



오픈소스SW기초 6분반

분산 투자 전략 지원, SABU

#주식 #분산 투자

사고8조

정보통계학과 32200472 김동혁
컴퓨터공학과 32211228 김태형
소프트웨어학과 32232597 엄세훈

목차

1

GitHub 관리

2

데이터 수집 & 모델링 계획

3

서비스 설계

4

와이어프레임

5

API 명세

6

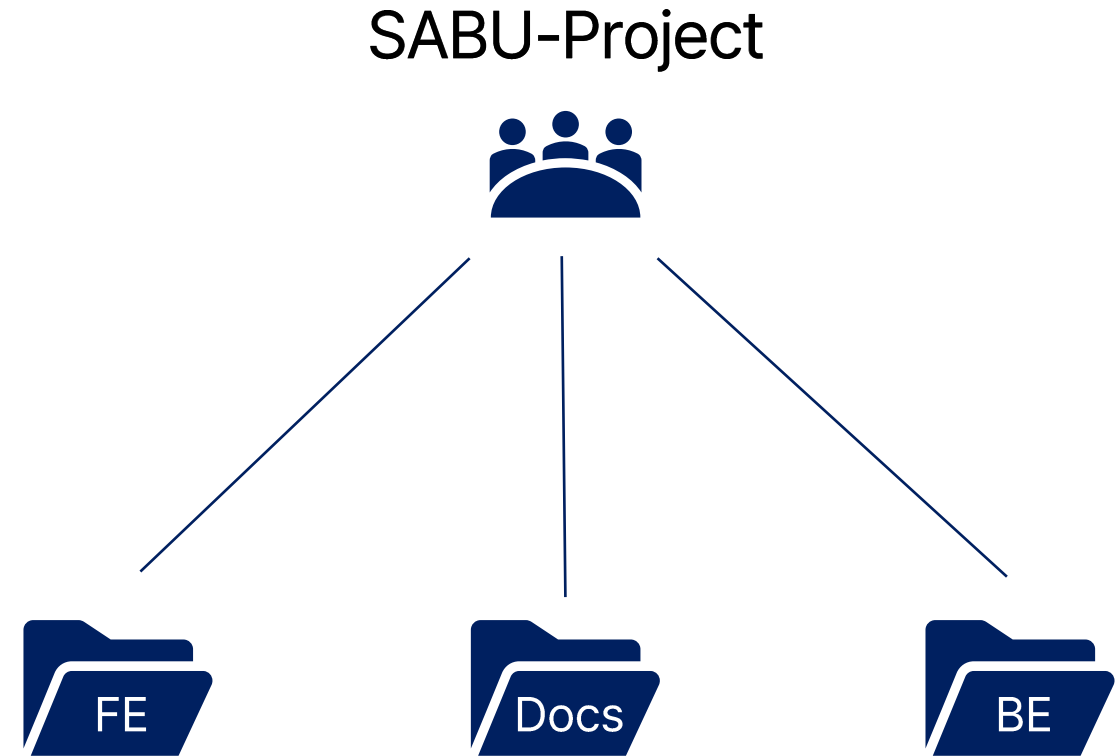
Open Source

1. GitHub 관리

GitHub - Project

Project를 생성하여 작업 리스트 관리

- ✓ 전반적인 작업의 진행 상황, 완료 상태 등 한눈에 볼 수 있음
- ✓ 어떤 일을 하거나 작업이 필요한 것은 SABU-Project → 이슈 생성
- ✓ 하위에 각각 Frontend, Backend, Docs 레포지토리 존재
 - Frontend : Front와 관련한 논의, 코드 작성 등 작업 시행
 - Backend : Back과 관련한 논의, 코드 작성 등 작업 시행
 - Docs : 모든 프로젝트 관리, 문서 작업 시행



GitHub - Project

1. GitHub관리

DKUOpenSource-SABU

Add status update

Not Completed

All Job

Status board

Roadmap

In review

New view

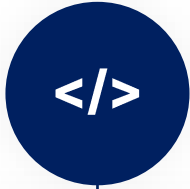
-status:Done

10

Discard

Save

Title	...	Status	...	Assignees	...	Linked pull requests	...	+
<div>DKUOpenSource-SABU/Docs 5</div>								
1	docs: 기능 명세하기 #2	In progress		hyuk-dd, Kim-Taeh...		#21		
2	docs: 중간 보고 발표 자료 ... #17	In progress		hyuk-dd and sehoo...		#23		
3	docs: 중간 보고 발표 자료 #23	In progress		hyuk-dd				
4	docs: 기능 명세하기 #21			Kim-Taehyeong				
5	feat: SABU 서비스 API 명세... #24			sehooneom04				
<div>repo:Docs</div> <div>Search issues and pull requests, create a new issue, or add multiple items</div>								
<div>DKUOpenSource-SABU/backend 1</div>								
6	feat: 추출 데이터 논의 #3			hyuk-dd, Kim-Taeh...				
<div>repo:backend</div> <div>Search issues and pull requests, create a new issue, or add multiple items</div>								
<div>DKUOpenSource-SABU/frontend 4</div>								
7	feat: 첫 페이지 화면 제작 #8			Kim-Taehyeong		#9		
8	feat : 화면 2개 제작 #9			Kim-Taehyeong				
9	feat: 백테스팅 화면 제작 및 ... #10			Kim-Taehyeong				
10	feat : 백테스팅 화면 제작 및... #11			Kim-Taehyeong				
<div>repo:frontend</div> <div>Search issues and pull requests, create a new issue, or add multiple items</div>								



코드 컨벤션

본문 내용을 잘 정리 해서
입력해주세요. 본문 내용이
잘 전달 시켜서 성공적인
제안서를 제출해주세요.

클라이언트를
감동시켜주세요. 본문 내용을
잘 정리해주세요



이슈 컨벤션

본문 내용을 잘 정리 해서
입력해주세요. 본문 내용이
잘 전달 시켜서 성공적인
제안서를 제출해주세요.

클라이언트를
감동시켜주세요. 본문 내용을
잘 정리해주세요



커밋 컨벤션

본문 내용을 잘 정리 해서
입력해주세요. 본문 내용이
잘 전달 시켜서 성공적인
제안서를 제출해주세요.

클라이언트를
감동시켜주세요. 본문 내용을
잘 정리해주세요

Frontend

코드 컨벤션

사용 언어

- JavaScript

기본 규칙

- 들여쓰기(Indentation): 2칸 (스페이스 사용)
- 함수/변수명: snake_case
- 컴포넌트명: PascalCase
- 파일명 : PascalCase.jsx 또는 kebab-case.js
- 세미콜론: 붙이기
- 따옴표: "더블 쿼터"
- 상수: UPPER_SNAKE_CASE

컨벤션 예시

```
// UserCard.jsx
import React from "react";

const TYPE = 1;

function UserCard({ name }) {
  return <div className="user-card">{name}</div>;
}

export default UserCard;
```



사용 언어: JavaScript



들여쓰기, 함수/변수명, 컴포넌트명, 파일명 등 기본 규칙 지정



Frontend 코드 컨벤션 예시

Backend

코드 컨벤션

사용 언어

- Python

기본 규칙

- 들여쓰기(Indentation): 4칸 (스페이스 사용)
- 최대 줄 길이: 79자
- 함수/변수명: snake_case
- 클래스명: PascalCase
- 상수: UPPER_SNAKE_CASE
- 불필요한 공백 금지:
 - a = 1 (O) / a=1 (X)

정렬, 줄바꿈

- 불필요한 줄바꿈 피하기
- 함수 간에는 2줄 띄우기
- import는 표준 -> 서드파티 -> 로컬 순서로 정리하고 각각은 빈 줄로 구분

컨벤션 예시

```
import os
import sys

import requests

from my_module import something

class TestClass():
    print("Something")

def test_function():
    print("Something")
```



사용 언어: Python



들여쓰기, 최대 줄 길이, 함수/변수명, 클래스명 등 기본 규칙 지정



Backend 코드 컨벤션 예시

이슈 컨벤션

이슈 작성 규칙

이슈 제목(Title)

- [타입] 간결하고 명확한 설명
- 타입은 커밋 타입과 동일: feat, fix, docs, style, refactor, test, chore, perf, ci
- 예시: [feat] 사용자 로그인 기능 추가

이슈 본문(Body)

설명

- 어떤 작업을 할 것인지 구체적으로 작성

작업 항목 (Checklist)

- ☐ 해야 할 작업을 체크리스트 형식으로 작성

기타 참고사항

- 관련 링크, 스크린샷 등 추가

이슈 템플릿 예시

설명
사용자가 이메일과 비밀번호를 통해 로그인할 수 있는 기능을 추가한다.

작업 항목
- [] 로그인 API 생성
- [] JWT 토큰 발급 기능 추가
- [] 로그인 실패 시 에러 메시지 반환

참고사항
- 디자인 시안: [링크]
- 관련 문서: [링크]



작성 필요

Commit Message Convention

커밋 메시지를 일관성 있게 작성하기 위한 가이드입니다.

커밋 타입 정리

타입	설명	이모지	예시 설명
feat	새로운 기능 추가	🚀	로그인 기능을 새로 개발했다
fix	버그 수정	🐛	회원가입할 때 이메일 중복 체크가 제대로 안 되는 문제를 고쳤다
docs	문서 수정 (README 등)	📖	README 파일에 설치 방법을 추가했다
style	포매팅, 세미콜론 누락 등 기능 변화 없는 수정	💡	코드 스타일을 맞추기 위해 포맷을 정리하고 세미콜론을 추가했다
refactor	리팩토링 (동작에는 변화 없음)	♻️	중복된 API 호출 코드를 깔끔하게 리팩토링했다
test	테스트 코드 추가/수정	✅	로그인 기능을 검증하는 테스트 코드를 작성했다
chore	빌드 설정, 패키지 추가 등 기타 변경	🔧	프로젝트에 eslint 설정 파일을 추가했다
perf	성능 향상 관련 커밋	⚡	데이터 조회 속도를 빠르게 개선했다
ci	CI 설정 변경	👤	GitHub Actions 워크플로를 수정해서 자동 배포가 되게 했다

커밋 메시지 작성 규칙

1. 제목(Subject)

- 한 줄로 요약 (50자 이내)
- 첫 글자는 대문자로
- 마지막에 마침표(.) 사용하지 않기
- 동사 원형 사용 ("Fixes" 대신 "Fix" 사용)

2. 본문(Body)

- 제목과 본문은 한 줄 띄워 분리
- 본문은 "무엇을", "왜" 했는지를 자세히 설명
- 72자마다 줄 바꿈 권장

3. 꼬리말(Footer)

- 이슈 트래킹 시 사용
- 예시: Closes #123, Related to #456

커밋 메시지 예시

🚀 feat: 사용자 로그인 기능 추가

사용자가 이메일과 비밀번호로 로그인할 수 있도록 인증 API를 추가했다.
이 기능을 통해 회원 전용 페이지 접근이 가능하다.

Closes #10

🐛 fix: 회원가입 시 이메일 중복 검사 오류 수정

이메일 중복 검사 API가 항상 true를 반환하던 문제를 수정했다.
이제 실제 데이터베이스를 조회하여 정확한 결과를 반환한다.

Related to #12

커밋 컨벤션

```
#####
# <타입> : <제목> 형식으로 제목을 아래 공백 줄에 작성하세요.
# 제목은 50자 이내로 작성합니다.
# 변경사항이 "무엇"인지 명확하게 작성합니다.
# 제목 끝에 마침표를 사용하지 않습니다.
# 첫 글자는 소문자 타입 후, 제목은 동사 원형으로 시작합니다.
# 예시: feat: 사용자 로그인 기능 추가
```

```
#####
# 본문(Body)을 아래에 작성합니다.
# 제목과 본문 사이에는 반드시 한 줄 공백을 둡니다.
# 본문은 "무엇을", "왜" 변경했는지 상세히 설명합니다.
# 여러 줄 작성 시 "-"로 구분하며, 한 줄은 72자 이내로 유지합니다.
```

```
#####
# 꼬릿말(Footer)을 아래에 작성합니다.
# 관련 이슈 번호를 명시할 때 사용합니다.
# 예시: Closes #10, Related to #12
```

```
#####
# 타입 가이드
# 🚀 feat : 새로운 기능 추가
# 🐛 fix : 버그 수정
# 📄 docs : 문서 수정 (README 등)
# 🍷 style : 포매팅, 세미콜론 누락 등 기능 변화 없는 수정
# ♻️ refactor : 리팩토링 (동작에는 변화 없음)
# ✅ test : 테스트 코드 추가/수정
# 🛠️ chore : 빌드 설정, 패키지 추가 등 기타 변경
# ⚡ perf : 성능 향상 관련 커밋
# 🤖 ci : CI 설정 변경
#####
```

gitmessage.txt

GitHub 작업 순서

1. GitHub관리

1. 해야 할 작업, 논의할 내용에 대해 Issue 생성
2. Issue 번호와 연결되는 Branch 생성
3. Issue와 관련한 작업 실행 후 생성한 Issue Branch에 Commit
4. Push 후 Pull Request 생성
5. PR에 대해 다른 두 팀원의 리뷰 및 승인, 마지막 검토자가 Merge 시행

GitHub 작업 순서

1. GitHub관리

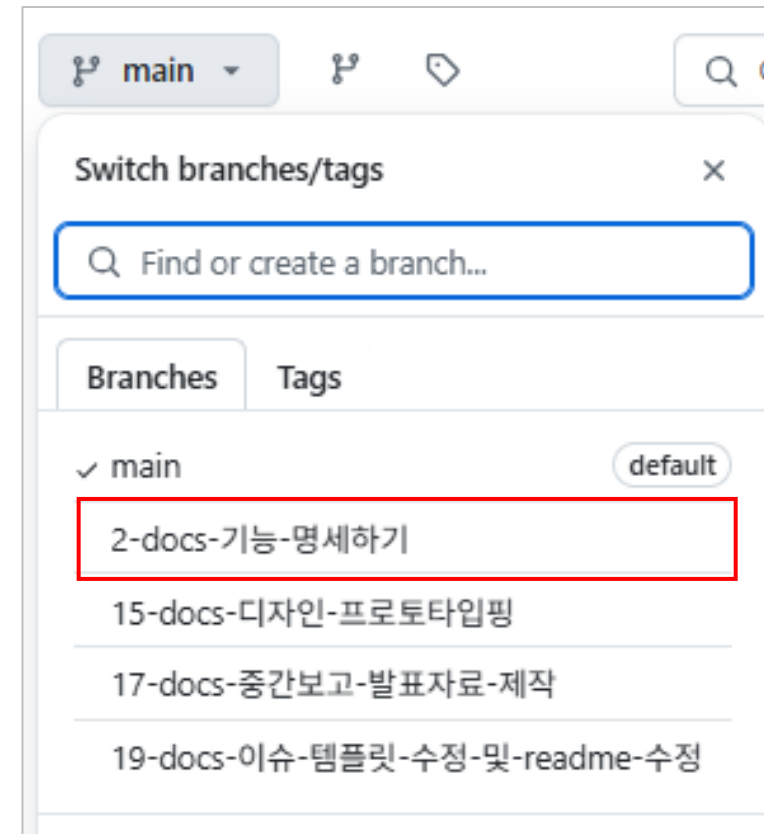
1. 해야 할 작업, 논의할 내용에 대해 Issue 생성

The screenshot shows a GitHub issue page for 'docs: 기능 명세하기 #2'. The issue is labeled 'Feature' and is public. It was opened last week by Kim-Taehyeong and edited by hyuk-dd. The issue description is in Korean, stating that the core features will be specified in the document. The task list includes '백테스팅 기능 명세' and '유사 주식 클러스터링 기능 명세'. The issue is assigned to Kim-Taehyeong, hyuk-dd, and sehooneom04. It is labeled 'documentation' and has a 'Feature' type. The project is 'DKUOpenSource-SABU' with a status of 'In progress'. The issue history shows that it was added to the project, assigned, added to the project, added the 'Feature' issue type, and moved to the Backlog in the 'DKUOpenSource-SABU' project.

GitHub 작업 순서

1. GitHub관리

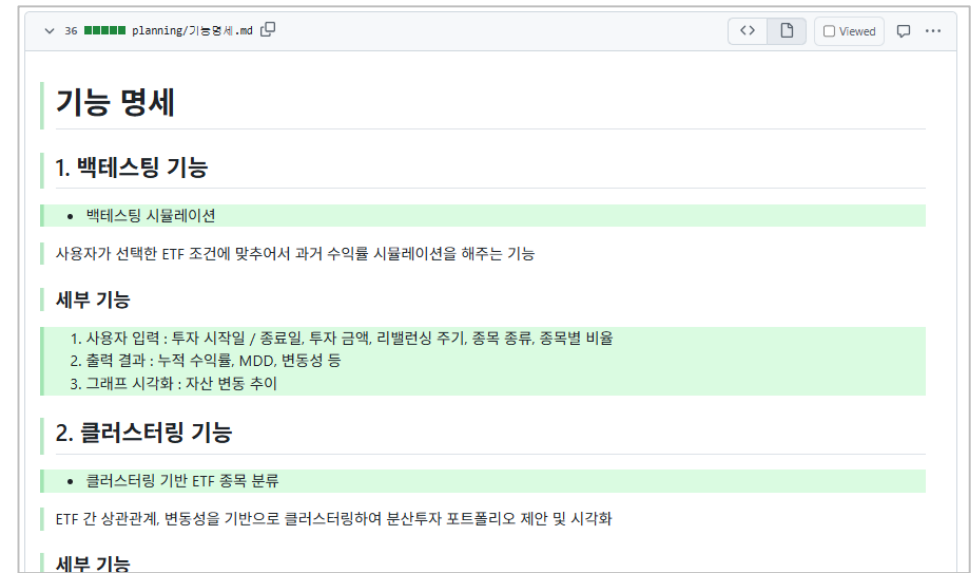
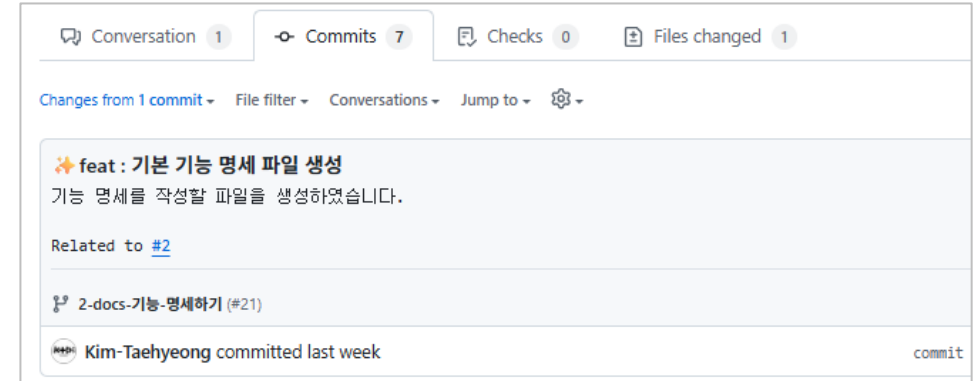
2. Issue 번호와 연결되는 Branch 생성



GitHub 작업 순서

1. GitHub관리

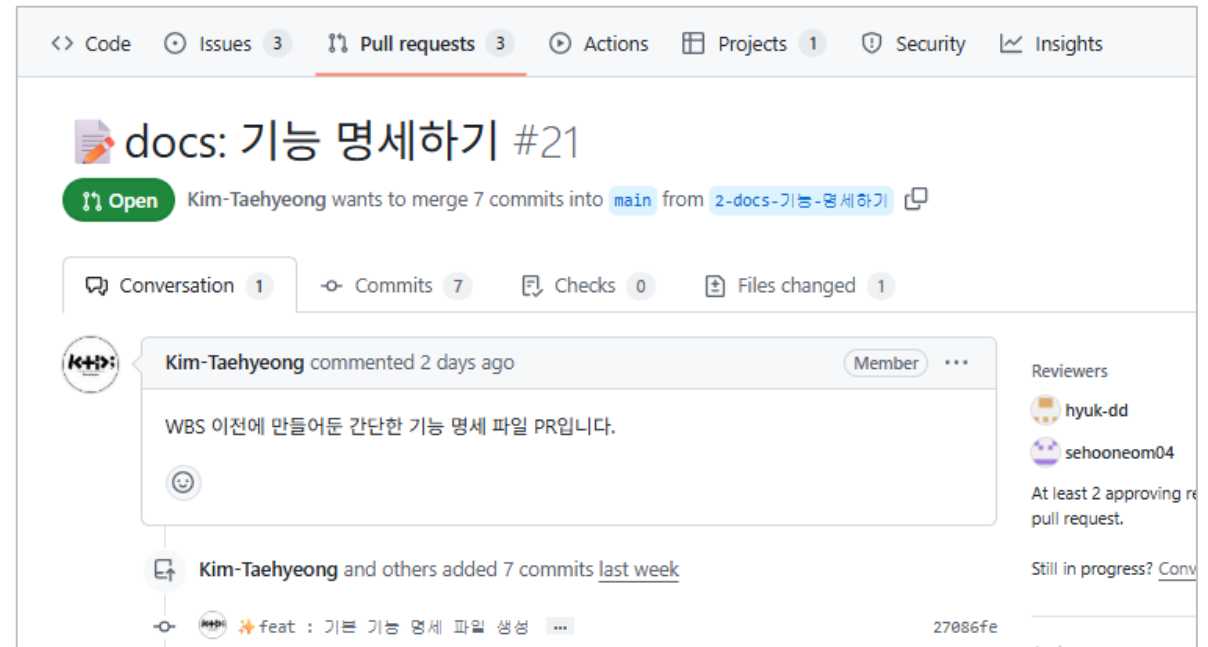
3. Issue와 관련한 작업 실행 후 생성한 Issue Branch에 Commit



GitHub 작업 순서

1. GitHub관리

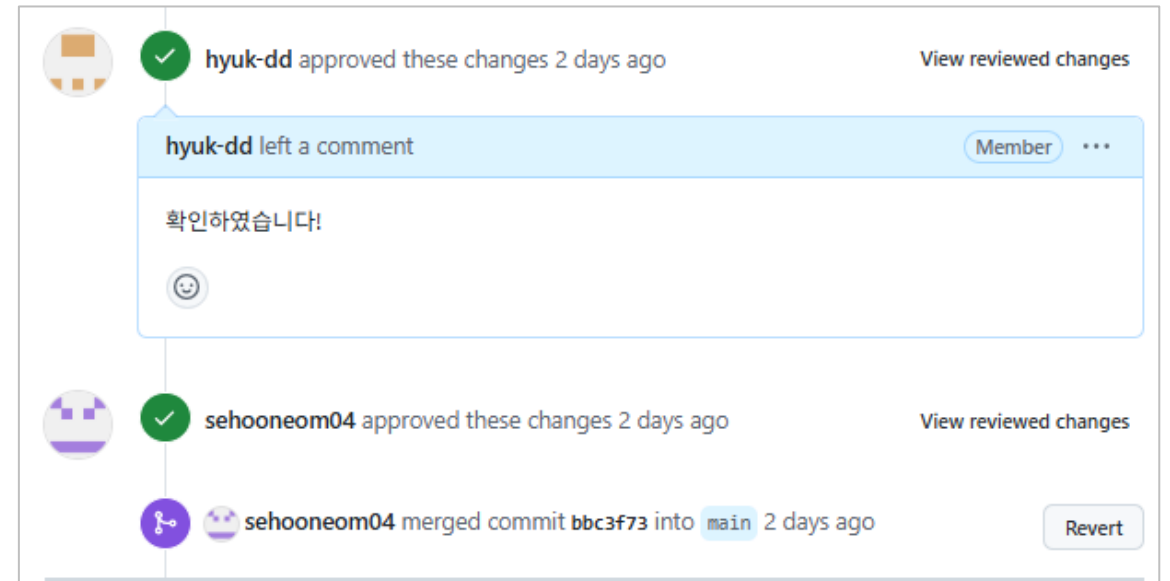
4. Push 후 Pull Request 생성



GitHub 작업 순서

1. GitHub관리

5. PR에 대해 다른 두 팀원의 리뷰 및 승인, 마지막 검토자가 Merge 시행



2. 데이터 수집 & 모델링 계획

데이터 수집

2.데이터 수집&모델링 계획



데이터 수집 계획 내용 작성

클러스터링 구현 계획

- ✓ K-Means 클러스터링 어떻게 활용할지 등 내용 설명

3. 서비스 설계

- ✓ MVP(Minimal Viable Product) 기준으로 최소한으로 구현해야 할 WBS 작성

추후 추가 기능



추가로 구현해볼 기능 설명

추후 추가 기능

키워드01 입력

본문 내용을 잘 정리 해서 입력해주세요. 본문 내용이 잘 전달
시켜서 성공적인 제안서를 제출해주세요



키워드02 입력

본문 내용을 잘 정리 해서 입력해주세요. 본문 내용이 잘 전달
시켜서 성공적인 제안서를 제출해주세요



키워드03입력

본문 내용을 잘 정리 해서 입력해주세요. 본문 내용이 잘 전달
시켜서 성공적인 제안서를 제출해주세요



키워드04 입력

본문 내용을 잘 정리 해서 입력해주세요. 본문 내용이 잘 전달
시켜서 성공적인 제안서를 제출해주세요



키워드입력

4. 와이어프레임

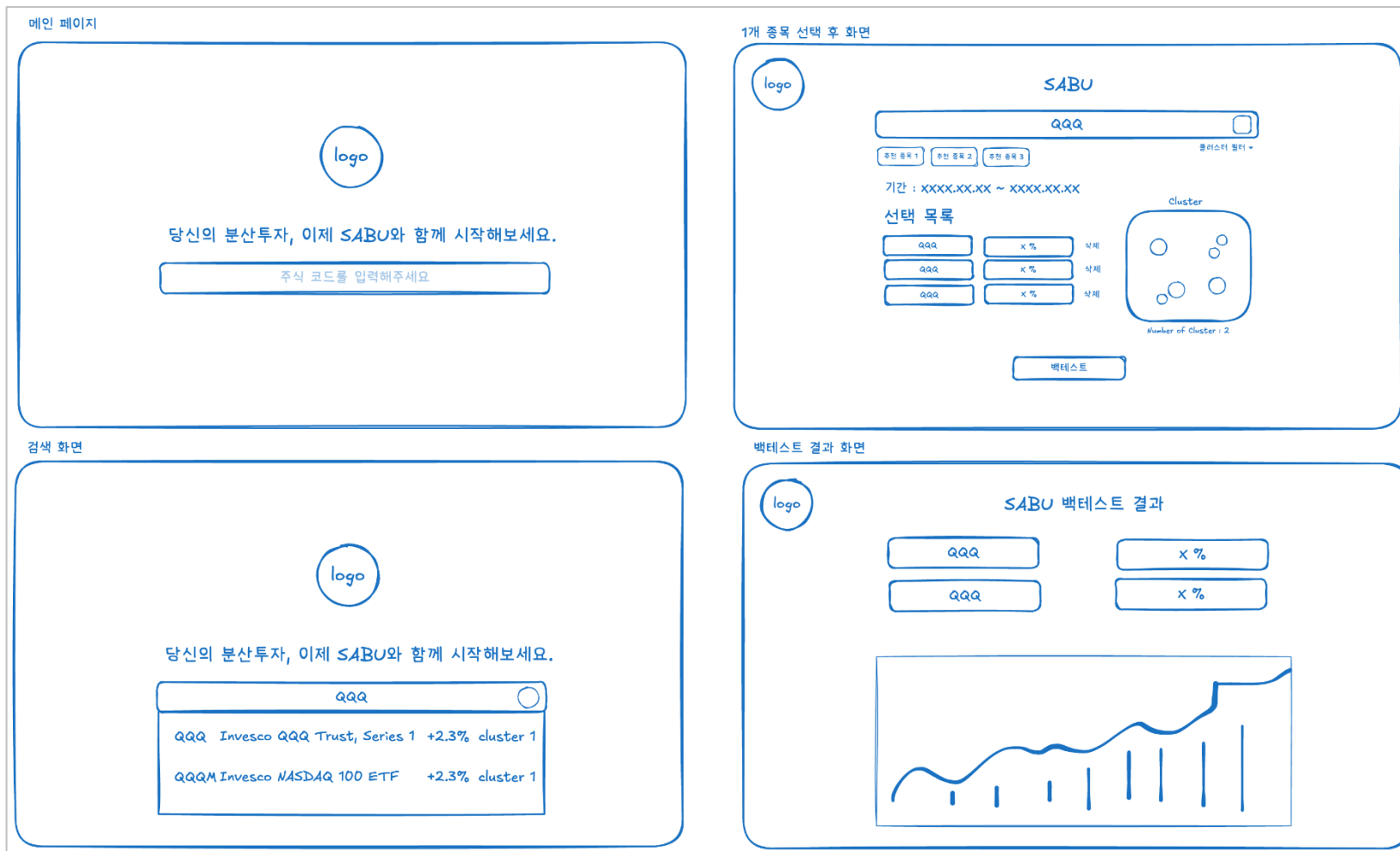
세훈 님이 작성하신 내용 및 사진 추가

와이어 프레임과 더불어 URL 시연 장면

<https://dkuopensource-sabu.github.io/frontend/>

소제목

4.와이어프레임



소제목




당신의 분산투자, 이제 SABU와 함께 시작해보세요.

Q

\$

▶

섹터	티커	이름	현재가	등락률
ETF	QQQ	Invesco QQQ Trust	\$439.87	+0.42%
Automotive	TSLA	Tesla Inc.	\$171.23	-2.35%



Q

여기에 주식 코드를 입력해주세요!

▶

백테스트 설정

시작 날짜

2025-05-01

종료 날짜

2025-05-11

초기 자본 (\$)

ex) 10000

수수료 (%)

ex) 0.1

선택한 종목

섹터	티커	비율 (%)
ETF	QQQ	ex) 20 %
Automotive	TSLA	ex) 20 %

클러스터링

클러스터링

클러스터링

클러스터링

5. API 명세

세훈 님이 작성하신 내용 및 사진 추가

6. Open Source

세훈 님이 작성하신 내용 및 사진 추가

Q&A