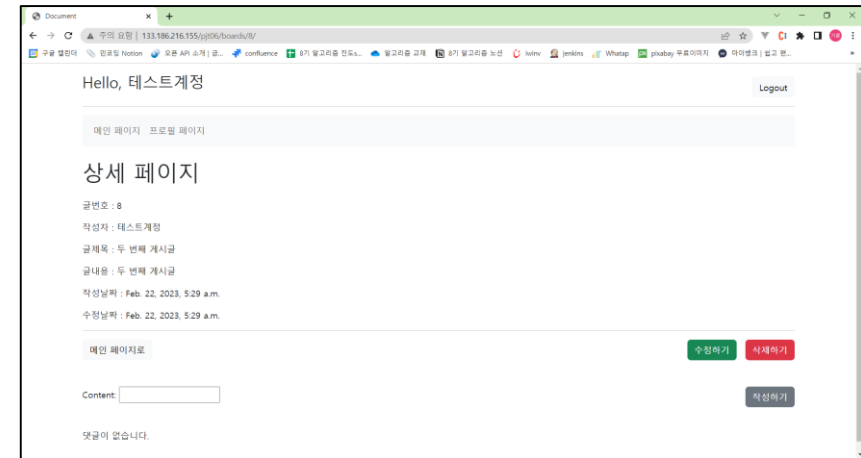
B. Boards 앱 기능 예시 화면

- 상세 게시물 조회 페이지
 - 내가 작성한 댓글만 삭제 가능하도록 구현합니다.

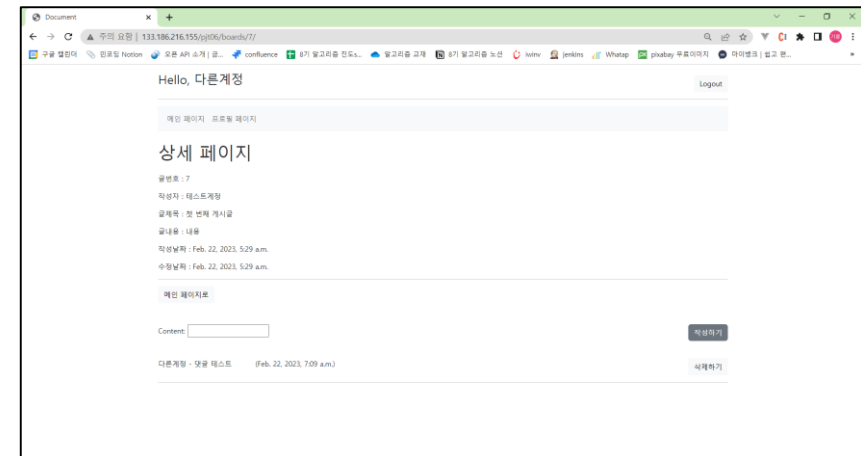
로그인 전



로그인 후 - 내가 작성한 게시물



로그인 후 - 다른 사람이 작성한 게시물



B. Boards 앱 기능 예시 화면

- 게시글 생성 페이지

Document x +

← → ↻ 주의 요함 | 133.186.216.155/pjt06/boards/create/ 🔍 📄 ☆ ▼ C i 📁 기물 ⋮

구글 캘린더 민코딩 Notion 오픈 API 소개 | 금... confluence 8기 알고리즘 진도s... 알고리즘 교재 8기 알고리즘 노션 kwinv jenkins Whatap »

Hello, 다른계정 Logout

메인 페이지 프로필 페이지

생성 페이지

Title:

Content:

생성하기

B. Boards 앱 기능 예시 화면

- 수정 페이지

Document x +

주의 요함 | 133.186.216.155/pjt06/boards/8/update/

구글 캘린더, 링크딩 Notion, 오픈 API 소개 | 글..., confluence, 8기 알고리즘 진도s..., 알고리즘 교재, 8기 알고리즘 노션, lwinv, jenkins, Whatap, pixabay 무료이미지, 마이뱅크 | 업고 현...

Hello, 테스트계정 Logout

메인 페이지 프로필 페이지

수정 페이지

Title: 두 번째 게시물

두 번째 게시물

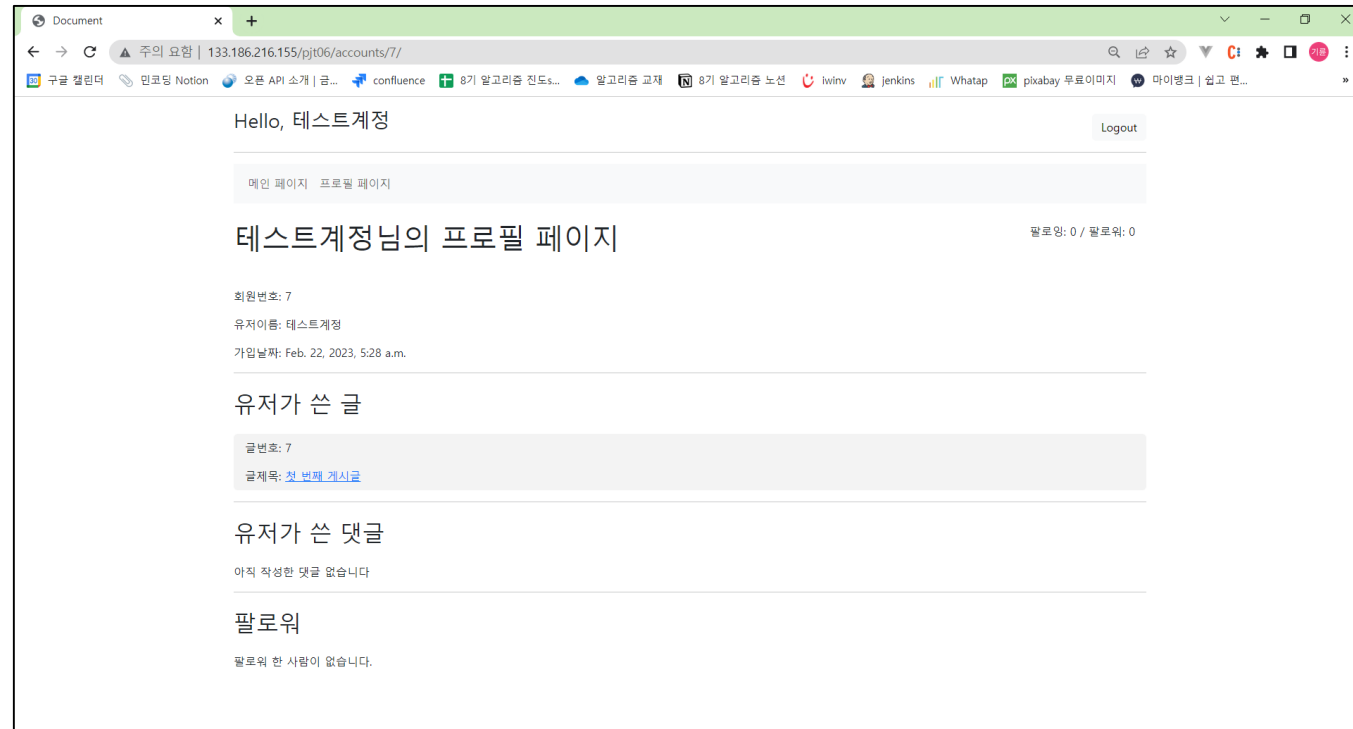
Content:

수정하기

| C. 프로필 페이지

- 기본적인 회원 정보(회원번호, 회원ID, 가입날짜 등)를 출력합니다.
- 적절한 UI를 활용하여 해당 회원이 작성한 게시글 목록을 제공합니다.
 - 각 게시글의 글 번호, 글제목을 출력합니다.
 - 게시글 제목 클릭 시 해당 게시글 상세 페이지로 이동합니다.
- 적절한 UI를 활용하여 해당 회원이 작성한 댓글 목록을 제공합니다.
 - 각 댓글이 작성된 게시글 제목, 댓글 내용을 출력합니다.
 - 게시글 제목 클릭 시 해당 게시글 상세 페이지로 이동합니다.

C. 프로필 페이지 예시 화면

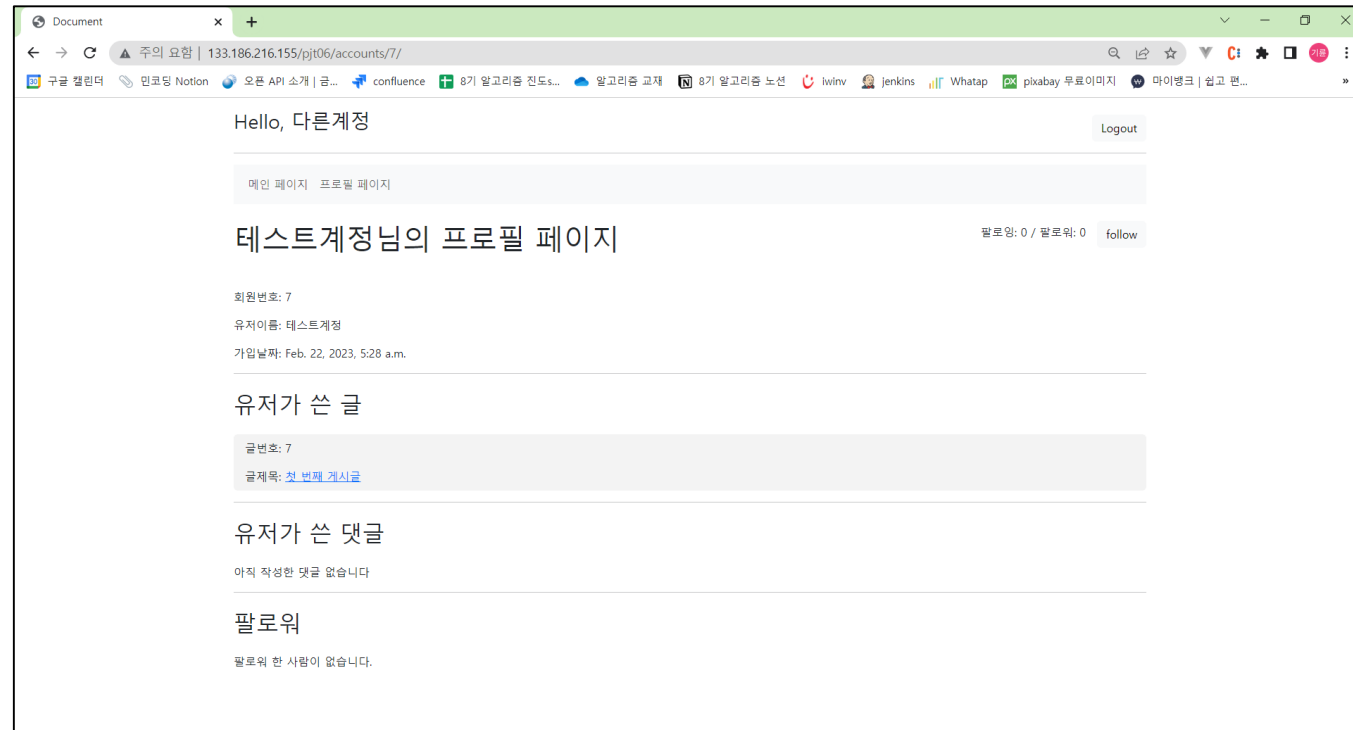


| D. 유저 팔로우 기능

- 프로필 페이지에 팔로워 수와 팔로잉 수를 표시합니다.
- 프로필 페이지에 해당 사용자를 팔로우 할 수 있는 버튼을 표시합니다.
- 인증된 사용자만 다른 사용자를 팔로우 할 수 있으며, 사용자는 자기 자신을 팔로우 할 수 없습니다.
- 팔로우 버튼을 클릭하는 경우, DB에 팔로우 정보를 저장한 후 리다이렉션을 활용하여 해당 프로필 페이지에서 변경된 팔로우 수를 확인할 수 있도록 구현합니다.

D. 유저 팔로우 기능 예시 화면

- 다른 사람의 프로필 페이지에 접근 시 팔로우를 할 수 있도록 구현합니다.



| E. 게시물 좋아요 기능 (선택 사항)

- 단일 상세 게시물 조회 페이지에 좋아요 버튼과 좋아요 개수를 표시합니다.
- 이미 좋아요 버튼을 누른 경우 좋아요 취소 버튼을 표시합니다.
- 인증된 사용자만 좋아요 기능을 사용할 수 있습니다.
- 좋아요 버튼을 클릭하는 경우 리다이렉션을 이용하여 새로고침 후 변경된 좋아요 개수를 확인할 수 있도록 구현합니다.
- 프로필 페이지에서 좋아요를 누른 게시물 목록을 확인할 수 있도록 구현합니다.

제출

| 제출 시 주의사항

- 제출기한은 금일 18시까지입니다. 제출기한을 지켜 주시기 바랍니다.
- 반드시 README.md 파일에 단계별로 구현 과정 중 학습한 내용, 어려웠던 부분, 새로 배운 것들 및 느낀 점 등을 상세히 기록하여 제출합니다.
 - 단순히 완성된 코드만을 나열하지 않습니다.
- 위에 명시된 요구사항은 최소 조건이며, 추가 개발을 자유롭게 진행할 수 있습니다.
- <https://lab.ssafy.com/> 에 프로젝트를 생성하고 제출합니다.
 - 프로젝트 이름은 '프로젝트 번호 + pjt' 로 지정합니다. (ex. 01_pjt)
- 반드시 각 반 담당 교수님을 Maintainer 로 설정해야 합니다.