

[SWPP] progress report 05.08. (TEAM01)

김다윤, 배새봄, 안단태, 유가온

1. 이번 주에 구현한 behaviors

이번주는 글쓰기(article) 및 댓글(article_article), 좋아요(like) 기능에 대해서 구현하였습니다.

여러 behavior들이 구현에 있어서 밀접한 관련을 가지고 있기 때문에 behavior들을 기준으로 역할을 나누기 힘들었습니다. 따라서 백엔드, 프론트엔드, 디자인, 파이프라인, 테스트코드 단위로 나누어 각자 작업하게 되었습니다.

김다윤

- CSS 디자인 : 로그인 페이지, 메인 페이지 및 주요 페이지들의 디자인을 깔끔하게 보이도록 설정했습니다. 조원들의 피드백을 받아 디자인을 보완할 생각입니다.
- 백엔드 및 프론트엔드 감수 : 여러 피드백과 테스트를 도왔습니다. 풀 리퀘스트 후 머지 과정에서 생길 수 있는 문제를 잡아주었습니다.
- 프론트엔드 스펙 정리 : 모델을 수정 및 정리하고 거기에 필요한 자료들과 코멘트를 남겼습니다.

배새봄

- 프론트엔드 설계 : 프론트엔드 모델을 구체적으로 설계하여 다른 팀원들과의 토의를 거쳐서 선정되었습니다
- 프론트엔드 구현 : 메인 페이지에서 글쓰기, 댓글 달기, 좋아요 기능을 구현하였고 URL이 다른 여러 페이지 사이에서 직관적으로 이동할 수 있게 구현하였습니다.
- 백엔드 피드백 : 필요한 메소드, URL 정보를 잘 전달해주었고 후에 부족한 부분들을 함께 토의하였습니다.

안단태

- 백엔드 설계 : 백엔드 모델을 구체적으로 설계하여 다른 팀원들과의 토의를 거쳐서 선정되었습니다.
- 파이프라인 및 테스트 코드 : Bitbucket 백엔드 파이프라인을 완성하였고, 이번 주에 구현한

코드를 테스트하는 새로운 백엔드 테스트 코드를 작성하였습니다.

- 동기부여 역할 & 검토 : 동기부여 역할을 맡아서 팀의 원활한 작업을 훌륭하게 도왔습니다. 테스트 코드뿐만 아니라 각종 검토에도 참여해주었습니다.

유가온

- 서버 관리자 : 포트 2개를 더 열고, 영구 포트를 개방 중입니다.
- 이슈 관리자 : 오래된 이슈를 닫고, 새로운 이슈를 많이 열었습니다.
- 백엔드 구현 : 토의를 거쳐 결정된 모델을 기반으로 글, 댓글, 댓댓글, 좋아요 기능을 구현하였습니다. 그리고 편의를 위해서 백엔드에 추가 URL을 만들었습니다. 차후에는 좋아요 기능을 강화할 예정입니다.

2. backend에 새로 구현/수정한 API 및 테스트코드

(Serializers.py) like serializer, article serializer 구현

(views.py) article, article_article(댓글), like view, main view 작성, 각종 권한 설정

(urls.py) article, article/article, article/like, mainpage, like URL 작성

(backend testcode) 보고서 4번 항목 참고

백엔드에서 글 목록/디테일을 GET하거나 글을 새로 POST할 수 있고, 기존의 글을 PUT 또는 DELETE할 수 있다. 어떤 글에 달린 댓글 목록을 GET하거나 댓글을 새로 POST할 수 있다. 어떤 글에 유저당 한 번씩 좋아요를 누를 수 있고(like 페이지에서 POST), 좋아요 수를 볼 수 있다. 모든 좋아요 목록/디테일을 GET할 수 있고 그 중 유저 자신이 누른 좋아요를 DELETE할 수 있다. Mainpage 페이지에서는 댓글이 아닌 글(본 글)을 GET하거나 새로 POST할 수 있다.

3. frontend에 새로 구현/수정한 API 및 테스트코드

(TimeLinePage): 메인 페이지. 타임라인 글 읽기, 좋아요 누르기

(ArticleDetailPage): 상세 페이지. 해당 글에 달린 댓글 쓰기, 읽기

(AddArticlePage): 글쓰기 페이지. 글쓰기(댓글 쓰기도 동일)

매 페이지에서는 로그아웃이 가능하며, 일관된 레이아웃 디자인으로 구현되어 있다.

쓴 글을 Delete하거나 Edit하는 기능은 아직 구현되지 않았다.

4. 제출 서버 확인 및 테스트 스크립트 실행 방법

1. 로컬호스트에서 돌려보기

```
cd progress  
./init.sh
```

만약 실행이 안 된다면 `source ~/env/bin/activate` 를 하고 재시도합니다. tmux가 설치되어 있다는 전제조건이 있습니다. 이 테스트는 백엔드 테스트만을 포함합니다.

2. 원격 서버에서 돌려보기

```
cd progress  
python3 backend_tests.py http://wlxyzlw.iptime.org:8000/
```

만약 실행이 안 된다면 `source ~/env/bin/activate` 를 하고 재시도합니다. 이 테스트는 백엔드 테스트만을 포함합니다.

3. 이번 주에는 프론트엔드 테스트 코드를 제공하지 않습니다.

<http://wlxyzlw.iptime.org:3000/> 에서 수동으로 테스트하시기 바랍니다.

그리고 필독사항!

프론트엔드에서 버튼 하나를 누를 때마다 수동으로 새로고침(F5)을 눌러 주어야 이동한 페이지로 화면이 변합니다! 이는 Sign in, "반갑소 동무!", Write, Back to main, Like, Detail, Reply, Post 등의 버튼에 해당되며 Sign Up, Sign Out은 자동으로 redirect되니 새로고침 없이도 작동합니다.
그리고 위에서 설명하였지만 Delete, Edit 버튼은 아직 작동하지 않습니다.