

작품 계획서

작성일	2025/09/21
작품명	대학생 커뮤니티형 모의 투자 웹서비스
팀명	221
팀원	박재성, 배혜관, 정송기



목포해양대학교
MOKPO NATIONAL MARITIME UNIVERSITY

목 차

1. 개요 및 배경	3
2. 기존 사례	4
3. 주요 기능	6
4. 핵심 구현 기술	7
5. 시스템 구성도	9
6. 개발 일정	10
7. 팀원별 역할	11

1. 개요 및 배경

1-1. 프로젝트 개요

실제 투자 시장의 금전적 손실 위험 없이, 대학생들이 안전한 환경에서 실전 투자 경험을 쌓고 금융 지식을 학습할 수 있도록 지원하는 커뮤니티 기반 모의 투자 웹 플랫폼이다.

사용자들은 가상의 자본을 받아 실제 주식 및 암호화폐 시장 데이터에 기반하여 자유롭게 매매하고, 자신의 포트폴리오를 관리하며 수익률 랭킹을 통해 다른 학생들과 경쟁을 펼칠 수 있다.

1-2. 개발 배경

최근 재테크에 대한 사회적 관심이 높아지면서 많은 대학생이 투자에 흥미를 느끼고 있다. 하지만 금융 지식의 부족, 원금 손실의 두려움으로 실제 투자를 시작하기 주저하는 경우가 많다.

그래서 이번 프로젝트에서 이러한 문제의식을 바탕으로, 대학생들이 리스크 없는 가상 환경에서 투자의 전 과정을 체험하고, 투자 전략 게시판을 통해 동료들과 지식을 공유하며 함께 성장할 수 있는 플랫폼을 제공하고자 기획하였다.

2. 기존 사례 및 차별점

2-1. 실제 거래소 (업비트, 키움증권 등)

- 진입 장벽

실제 돈이 오가는 시장이라 투자에 대한 경험이 없는 초보자나 대학생에게는 심리적 부담이 크다. 투자 실패 시 경제적 손실이 바로 발생하기 때문에 소액으로 시작하더라도 부담이 될 수 있다.

- 복잡한 사용자 인터페이스(UI)

전문적인 투자자를 위한 기능들이 많아, 주문 방식이나 차트 분석 도구 등이 초보자에게는 어렵게 느껴질 수 있다. 복잡한 메뉴와 용어들 때문에 기본적인 거래 방법을 익히는 데 시간이 오래 걸릴 것이다.



키움증권의 영웅문 HTS

2-2. 거래소 연동 모의 투자

- 실제 거래소와 동일한 사용성 문제

실제 거래소의 UI/UX를 그대로 모방한 경우가 많아, 복잡하고 직관적이지 않은 인터페이스의 단점을 그대로 가져오기 때문에 쉽게 학습하기 어렵다.

- 커뮤니티 부재

대학생들이 정보를 교환하고 서로 동기를 부여할 수 있는 전용 커뮤니티 기능이 부족하다. 단순히 거래 연습만 할 수 있을 뿐, 투자에 대한 궁금증을 해소하거나 다른 사용자들과 경험을 공유하기가 힘들다.

2-3. 일반 모의 투자 (거래소 연동되지 않은 독립형 서비스)

- 커뮤니티 기능의 한계

대학생을 대상으로 한 커뮤니티 기능이나 소셜 요소가 부족하여, 투자에 대한 흥미를 지속적으로 유지하기 어렵다. 혼자서만 연습하게 되므로 꾸준히 참여할 동기가 약해지고, 학습 효과가 떨어질 수 있다.

2-4. 차별점

- 명확한 타겟층 설정

'대학생'이라는 동질적인 사용자 그룹을 형성하여 신뢰도 높은 커뮤니티 환경을 구축한다.

- 커뮤니티 및 경쟁 요소 강화

개인별/학교별 수익률 랭킹 시스템과 투자 전략 게시판을 통해 학습 동기를 부여하고, 정보 교류를 활성화한다.

- 초보자 친화적 UI/UX

복잡한 기능을 과감히 배제하고, 투자를 처음 접하는 학생들도 쉽고 재미있게 서비스를 이용할 수 있도록 직관적인 디자인을 제공한다.

- 비용이 들지 않음

모든 기능과 투자를 비용 없이 사용할 수 있어서 학생들의 접근성이 높다.

3. 주요 기능

1. 대학 웹메일 기반 인증

소속 학교 웹메일로만 가입이 가능하게 하여 대학생 사용자 인증 및 커뮤니티의 신뢰성을 확보.

2. 시장 현황판 (주식/코인 정보 조회)

국내/해외 주요 주식 및 암호화폐의 현재 시세, 차트 등 기본적인 투자 정보를 실시간으로 조회할 수 있는 대시보드를 제공.

3. 모의 투자 (매수/매도)

실제 시장 데이터를 기반으로 주식과 코인을 매매할 수 있도록 함.

[매매 시스템]

- 모든 사용자에게 초기 자본 1,000만원을 지급.
- 일봉 마감 시점 기준, 총 자산이 초기자본의 10%(100만원) 미만인 경우 퀴즈 참여 가능.
- 퀴즈 통과 시 초기 자본의 20%에 해당하는 200만원의 추가 자금을 지급.
- 퀴즈 참여 기회는 계정당 총 2회로 제한.
- 퀴즈 보상으로 지급된 200만원 자체는 수익률(랭킹) 계산에서 제외.
- 지급금을 통해 얻은 수익에 대해서만 수익률(랭킹)에 반영하여 공정한 경쟁 환경 유지.

4. 개인 포트폴리오

현재 보유 자산 현황, 총 자산 가치, 수익률, 거래 내역 등을 시각적으로 명확하게 확인할 수 있는 개인화된 페이지를 제공.

5. 수익률 랭킹 대시보드

전체 사용자 순위 및 소속 학교 내 순위를 실시간으로 보여주어 건전한 경쟁과 참여를 유도.

6. 매매 전략 게시판

자유롭게 자신의 투자 전략을 공유하거나 다른 사람의 투자 방법에 대해 토론하며 집단지성을 통해 함께 배우는 공간.

7. 금융/투자 콘텐츠

초보자들을 위한 기본적인 투자 용어 설명, 최신 금융 뉴스 등 학습 콘텐츠를 제공.

4. 핵심 구현 기술

1. 프론트엔드: React.js

- 부품(컴포넌트)을 조립하듯 UI를 만들 수 있어 체계적인 개발이 가능하다.
- 관련 자료와 커뮤니티가 방대하여 문제 해결에 용이.
- UI 라이브러리(MUI)로 구글의 디자인 가이드가 적용된 버튼, 입력창 등을 쉽게 만들 수 있다.

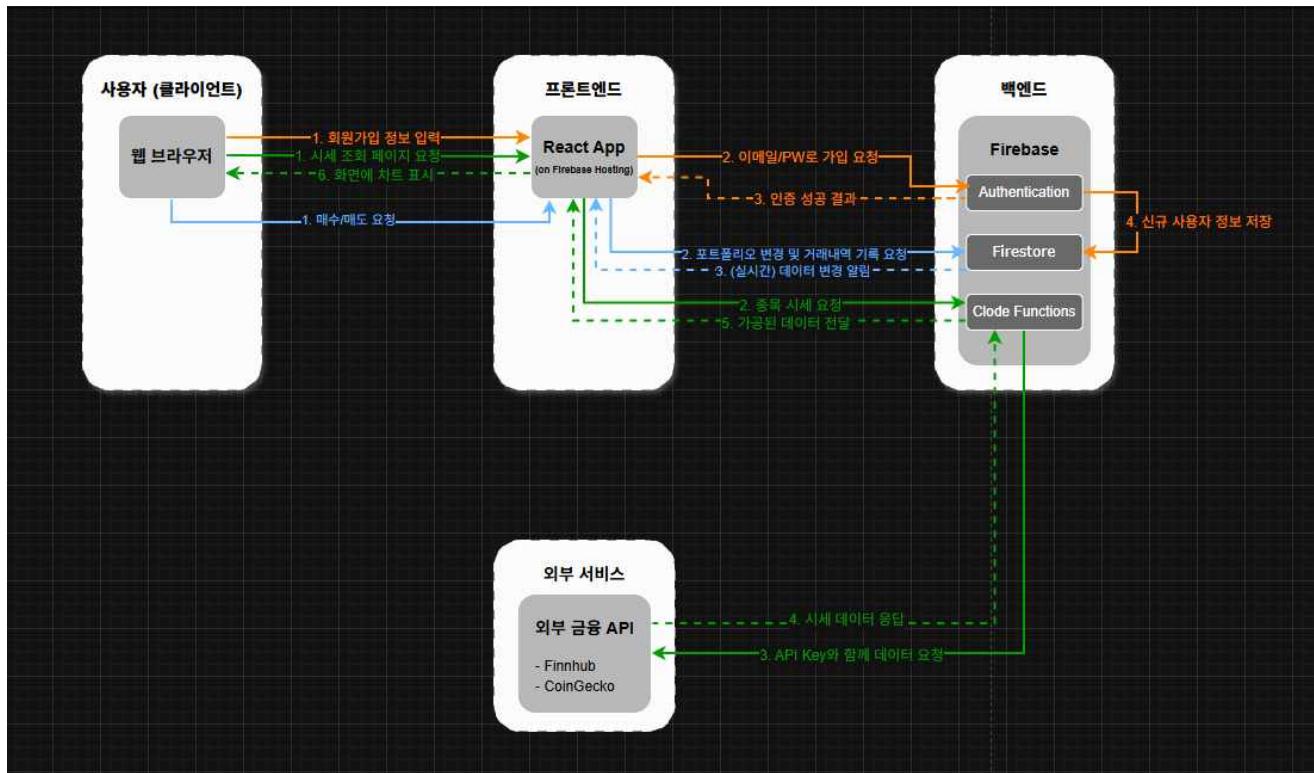
2. 백엔드 & 데이터베이스 (Backend & Database): Firebase

- 회원가입/로그인, 데이터 저장, 서버 기능 등을 매우 간단하게 구현할 수 있게 도와준다.
- Firebase Authentication: 이메일/비밀번호 기반 회원가입 및 로그인 기능을 손쉽게 구현 가능.
- Firestore: 사용자 정보, 포트폴리오, 게시글 등 모든 데이터를 저장하는 NoSQL 데이터베이스.
- Firebase Hosting: 개발 중인 웹사이트를 팀원들과 쉽게 공유하고 테스트 가능.

3. 외부 데이터 연동

- Finnhub.io API, CoinGecko API 이용
- 실제 시장 데이터를 기반으로 현재 시세, 과거 차트 데이터를 실시간으로 제공.
- API 호출은 Firebase Cloud Functions를 통해 이루어져 안전하게 처리.

5. 시스템 구성도 (그림으로 작성)



1. 클라이언트

- 웹 브라우저를 통해 서비스에 접속

2. 프론트엔드

- 사용자가 보는 모든 화면
- React로 만들어짐
- Firebase Hosting을 통해 사용자에게 보여줌

3. 백엔드

- 서버 영역으로, 모든 기능이 Firebase 하나로 통합
- Authentication: 회원가입, 로그인 등 사용자 인증을 담당
- Firestore: 사용자의 자산, 게시글 등 모든 데이터를 저장하는 실시간 데이터베이스
- Cloud Functions: 외부 금융 API를 안전하게 호출하는 중간 다리 역할 (API Key 노출 방지)

4. 외부 서비스

- Finnhub나 CoinGecko 같은 곳에서 실시간 주식/코인 시세 정보를 받아옴

6. 개발 일정

주차	단계	주요 활동
1-2 (9/21)	계획 및 설계	요구사항 분석 기능 명세 확정 UI/UX 뼈대 제작 DB 설계 GitHub 저장소 생성
3 (10/5)	기반 구축	React 프로젝트 생성 Firebase 연동 회원가입/로그인 기능 구현
4-5 (10/12)	핵심 기능 개발1	외부 API 연동 시장 현황판(시세 조회) 및 포트폴리오 화면 UI 개발
6-7 (10/26)	핵심 기능 개발2	모의 투자 기능(매수/매도 로직) 구현 포트폴리오 데이터 연동
8 (11/9)	핵심 기능 개발3	수익률 랭킹 대시보드(퀴즈 시스템 로직 반영) 구현 패자부활전 퀴즈 시스템 (UI 및 로직) 구현
9 (11/16)	커뮤니티 기능 개발	매매 전략 게시판 CRUD(생성, 읽기, 수정, 삭제) 기능 구현
10 (11/23)	수정 / 통합 / 테스트	기능 추가 전체 기능 통합 테스트 버그 수정
11 (11/30)	최종 점검 및 개선	UI/UX 및 기능 개선 최종 코드 리뷰 발표 자료 작성

7. 팀원별 역할

박재성	UI/UX 디자인 구현 데이터 시각화 백엔드와의 통신 로직 구현 퀴즈 화면 UI 개발
정송기	Firestore 데이터 모델링 매수/매도 및 퀴즈 보상 지급 Cloud Function 작성 사용자 데이터 관리
배혜관	Finnhub/CoinGecko API 연동 수익률 기반 랭킹 시스템 로직 구현 (퀴즈 시스템 반영) 전체 기능 통합 테스트 지원