Progress2 (~5/8)

1. 이번 주에 구현한 behavior의 목록

이전 계획서에 명시한 바와 같이 글 쓰기, 댓글 남기기 그리고 좋아요를 구현하였다. 구현을 백엔드 프론트엔드로 나누지않고, 기능별로 나누어, 프로젝트 진행이 더 수월하게 이루어질 수 있었다. 또한, 테스트 스크립트의 경우 본인이 구현한 기능에 한에 프론트엔드와 백엔드 둘다 작성하였다. 서로가 구현한 코드에 대해 구체적인 리뷰와 개선점을 남겨주었고, 피드백 또한 저번주에 비해서는 비교적 즉각적으로 이루어졌다.

이한결: **좋아요 기능** 구현(백엔드, 프론트엔드) 및 해당 테스트 스크립트 작성

로그인 프론트엔드 **폼 validation** 구현

배상환**: 글 쓰기/읽기**(백엔드) 구현 및 **백엔드 테스트 스크립트** 작성

**코드 대폭 정리** – 프론트엔드/ 백엔드 / 테스트 스크립트

김명곤: **댓글 남기기** 기능 구현(백엔드, 프론트엔드) 및 해당 테스트 스크립트 작성

로그인 프론트엔드 **폼 validation 테스트 스크립트** 작성

류호: **글 쓰기/읽기**(백엔드, 프론트엔드) 및 **프론트엔드 테스트 스크립트** 작성

2. backend 새로 구현한 api 스펙 및 테스트 코드

2.1. api 스펙

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| url | GET | POST | DELETE | 비고 |
| posts/ | 인증된 사용자에게만 전체 글 목록을 가져오도록 허용. | 글 생성 | - | - |
| users/exists/ | 중복된 유저로 가입하는 것을 방지하기 위함. 유저가 있으면 200반환, 없으면 204 반환. | - | - | - |
| comments/  / | 인증된 사용자에게만 전체 댓글 목록을 가져오도록 허용. | 댓글 생성 | - | - |
| posts/id/comments/ | 인증된 사용자에게만 허용, id에 해당하는 글의 댓글들을 반환 | 특정 id의 글에 댓글 생성 |  |  |
| posts/like | - | 특정 글에 좋아요 상태 업데이트 |  |  |

2.2. 테스트코드

실행은 $ python backendTest.py

테스트 코드는 회원가입, 로그인, 글 읽기/쓰기, 좋아요 그리고 댓글 쓰기로 이루어져 있다.

회원가입 성공 > 중복된 회원이름으로 가입 요청 실패

> 로그인 성공 > 로그인 실패

> 권한 없는 유저로 지우기 요청 실패

> 글쓰기 > 권한 없는 유저로 글쓰기 요청 실패

> 글읽기 > 권한 없는 유저로 글읽기 실패

> admin 권한으로 유저 지우기

> 좋아요 (+1) 요청 성공 > 좋아요 (-1) 요청 성공 > 없는 글에 대한 좋아요 요청 실패

> 댓글 쓰기 > 댓글 읽기

3. frontend 테스트코드

**회원가입, 로그인**

초기화면 접속 > 회원가입페이지로 이동 > 회원가입 > 초기화면 이동 확인 > 로그인페이지 이동 > 회원가입페이지 이동 > 회원가입 페이지 확인 > 로그인 페이지에서 로그인 > 성공 확인

으로 구성되어 있다.

**좋아요**

로그인 > 첫번째 포스트에 좋아요 클릭 > 좋아요 수 비교 > 성공 확인

**댓글**

로그인 > 포스트에 댓글 달기 > 성공 확인

**글쓰기**

로그인 > 빈 포스트 쓰기 > 성공 확인 > 내용 있는 포스트 쓰기 > 성공 확인

실행은

'$ python3 signup\_login\_test.py'

'$ python3 like\_test.py'

'$ python3 comment\_test.py'

'$ python3 write\_read\_test.py'

4. aws 링크

Frontend: <http://34.208.93.214:3000/>

Backend: <http://34.208.93.214:8000/>