

WEEKLY SEMINAR (A B A R A 2 0 2 3)

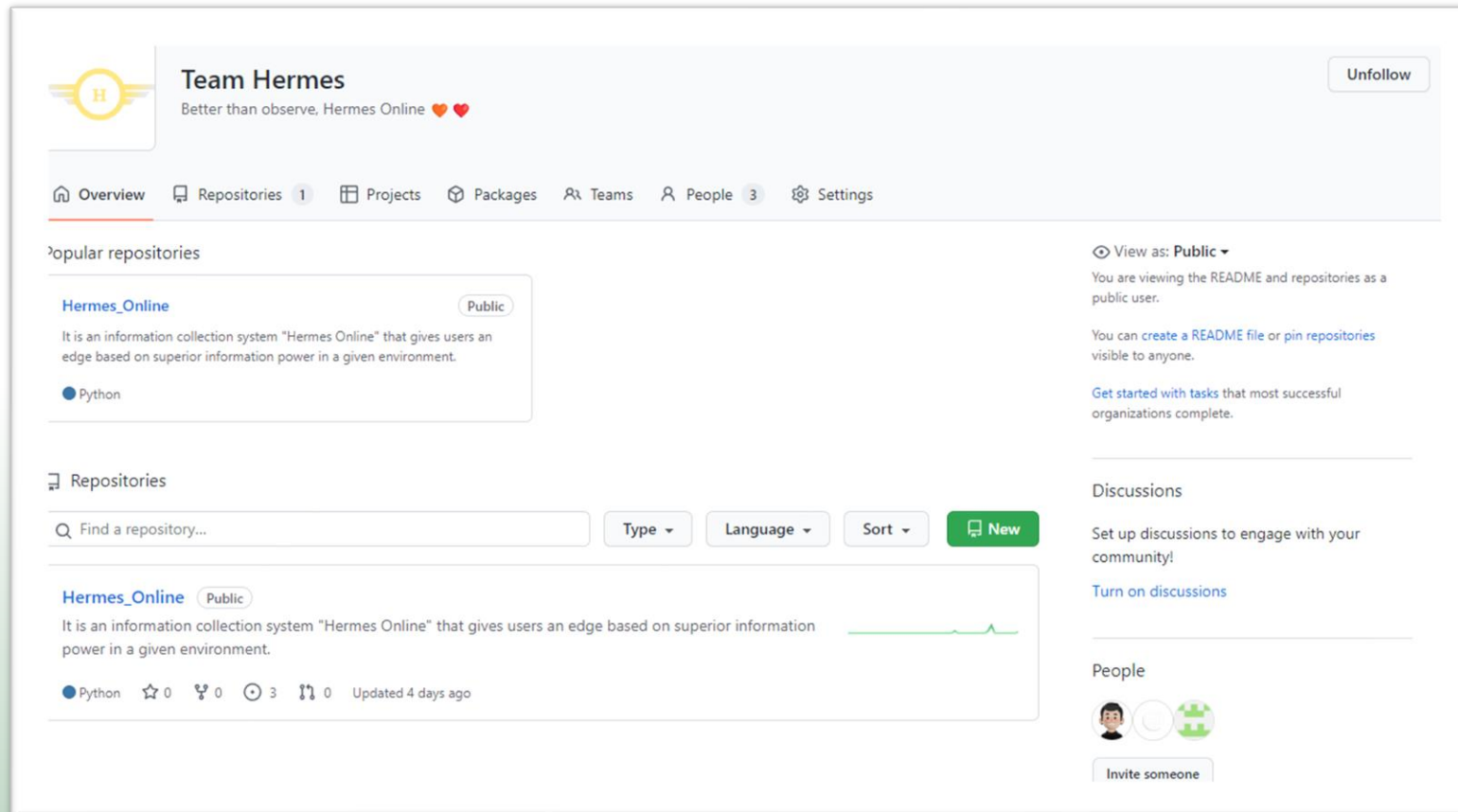
Jan.15. 2023



CONTENTS

- ◆ About 'Hermes'
- ◆ 'Hermes' Project Design
- ◆ 'Hermes' BE Parts
 - ◆ Feedback to Project procedure
 - ◆ Why We choose FastAPI?
 - ◆ How to design BE directory tree
- ◆ After Improvements
 - ◆ Renewal Point
 - ◆ After Plan

ABOUT 'HERMES'



The screenshot shows the profile page for 'Team Hermes'. At the top, there's a header with the team's logo (a yellow 'H' with wings), the name 'Team Hermes', and the tagline 'Better than observe, Hermes Online' with two red hearts. A 'Unfollow' button is in the top right. Below the header is a navigation bar with tabs: Overview (selected), Repositories (1), Projects, Packages, Teams, People (3), and Settings. The main content area is divided into three columns. The left column, titled 'Popular repositories', features a card for 'Hermes_Online' (Public) with a description: 'It is an information collection system "Hermes Online" that gives users an edge based on superior information power in a given environment.' and a Python language tag. The middle column, titled 'Repositories', has a search bar, filters for Type, Language, and Sort, and a 'New' button. It also displays the 'Hermes_Online' repository card with more details: 'It is an information collection system "Hermes Online" that gives users an edge based on superior information power in a given environment.', Python language, 0 stars, 0 forks, 3 watchers, 0 issues, and 'Updated 4 days ago'. The right column contains a 'View as: Public' dropdown, a note about viewing the README and repositories as a public user, links to 'create a README file' and 'pin repositories', a link to 'Get started with tasks', a 'Discussions' section with a link to 'Turn on discussions', and a 'People' section with three avatars and an 'Invite someone' button.

Team Hermes
Better than observe, Hermes Online ❤️❤️

Unfollow

Overview Repositories 1 Projects Packages Teams People 3 Settings

Popular repositories

Hermes_Online Public

It is an information collection system "Hermes Online" that gives users an edge based on superior information power in a given environment.

Python

Repositories

Find a repository... Type Language Sort New

Hermes_Online Public

It is an information collection system "Hermes Online" that gives users an edge based on superior information power in a given environment.

Python 0 stars 0 forks 3 watchers 0 issues Updated 4 days ago

View as: Public ▼

You are viewing the README and repositories as a public user.

You can [create a README file](#) or [pin repositories](#) visible to anyone.

[Get started with tasks](#) that most successful organizations complete.

Discussions

Set up discussions to engage with your community!

[Turn on discussions](#)

People

Invite someone

'HERMES' PROJECT DESIGN


main 4 branches 0 tags

Go to file Add file <> Code

ash-hun Update README.md fddb54b 4 days ago 12 commits

be/server	Delete backend.txt	2 months ago
design	upload structure	2 months ago
fe	upload structure	2 months ago
README.md	Update README.md	4 days ago

README.md



Service

Hermes Online

About

학교 홈페이지의 게시글을 크롤링해서 대시보드로 만들어봅시다!

About

It is an information collection system "Hermes Online" that gives users an edge based on superior information power in a given environment.

Readme 0 stars 1 watching 0 forks

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Languages

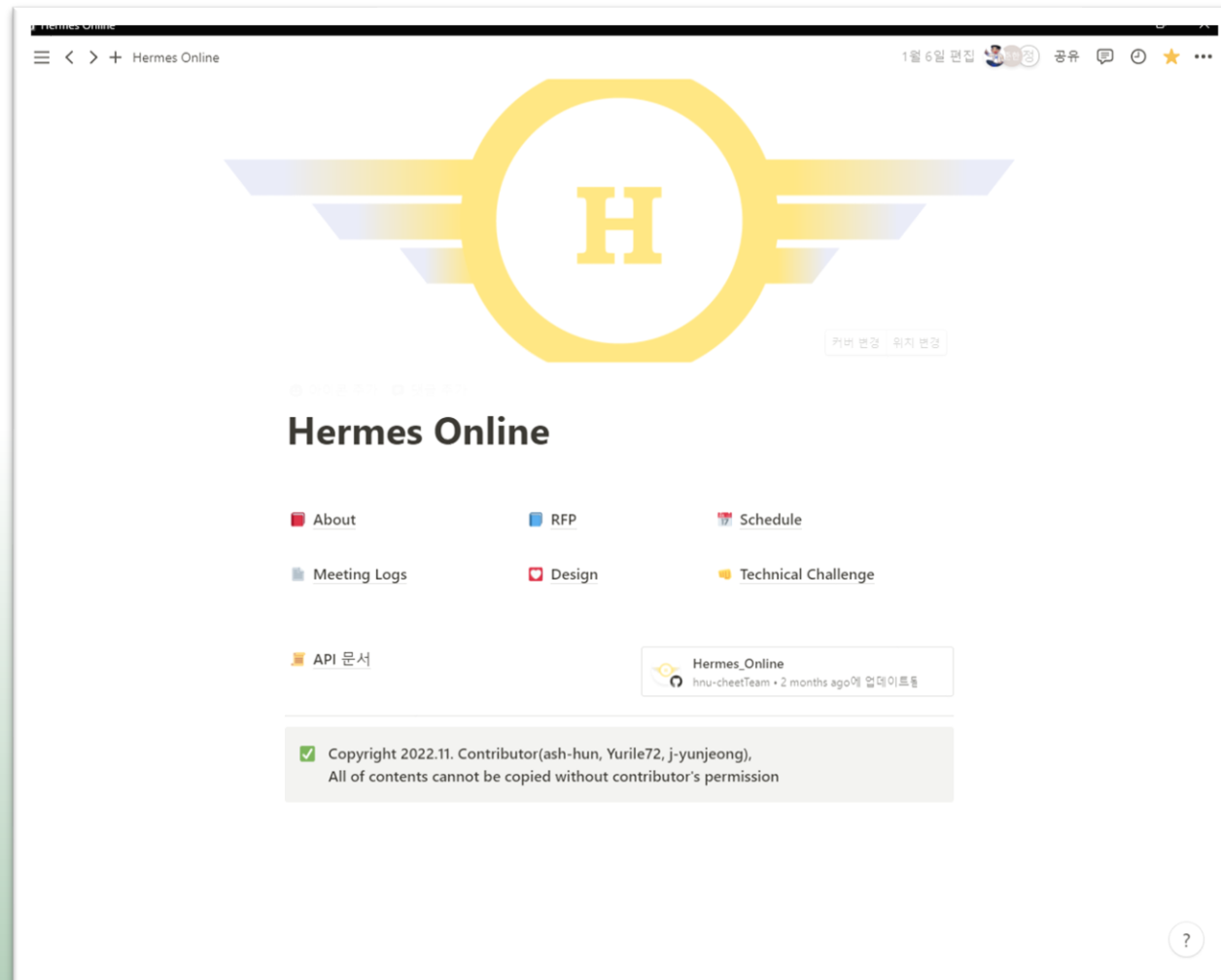
Python 100.0%

Suggested Workflows

Based on your tech stack

Python package By GitHub Actions [Configure](#)

'HERMES' PROJECT DESIGN



'HERMES' BE PARTS



FEEDBACK TO PROJECT PROCEDURE

- ◆ 미숙한 fastAPI 활용
- ◆ 원초적인 프로젝트 구조파악 미흡
- ◆ 과하게 러프한 프로젝트 진행
- ◆ 첫 BE framework 사용과 동시에 다수의 첫번째 시도가 내포되어 진행이 더더짐
 - ◆ JWT
 - ◆ Schema Structure

WHY WE CHOSE FASTAPI

올바른 프레임워크를 선택하기 위해 고려할 사항

파이썬 프레임워크를 선택할 때 가장 주안점을 두어야 할 부분은 내가 진행할 프로젝트의 사이즈와 복잡도입니다: 소규모 프로젝트를 위한 마이크로프레임워크와 대규모 프로젝트를 위한, 다양한 기능을 지닌 풀스택 프레임워크가 그 선택지입니다. 풀스택 프레임워크는 어떤 개발자에겐 지나치게 방대하고 배워야 할 게 너무 많을 수도 있습니다.

두 번째, 종으로 횡으로 확장이 용이한지 따져보아야 합니다. 프로젝트 수행 중에 기능을 추가해야 할 수도, 서버를 옮겨서 배포해야 하는 경우도 있을 것이기 때문입니다.

세 번째, 프레임워크에 대해 학습하고 지식을 얻을 수 있는 문서(documentation)나 커뮤니티도 접근하기 쉬워야 합니다. 개발 중에 막히는 부분이 생기는 일은 다반사고 그 답을 얻는 게 어떤 경우 무척 어려울 수도 있기 때문입니다.

마지막으로 간과하기 쉬운 부분인데, 프레임워크의 라이선스 정책을 고려해야 합니다. 어떤 프레임워크는 오픈소스이지만 어떤 프레임워크는 소유권을 요구하는 경우가 있습니다.

WHY WE CHOSE FASTAPI

풀스택 프레임워크(Full-stack Framework)

풀스택 프레임워크는 소위 올인원 솔루션이라고 생각하면 됩니다. 프론트엔드 UI나 백엔드 로직, 데이터베이스와의 소통까지 모두 지원하기 때문입니다. 풀스택 프레임워크는 어플리케이션을 빌드하는 데 필요한 모든 것을 제공하기 때문에 가장 대중적인 프레임워크이기도 합니다.

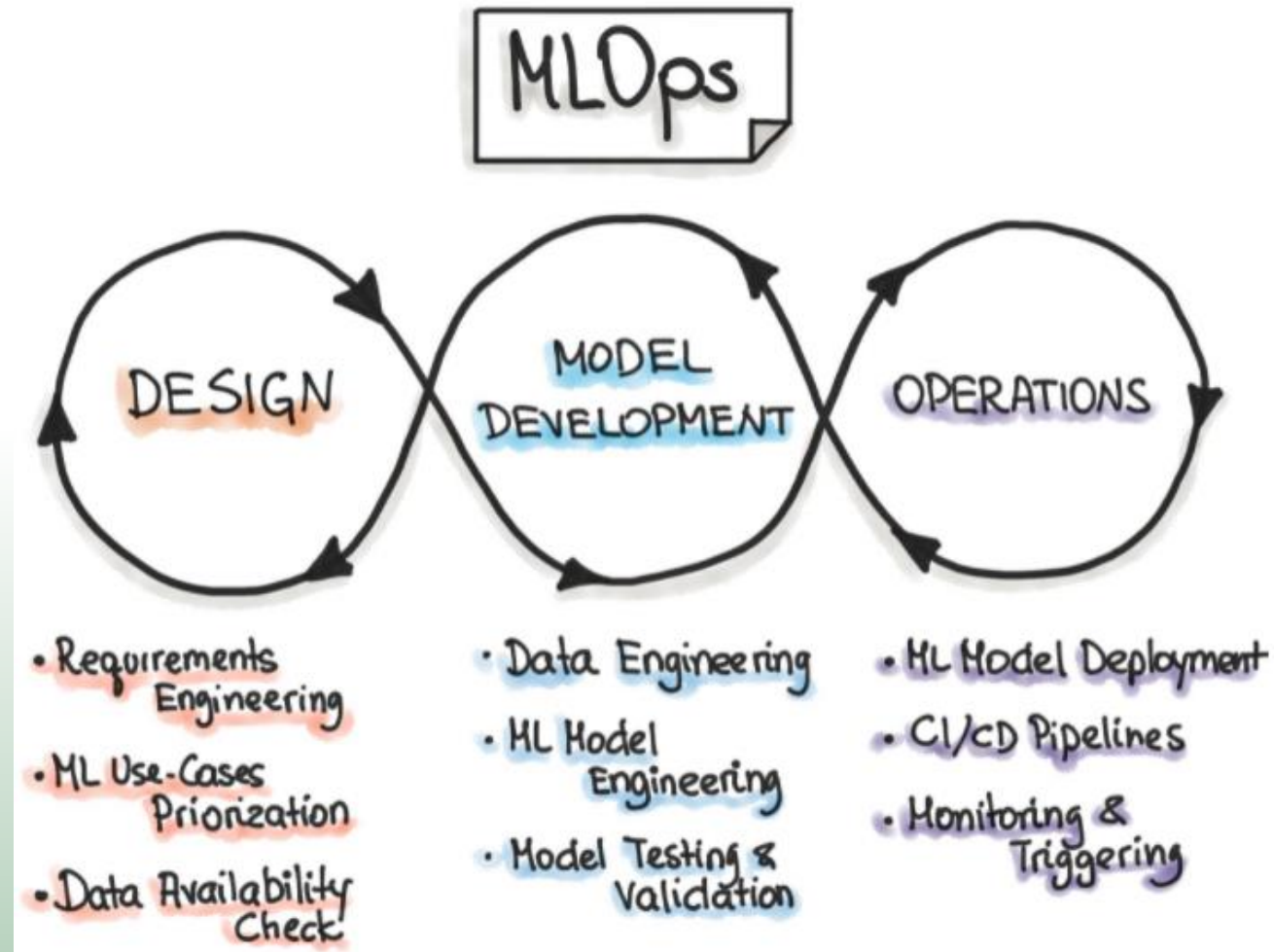
마이크로프레임워크(Micro-framework)

풀스택프레임워크가 아닌 프레임워크 대부분은 마이크로프레임워크라고 생각하면 되는데, 풀스택 프레임워크만큼 풍성한 기능을 가지고 있지 않고 핵심적인 기능만을 제공하는 경우가 많고, 부가적인 기능은 따로 추가를 해야 합니다. 이런 프레임워크에서는 개발자에게 요구되는 의사결정이 단순하기 때문에 개발을 시작하는 초심자에게 권합니다.

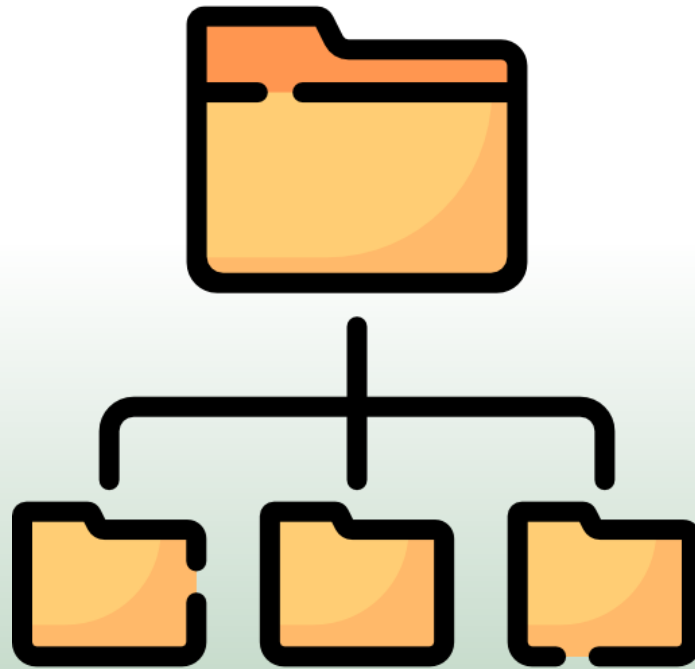
WHY WE CHOSE FASTAPI

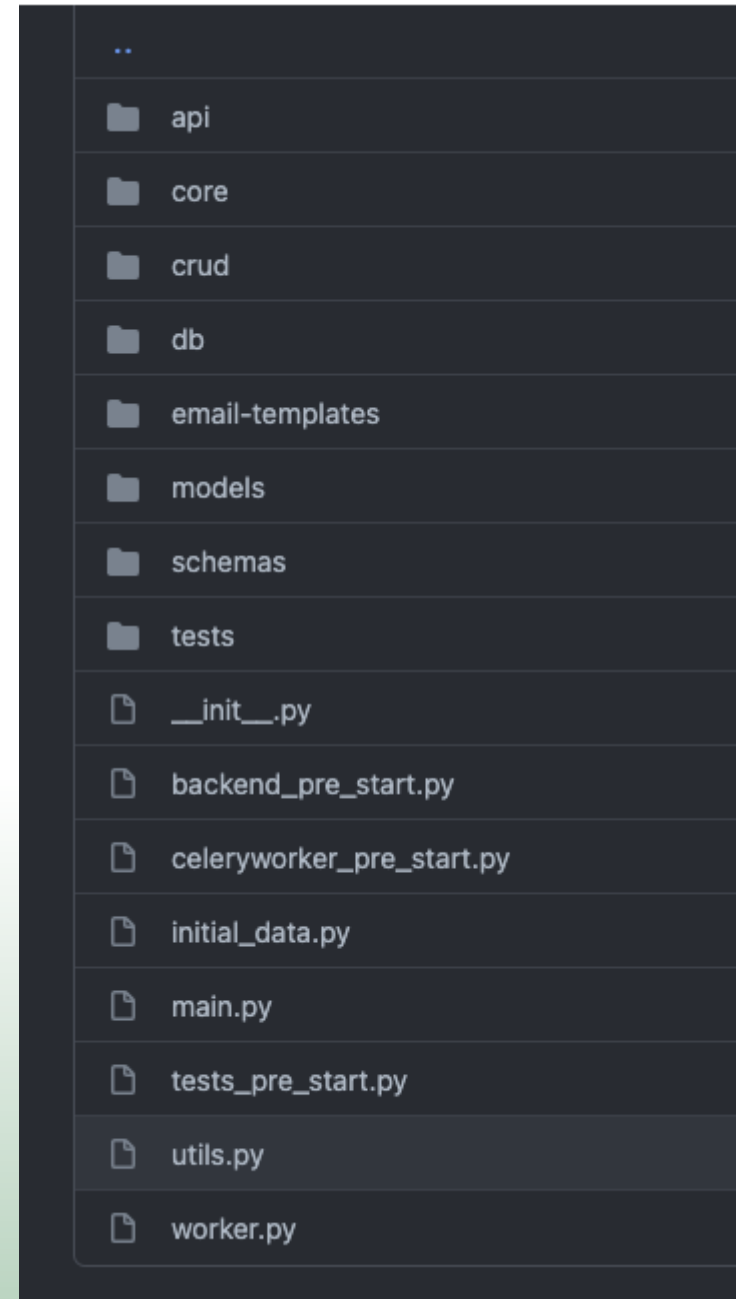
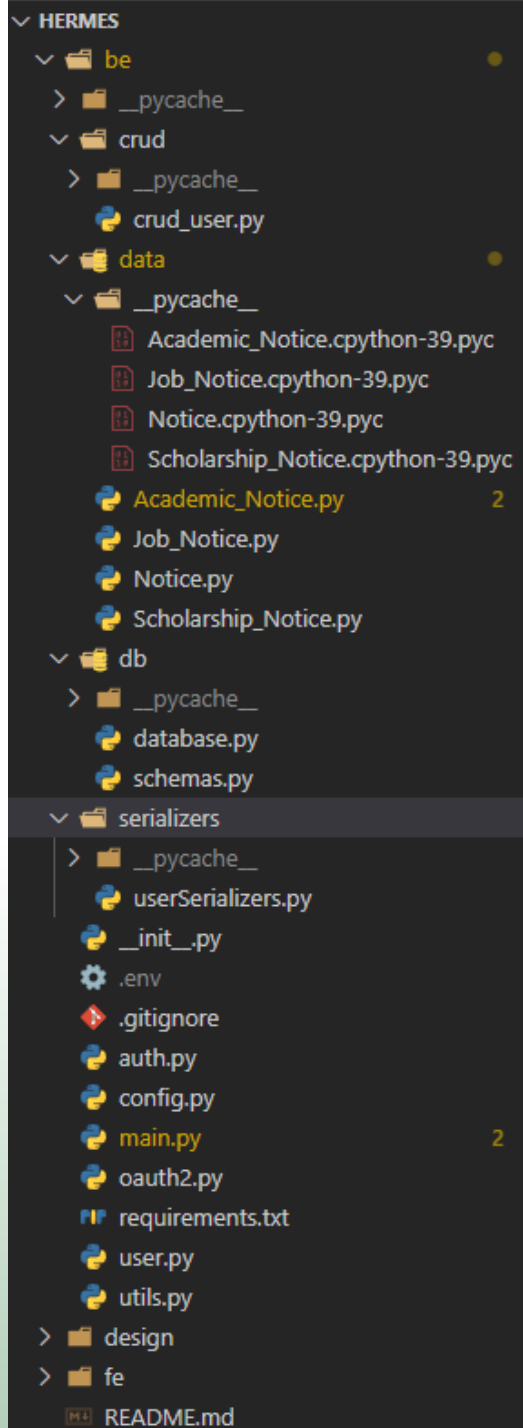


WHY WE CHOSE FASTAPI

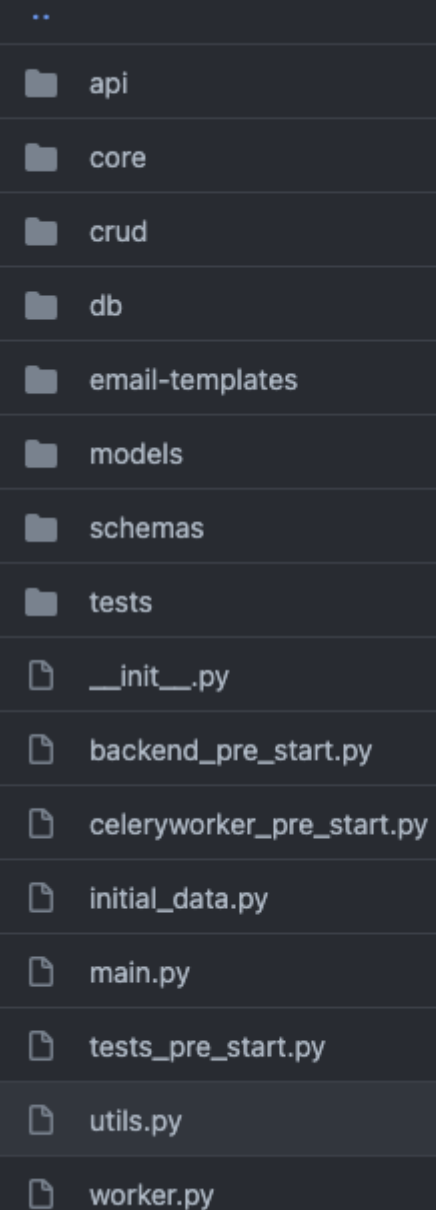


HOW TO DESIGN BE DIR TREE





- api : 엔드포인트별로 파일 구분
 - api_v1
 - endpoints: items.py, login.py, users.py, utils.py
- core: 공통 파일 config.py, security.py, celery_app.py
- crud: 기본 get, create, update, remove base.py, crud_item.py, crud_user.py
- db: 데이터베이스 관련 파일 base.py, base_class.py, init_db.py, session.py
- models: 데이터베이스 테이블과 매칭되는 모델 item.py, user.py
- schemas: 스프링의 DTO와 비슷 item.py, user.py, msg.py, token.py
- tests: 테스트 파일



A screenshot of a file explorer window with a dark theme. The left sidebar shows a tree view with a blue folder icon and the text '..'. The main pane displays a list of files and folders. Folders are indicated by a folder icon and files by a document icon. The items listed are: api, core, crud, db, email-templates, models, schemas, tests, __init__.py, backend_pre_start.py, celeryworker_pre_start.py, initial_data.py, main.py, tests_pre_start.py, utils.py, and worker.py.

- ..
- api
- core
- crud
- db
- email-templates
- models
- schemas
- tests
- __init__.py
- backend_pre_start.py
- celeryworker_pre_start.py
- initial_data.py
- main.py
- tests_pre_start.py
- utils.py
- worker.py

AFTER IMPROVEMENTS : RENEWAL POINT

- ◆ 프로젝트 구조변경
- ◆ 정기적으로 프로젝트 진행 + 천천히, 점진적으로 성과 구축
- ◆ 기존에 사용된 기술의 100% 파악 *
- ◆ 효율성과 성능개선을 우선으로 한 개편 *

AFTER IMPROVEMENTS : AFTER PLAN

- ◆ 개발환경 개편 (Windows -> Linux 기반(wsl)) *
- ◆ Docker를 이용한 배포 *
- ◆ Github action을 이용한 CI/CD *
- ◆ mongoDB에 이어 캐싱DB(Redis)를 사용한 효율성 개선 *
- ◆ 데이터 프로세싱 효율화 *