

# 대학생 커뮤니티형

# 모의투자 웹애플리케이션

팀명: 221

- 20183345 정송기
- 20213313 박재성
- 20213317 배예관

# 목 차

01

개요 및 개발 배경

02

기존 사례 및 차별점

03

주요 기능

04

핵심 구현 기술

05

시스템 구성도

06

개발 일정

07

팀원별 역할

개요

# 개요 및 개발 배경

개요	컨셉	<ul style="list-style-type: none"><li>• 대학생을 위한 커뮤니티 기반 모의 투자 웹 플랫폼</li></ul>
	해결	<ul style="list-style-type: none"><li>• 금전적 손실 위험 없이 실전 투자 경험 제공</li><li>• 안전한 환경에서 금융 지식 학습 지원</li></ul>
	주요 기능	<ul style="list-style-type: none"><li>• 가상 자본으로 실제 시장(주식/암호화폐) 데이터 기반 매매</li><li>• 개인 포트폴리오 관리</li><li>• 사용자 간 수익률 랭킹 경쟁</li></ul>

# 개요 및 개발 배경

개발 배경	현황	<ul style="list-style-type: none"><li>• 대학생들의 재테크 관심 증가</li><li>• 그러나, 금융 지식 부족 + 원금 손실 두려움으로 실제 투자 주저</li></ul>
	해결	<ul style="list-style-type: none"><li>• 리스크 없는 가상 투자 환경 제공</li><li>• 투자의 전 과정(매매, 관리) 체험</li><li>• 투자/금융 학습 콘텐츠 제공</li></ul>
	기대 효과	<ul style="list-style-type: none"><li>• 모의 투자를 통한 실전 감각 배양</li><li>• 커뮤니티(게시판)를 통한 지식 공유 및 동반 성장</li></ul>

금융/투자 지식을 실질적으로 향상시킬 수 있는 교육적·실습적 플랫폼을 개발하고자 함.

# 기존 플랫폼 차별점

# 기존 사례

# 실제 거래소 (예: 업비트, 키움증권)

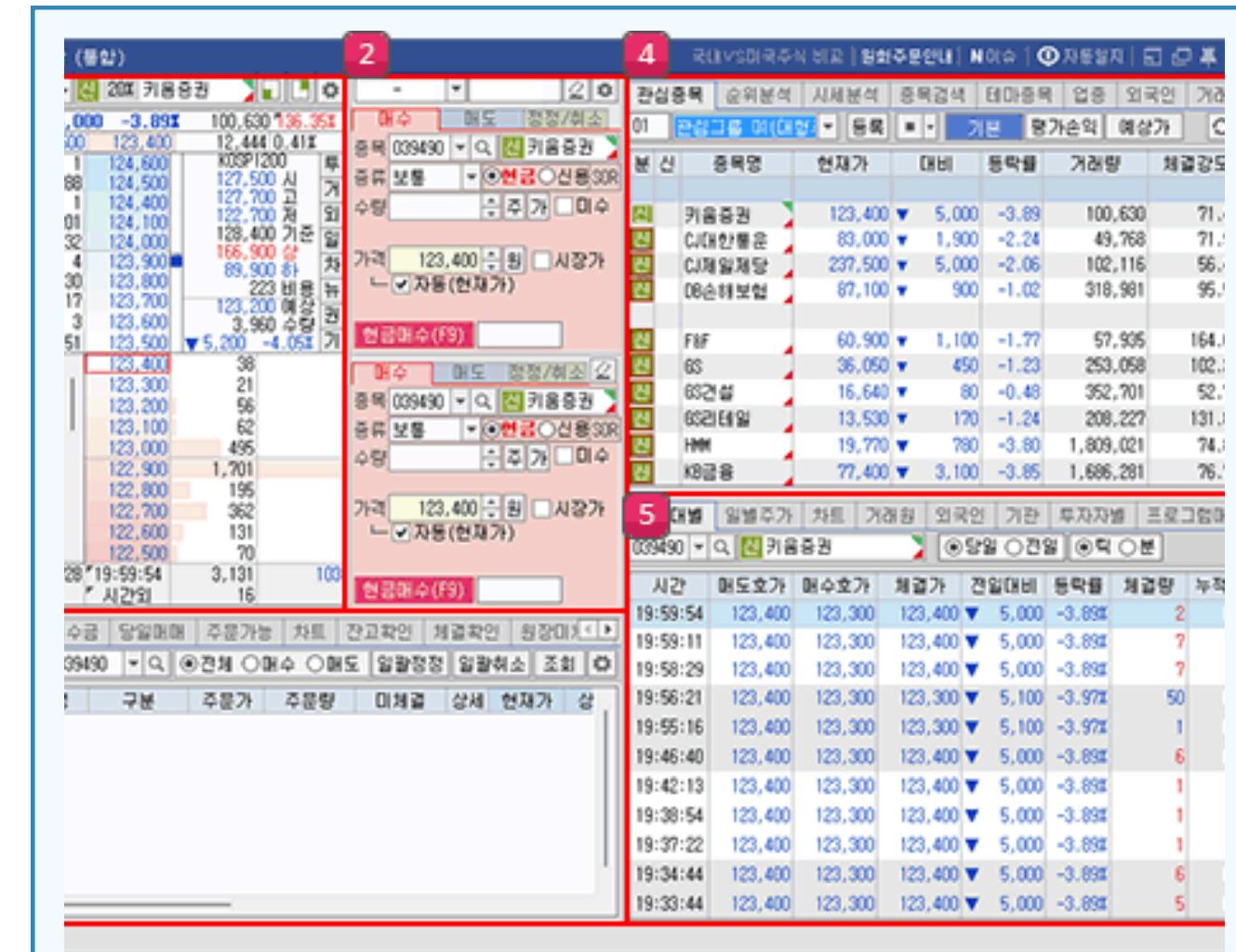
- 진입 장벽: 원금 손실의 심리적 부담
  - 복잡한 UI/UX: 숙련자 위주의 기능 (초보자 학습 어려움)

# 거래도 연동 모의 투자

- 복잡한 UI/UX: 실제 거래소의 복잡한 UI를 그대로 사용
  - 커뮤니티 부재: 정보 교환, 동기 부여를 위한 공간 전무

# 일반 모의 투자

- 커뮤니티 한계: 타겟층(대학생) 특화 커뮤니티 기능 부족
  - 동기 부여 약화: 지속적인 흥미 유발 실패 (학습 효과 저하)



키움증권의 영웅문 HTS

# 기존 차례 및 차별점

## 차별점

명확한 타겟층 설정 (대학생)	<ul style="list-style-type: none"><li>동질적 사용자 그룹을 통한 신뢰도 높은 커뮤니티 환경 구축</li></ul>
커뮤니티 및 경쟁 요소 강화	<ul style="list-style-type: none"><li>수익률 랭킹 시스템</li><li>투자 전략 게시판 활성화 → 강력한 학습 동기 부여</li></ul>
초보자 친화적 UI/UX	<ul style="list-style-type: none"><li>복잡한 전문 기능 배제 → 핵심 기능에 집중</li><li>직관적인 디자인으로 쉽게 학습</li></ul>
비용 부담 없음	<ul style="list-style-type: none"><li>가상 자본으로 실제 시장(주식/암호화폐) 데이터 기반 매매</li><li>개인 포트폴리오 관리</li><li>사용자 간 수익률 랭킹 경쟁</li></ul>

# 주요 기능

## 대학 웹메일 기본 인증

- 소속 학교 웹메일로 가입 → 대학생 사용자 인증
- 커뮤니티 신뢰성 확보

## 미장 현황판 (대기보드)

- 주식, 암호화폐 실시간 시세 및 차트 정보 제공

## 개인 포트폴리오

- 자산 현황, 총자산, 수익률, 거래 내역 시각화

## 금융/투자 학습 콘텐츠

- 초보자를 위한 투자 금융/투자 학습 콘텐츠 제공

# 주요 기능

## 모의 투자 (매수/매도)

- 실제 시장 데이터 기반 매매
- 초기 자본 1,000만원 지급

## 째도전 퀴즈

- 자산 10% (100만원) 미만 시 퀴즈 참여 (총 2회)
- 통과 시 200만원 지급
- 추가 지급액은 랭킹(수익률) 계산에서 제외
- 지급액을 통해 얻은 수익은 랭킹(수익률)에 반영

## 수익률 랭킹 대기보드

- 전체 순위 및 소속 학교 내 순위 실시간 제공
- 견전한 경쟁 및 지속적인 참여 유도

## 매매 전략 게시판

- 투자 전략 공유 → 집단 지성을 통한 동반 학습

# 핵심 구현 기술

## 프론트엔드: React.js

- 컴포넌트 기반 개발 (체계적, 높은 재사용성)
- 방대한 커뮤니티 및 자료 (빠른 문제 해결)
- MUI 라이브러리로 검증된 Google 디자인 가이드(버튼, 입력창 등)를 쉽게 적용

# 핵심 구현 기술

## 백엔드 & DB: Firebase

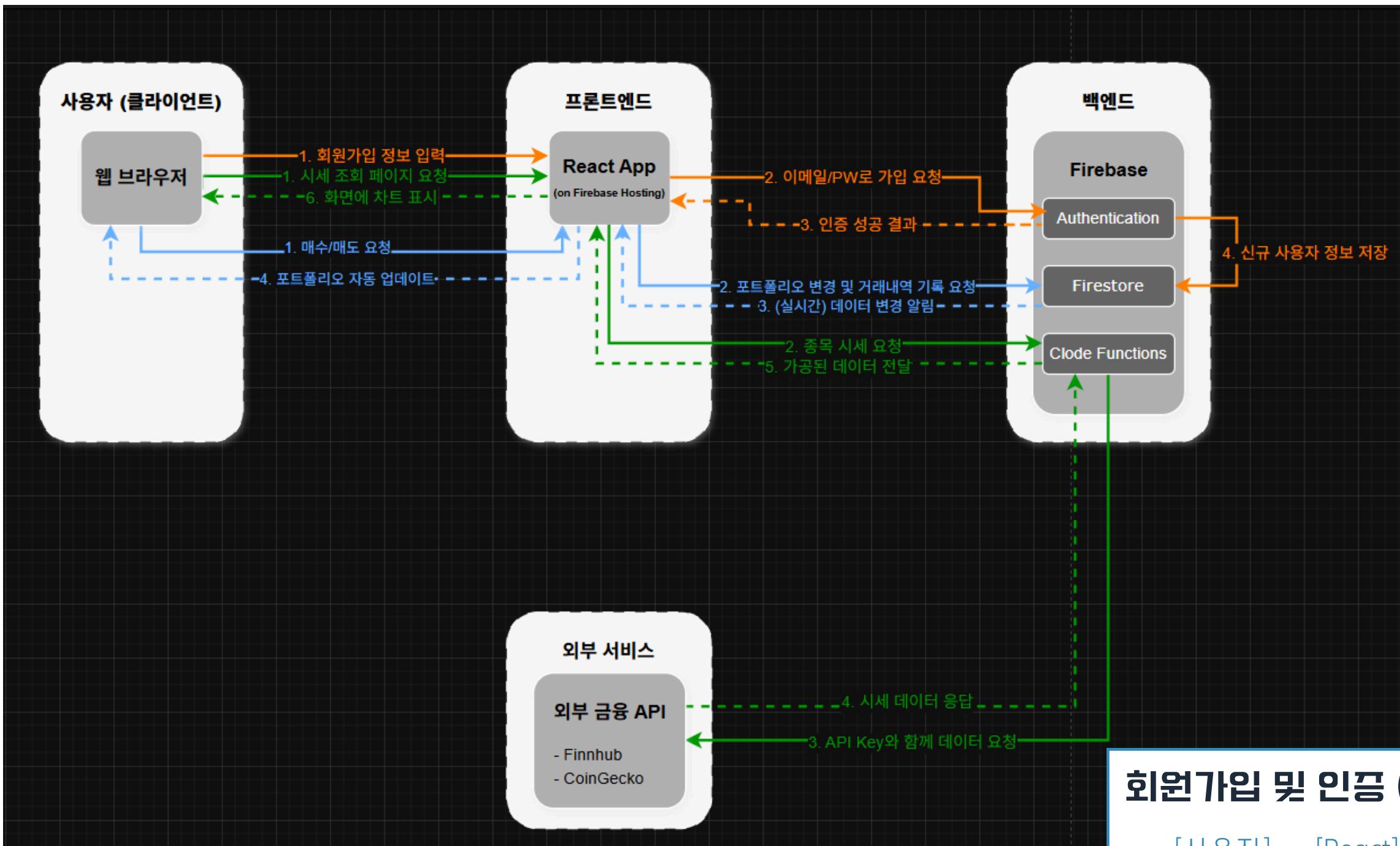
- 서버 구축 없이 핵심 기능(인증, DB)을 간편하고 빠르게 구현
- Authentication: 이메일 기반 회원가입/로그인
- Firestore (NoSQL DB): 사용자 정보, 포트폴리오, 게시글 등 모든 데이터 저장
- Hosting: 팀원 간 테스트 및 공유를 위한 간편 배포

# 핵심 구현 기술

## 외부 데이터 연동

- Finnhub.io API (주식) / CoinGecko API (암호화폐) 사용
- 실제 시장의 실시간 시세 및 과거 차트 데이터 확보
- Firebase Cloud Functions로 API 호출을 서버(백엔드)에서 안전하게 처리

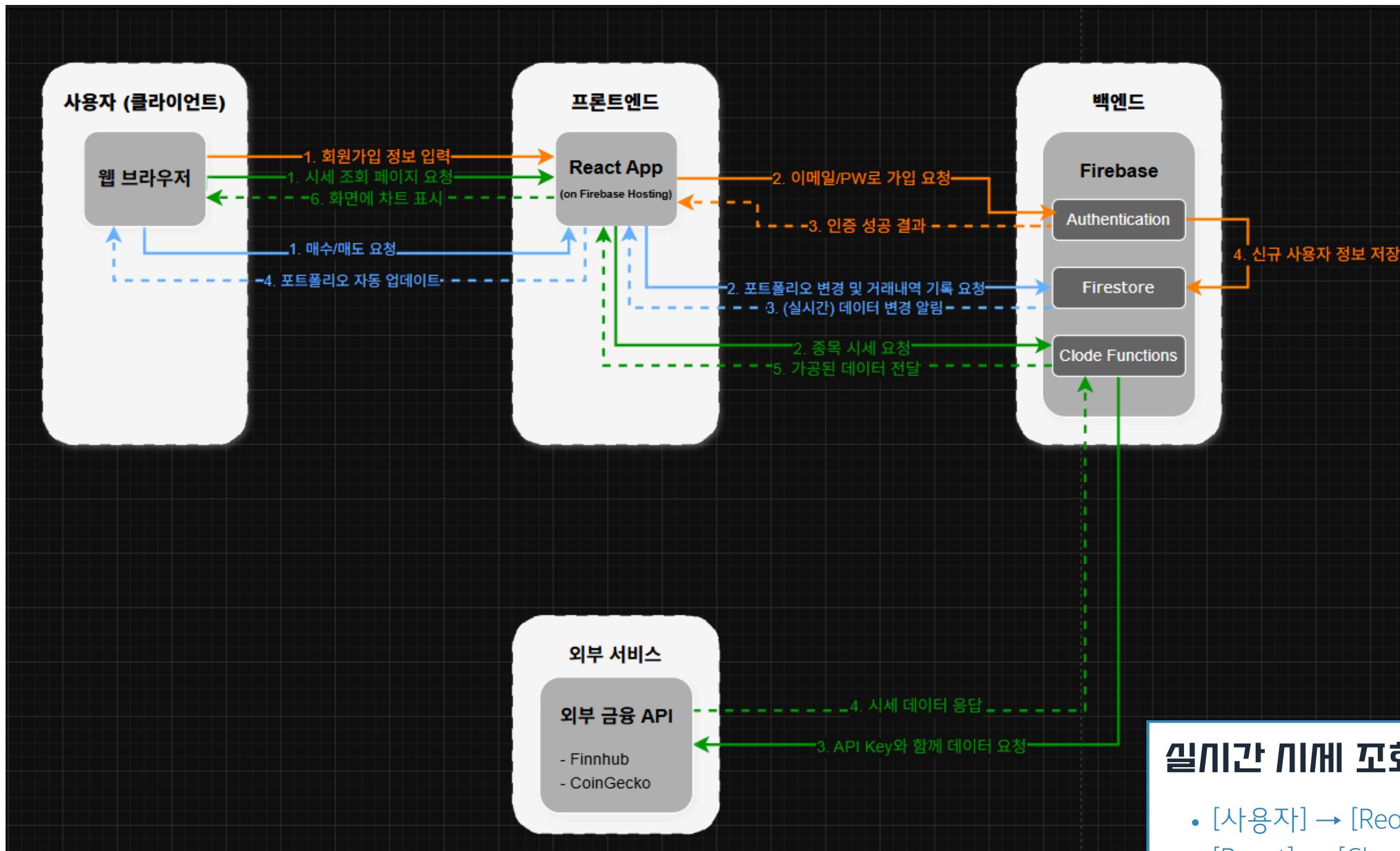
# 프로젝트 구성도



## 회원가입 및 인증 (주황선)

- [사용자] → [React] : 회원가입 정보(이메일/PW) 입력
- [React] → [Firebase Auth] : 인증(가입) 요청
- [Firebase Auth] → [Firestore] : 인증 성공 시, 신규 사용자 정보를 DB에 저장

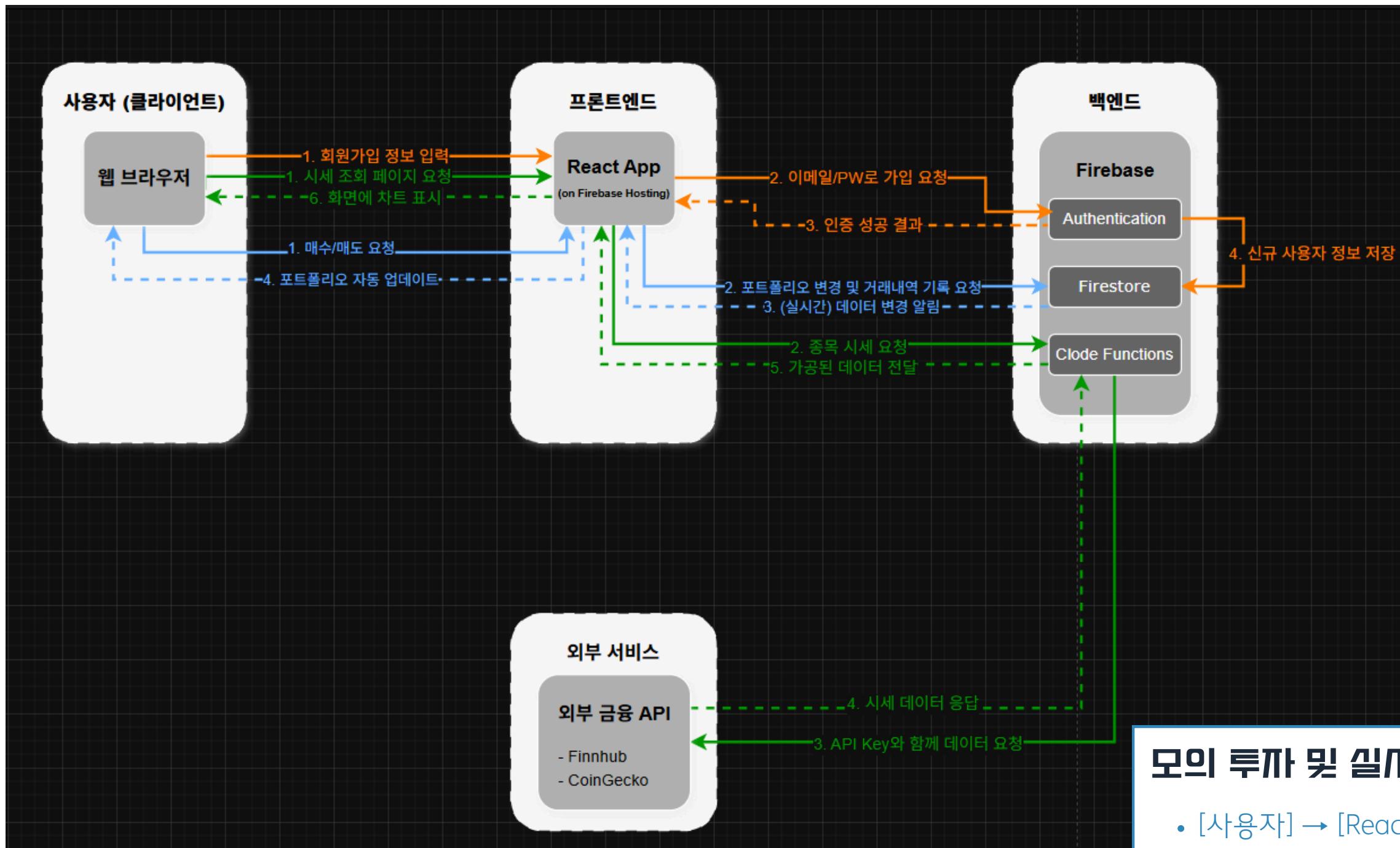
# 프로그램 구성도



## 실제간 실제 모회 (조록선)

- [사용자] → [React] : 주식/코인 시세 페이지 요청
- [React] → [Cloud Functions] : 종목 데이터 요청
- [Cloud Functions] → [외부 API] : (보안) API Key와 함께 데이터 요청
- [외부 API] → [Cloud Functions] : 시세 데이터 응답
- [Cloud Functions] → [React] : 가공된 데이터를 React 앱으로 전달
- [React] → [사용자] : 화면에 차트 및 시세 표시

# 프로그램 구성도



## 모의 투짜 및 실시간 동기화 (파란색)

- [사용자] → [React] : 매수 또는 매도 요청
- [React] → [Firestore] : 포트폴리오 변경 및 거래 내역을 DB에 기록
- [Firestore] → [React] : (실시간) DB 변경 사항을 React 앱에 즉시 알림
- [React] → [사용자] : 포트폴리오 자동 업데이트

## 개발 일정

# 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 (10/05)	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 (11/09)	핵심 기능 개발3	수익률 랭킹 대시보드(퀴즈 시스템 로직 반영) 구현   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

# 팀원별 역할

팀원	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

# 감사합니다