

PNU SW 학습공동체 최종보고서

PNUSW-05

링크 : <https://www.hambookstocks.store/>

가. 배경 및 필요성

햄북 Stock 는 실제 주식 투자의 진입 장벽이 높아 실제 돈으로 선뜻 발을 딛기 어려운 사람들을 위해 개발된 모의 주식 투자 플랫폼입니다.

주요 배경:

- 실제 주식 투자는 높은 금전적 위험과 복잡한 시장 환경으로 인해 초보자들이 접근하기 어려움
- 가상 환경에서 안전하게 투자 경험을 쌓을 수 있는 플랫폼 필요
- 실시간 시장 환경 체험을 통한 금융 교육 효과

나. 개발목표 및 주요내용

핵심 목표:

- 직관적인 UI 와 백엔드 서버를 통한 실제 투자 환경 제공
- 가상 머니를 활용한 안전한 투자 체험
- 실시간 가격 변동과 다이내믹한 시장 환경 구현

주요 기능:

- 모의 주식 거래 시스템
 - 실시간 매수/매도 기능

- 가격 변동 알고리즘 ($\alpha = 0.1$ 계수 적용)
- 거래 내역 관리
- 뉴스 및 정보 시스템
 - 매일경제 최신 기사 크롤링
 - OpenAI API 를 활용한 뉴스 요약 기능
 - 실시간 뉴스 피드
- 소셜 기능
 - 웹소켓 기반 익명 채팅방
 - 수익률 기반 유저 랭킹 시스템 (명예의 전당)
 - 실시간 사용자 간 소통
- 개인화 기능
 - JWT 기반 사용자 인증 및 관리
 - 개인 주식 포트폴리오 관리
 - 나만의 주식 상장 시스템

다. 사회적 가치 도입 계획

교육적 가치:

- 금융 리터러시 향상을 위한 실습 환경 제공
- 위험 없는 투자 경험을 통한 금융 교육
- 실시간 시장 환경 체험을 통한 경제 이해도 증진

기술적 가치:

- 오픈소스 기반의 투명한 시스템 구축
- 확장 가능한 아키텍처로 다양한 금융 서비스 개발 기반 제공

2. 상세설계

가. 시스템 구성도

아키텍처:

Frontend (React) ↔ Backend (FastAPI) ↔ Database (MySQL + Redis)

기술 스택:

- Frontend: React 18.2.0, Axios, React Router DOM, Recharts, Swiper
- Backend: FastAPI, SQLAlchemy, APScheduler
- Database: MySQL, Redis (캐싱 및 세션 관리)
- 추가 기술: OpenAI API, WebSocket, JWT

주요 라이브러리:

- 백엔드: FastAPI, SQLAlchemy, PyJWT, Redis, OpenAI, BeautifulSoup4
- 프론트엔드: React, Axios, Recharts (차트 시각화), Swiper (슬라이더)

3. 개발결과

가. 전체시스템 흐름도

사용자 로그인 → JWT 토큰 발급 → Redis 세션 관리

↓

메인 페이지 → 뉴스/채팅/거래 선택

↓

각 기능별 실시간 처리:

- 뉴스: 크롤링 → GPT 요약 → 프론트엔드 전송
- 채팅: WebSocket 연결 → 실시간 메시지 브로드캐스트
- 거래: 매수/매도 요청 → 가격 변동 → 포트폴리오 업데이트

나. 기능설명

1. 사용자 인증 시스템

- JWT 토큰 기반 인증
- Redis 를 활용한 세션 관리
- 비밀번호 정규표현식 검증 (8~20 자, 숫자+영어 필수)
- 이메일 형식 검증

2. 주식 거래 시스템

- 실시간 매수/매도 기능
- 가격 변동 알고리즘 ($\alpha = 0.1$)
- 거래 내역 추적
- 잔액 관리 및 검증

3. 뉴스 시스템

- 매일경제 기사 자동 크롤링
- OpenAI GPT 를 활용한 기사 요약
- 실시간 뉴스 피드 제공

4. 채팅 시스템

- WebSocket 기반 실시간 채팅
- 로그인/비로그인 사용자 모두 참여 가능
- 익명 채팅 기능

5. 개인화 기능

- MyPage 를 통한 개인 정보 관리
- 주식 포트폴리오 관리
- 수익률 기반 랭킹 시스템

4. 설치 및 사용방법

[백엔드 설치]

```
cd hambookstocks.v2
```

```
pip install -r requirements.txt
```

```
uvicorn main:app --reload
```

[프론트엔드 설치]

```
cd front-react
```

```
yarn install
```

```
yarn start
```

[환경 설정]

- MySQL 데이터베이스 설정
- Redis 서버 설정
- OpenAI API 키 설정
- 환경 변수 설정 (.env 파일)

5. 소개 영상 또는 시연영상

홈화면



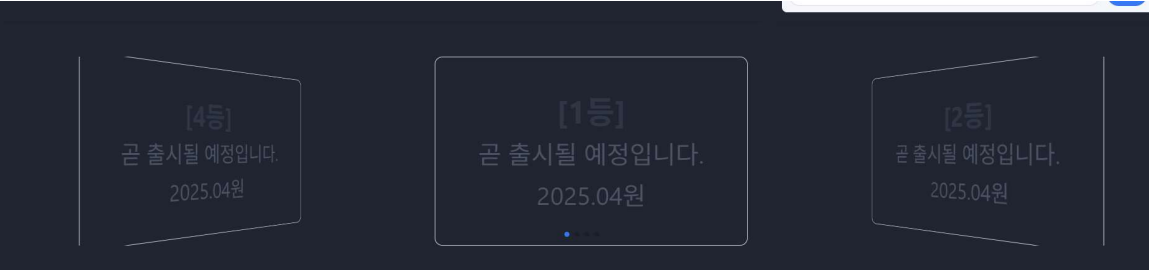
로그인 된 화면



마이페이지



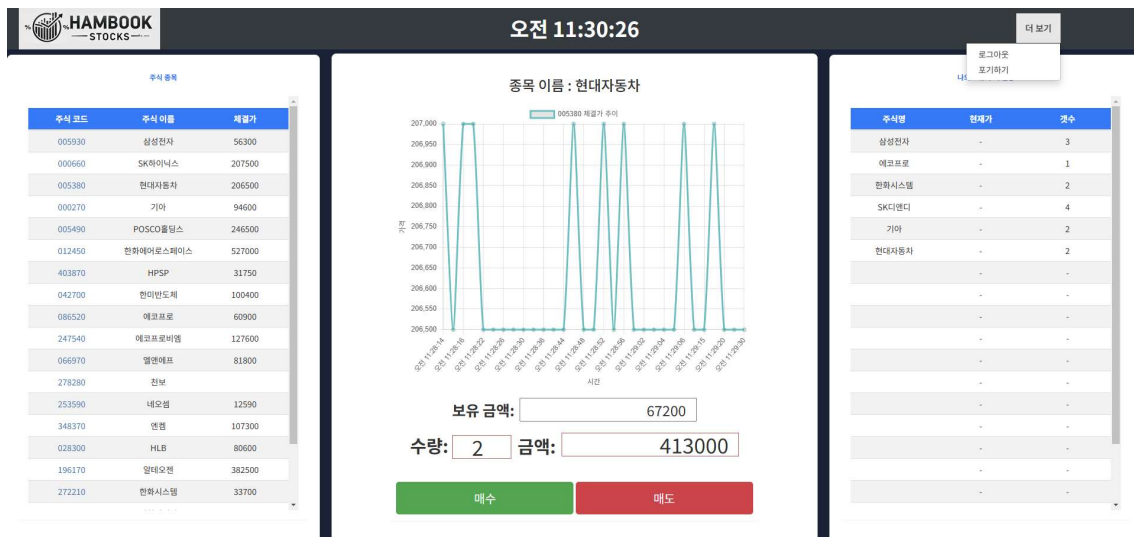
명예의 전당



실시간 종목

주식명	현재가	갯수
삼성전자	-	-
SK하이닉스	-	-
현대자동차	-	-
기아	-	-
POSCO홀딩스	-	-
한화에어로스페이스	-	-
HPSP	-	-
한미반도체	-	-
에코프로	-	-
에코프로비엠	-	-
엘앤에프	-	-
천보	-	-
네오셈	-	-
엔켐	-	-

실시간 매도/매수 화면



6. 팀 소개

개발팀 구성

하승원 : Frontend 개발

김민찬 : Backend 개발

노현민 : 기획

김다영 : 디자인

7. 참여후기

하승원

이번 프로젝트를 통해 React 를 활용한 프론트엔드 개발에 대한 깊은 이해를 얻을 수 있었습니다. 특히 실시간 데이터 처리와 사용자 인터페이스 설계 과정에서 많은 도전과 성장의 기회를 경험했습니다. 팀원들과의 협업을 통해 다양한 관점에서 문제를 해결하는 방법을 배웠고, 이는 향후 개발자로서의 성장에 큰 도움이 될 것입니다.

김민찬

FastAPI 를 활용한 백엔드 개발을 통해 현대적인 웹 API 설계 방법을 학습할 수 있었습니다. 특히 JWT 인증, Redis 캐싱, WebSocket 을 통한 실시간 통신 등 다양한 기술을 실제 프로젝트에 적용해보면서 실무적인 개발 경험을 쌓을 수 있었습니다. 데이터베이스 설계와 비즈니스 로직 구현 과정에서 많은 인사이트를 얻었습니다.

노현민

프로젝트 기획자로서 사용자 관점에서 서비스를 바라보는 시각을 기를 수 있었습니다. 실제 사용자들의 니즈를 파악하고, 이를 기술적으로 구현 가능한 요구사항으로 변환하는 과정에서 많은 학습이 있었습니다. 팀원들과의 원활한 소통을 통해 프로젝트가 성공적으로 완성될 수 있었고, 이는 향후 다른 프로젝트에서도 활용할 수 있는 귀중한 경험이 되었습니다.

김다영

사용자 경험과 인터페이스 디자인을 담당하면서 기술적 제약사항과 사용자 편의성 사이의 균형을 맞추는 것의 중요성을 깨달았습니다. 직관적이고 사용하기 쉬운 인터페이스를 설계하기 위해 많은 고민과 시행착오를 거쳤지만, 그 과정에서 UX/UI 디자인의 핵심 가치를 이해할 수 있었습니다. 개발팀과의 협업을 통해 디자인이 실제로 구현되는 과정을 경험할 수 있어 매우 의미 있는 시간이었습니다.

8. 참고문헌 및 출처

기술 문서

FastAPI 공식 문서: <https://fastapi.tiangolo.com/>

React 공식 문서: <https://reactjs.org/>

SQLModel 문서: <https://sqlmodel.tiangolo.com/>

Redis Python 클라이언트: <https://redis.io/docs/clients/python/>

라이브러리

OpenAI API: <https://openai.com/api/>

BeautifulSoup4: <https://www.crummy.com/software/BeautifulSoup/>

PyJWT: <https://pyjwt.readthedocs.io/>

APScheduler: <https://apscheduler.readthedocs.io/>

참고 자료

-주식 거래 알고리즘 설계 참고

-웹 크롤링 기법 참고

-실시간 채팅 시스템 구현 참고