

# 초보 투자자를 위한 모의투자 결과 AI분석·이론 연계 학습 서비스: Qbit

Team 22 Curihous 정유정, 노현선

## 1. Team info

- 과제명: 초보 투자자를 위한 모의투자 결과 AI 분석·이론 연계 학습 서비스
- 팀 번호·팀 이름: 22-큐리호스
- 팀 구성원
  - 정유정(2276288): 팀장, 백엔드 및 프론트엔드 개발, AI 피드백 리포트 기능 구현, UI 디자인
  - 노현선(2217010): 프론트엔드 개발, AI 종목 칼럼 생성 기능 구현
- 과제 키워드: 초보 투자자, 모의투자, AI 기반 피드백 리포트, RAG 기반 맞춤형 이론 학습 콘텐츠
- 과제 내용 요약: 금융 및 디지털 금융 이해도가 낮은 초보 투자자를 위해, 이론 학습과 투자 경험을 통합적으로 제공하는 서비스로, 기초 학습 콘텐츠, AI 기반 종목 칼럼, 모의투자, AI 기반 피드백 리포트를 통해 사용자가 자신의 투자 역량을 점진적으로 키울 수 있도록 설계되었다. 사용자는 투자 성향 진단 → 모의투자 → AI 피드백 → 맞춤형 학습 → 재진입의 반복 루프 구조를 통해 투자 원리를 체계적으로 익히고, 시장 변화에 대응하는 실전 감각을 자연스럽게 길러갈 수 있다.

## 2. Project Summary

### 1) 문제 정의

최근 청년층을 중심으로 자산 관리에 대한 관심이 급증하며 주식 투자가 대중화되었다. 전국 19~34세 청년의 76.5%가 저축 및 투자를 하고 있으며<sup>1</sup>, 주식 거래 계좌 수는 2024년 기준 7,118만 개를 돌파하며 사상 최대치를 기록했다.<sup>2</sup>

그러나 투자 접근성이 높아진 만큼, 기초 금융 이해도와 정보 분석 역량의 부족이 뚜렷한 문제로 나타나고 있다. 한국의 디지털 금융 이해력은 OECD 평균(55점)에 미치지 못하는 43점 수준<sup>3</sup>이며, 초보 투자자의 약 90%가 투자 경력 3년

---

<sup>1</sup> 2024년 청년금융 실태조사, 서민금융진흥원

<sup>2</sup> "지금 이럴 때가 아니야...개미들은 계좌부터 뺐다", 한국경제, 2024.12.07

<sup>3</sup> "'디지털 강국 코리아' 국민 디지털 금융 이해력은 'D학점'", 세계일보, 2024.03.07

미만으로 보고된다. 이들은 투자 관련 정보의 대부분을 개인 방송(33%)이나 커뮤니티(18%)에서 얻고 있으며<sup>4</sup>, 공신력 있는 데이터 기반 분석이나 체계적 학습 환경은 거의 경험하지 못하고 있다.

특히 미장 투자자의 경우, 해외 뉴스가 영어로만 제공되고 기업명이나 용어가 생소하여 시장 이슈의 맥락을 파악하기 어렵다는 정보 접근성의 한계를 겪고 있다. 단순 번역 뉴스만으로는 각 종목의 연관성이나 투자 의미를 이해하기 힘들어, 결국 단편적 이슈 소비나 감정적 매매 결정으로 이어지는 사례가 많다.

또한 기존 투자 교육 서비스는 ‘이론 학습’ 또는 ‘모의투자 체험’ 중 하나에 치우쳐 있으며, 실제 투자 데이터 기반으로 AI가 개인의 판단을 분석하고 학습으로 연결하는 구조는 부재하다. 이로 인해 초보 투자자들은 어떤 부분이 잘못된 의사결정이었는지 알기 어렵고, 투자 결과를 학습으로 이어갈 피드백 루프를 구축하지 못한다.

따라서, 본 서비스는

- ① 초보 투자자가 실전 투자 데이터와 AI 분석을 통해 자신의 판단을 복기하고,
- ② 해외 시장 뉴스로부터 인사이트를 습득하여,
- ③ 다시 이론 학습으로 연결되는 선순환 학습 환경을 구축하는 것을 목표로 한다.

## 2) 기존 연구와의 비교

### 2-1. 유사 서비스 현황 및 한계

현재 투자 서비스 시장은 실제 거래를 지원하는 증권사 MTS와 투자 정보 커뮤니티 앱으로 양분된다. 이 중 Qbit과 같이 초보 투자자의 '학습'을 돕는 직접 경쟁사로는 **마루랩, Vestin, 알투플러스**를 꼽을 수 있다. 모의투자 기능이나 커뮤니티를 제공하는 **증권플러스, Stok'er, 뉴드림** 등은 간접 경쟁 관계로 분류한다.

다음의 경쟁사 분석을 통해 시장의 뚜렷한 한계를 파악할 수 있었다.

- **자기주도적 복기 습관을 형성할 기회의 부재:** Qbit을 제외한 직접, 간접 경쟁사는 사용자가 자신의 투자를 되돌아볼 수 있는 체계적인 기능을 제공하지 않는다. 투자 매매 결과에 대한 개인 맞춤형 분석 피드백이 없을 뿐만

---

<sup>4</sup> ["2030 토스 설문 응답자 90% “주식투자 지속 혹은 확대할 것”, 토스피드, 2021.01.25](#)  
[2022 MZ세대 투자인식 보고서 - ESG·임팩트 투자 편, 대학내일20대연구소](#)

아니라, 투자 과정을 기록하고 성찰하는 데 필수적인 거래 일지 기능 자체가 부재하여 자기주도적 복기 습관을 형성하기 어렵다.

- **단절된 학습 경험 제공:** 마루랩, Vestin 등은 모의투자 기능을, 알투플러스, 뉴드림은 교육 콘텐츠를 제공하지만, 이 기능들은 서로 유기적으로 연결되지 않는다. 결국, 모의투자 경험이 다음 학습으로 이어지지 않는 단절된 구조를 가지고 있다. 이는 초보 투자자가 자신의 잘못된 투자 습관을 개선하지 못한 채 같은 실수를 반복하게 만드는 원인이 된다.

## 2-2. 본 서비스의 차별성

Qbit은 기존 서비스들이 해결하지 못하는 시장의 한계를 파고들어, 초보 투자자의 성장에 초점을 맞춘 다음과 같은 차별점을 가진다.

- **AI 피드백과 거래 일지를 통한 자기주도적 복기 습관 형성:** 경쟁사와 달리 Qbit은 사용자가 자신의 투자를 되돌아볼 수 있는 2가지 기능을 제공한다. AI는 모의투자 결과를 차트 데이터와 뉴스 정보를 종합해 개인 맞춤형 피드백을 제공하고, ‘거래 일지’ 기능을 통해 사용자가 스스로 매매를 돌아보며 꾸준한 투자 습관을 형성하도록 돕는다.
- **학습과 실전을 연결하는 선순환 구조 제공:** Qbit은 단절된 학습 경험을 제공하는 경쟁사들과 달리, 모의투자 경험을 다음 이론 학습으로 직접 연결하는 유기적인 선순환 구조를 완성했다. AI 피드백을 통해 파악된 개선점을 보완할 수 있는 맞춤형 학습 콘텐츠를 추천함으로써, 사용자가 같은 실수를 반복하지 않고 체계적으로 성장할 수 있는 통합적인 학습 환경을 제공한다.

## 3) 제안 내용

본 과제는 초보 투자자의 낮은 금융 이해도와 불확실한 투자 의사결정 문제를 해결하기 위해, **모의투자 결과와 AI 분석을 연계한 투자 학습 서비스**를 제안한다. 단순히 이론을 학습하거나 모의투자를 수행하는 것에서 그치지 않고, “모의투자 → AI 기반 피드백 → 맞춤형 학습 추천”이라는 순환 구조를 통해 이론 학습과 시뮬레이션 기반 투자 실행 과정을 자연스럽게 연결한다. 사용자는 자신의 모의투자 거래 기록을 기반으로 AI가 생성한 피드백 리포트를 확인하고, 그 리포트에서 제시하는 추천 학습 모듈을 학습함으로써 투자 이해도를 점진적으로 향상시킬 수 있다. 이를 통해 초보 투자자가 실질적 성과 개선과 금융 지식 축적을 동시에 달성할 수 있도록 돕는다.

## 4) 기대 효과 및 의의

본 서비스는 다음과 같은 세 가지 기대 효과를 가진다.

1. **초보 투자자의 건강한 투자 습관 형성:** AI 피드백과 거래 일지를 통한 자기주도적 복기 경험은 초보 투자자가 감이나 소문에 의존하는 투기적 습관에서 벗어나, 데이터와 자신만의 원칙에 기반하여 투자하는 건강한 습관을 형성하도록 돕는다. 이는 장기적으로 안정적인 투자자로 성장하는 발판이 될 것이다.
2. **투자 교육 시장의 새로운 패러다임 제시:** 일방적인 이론 전달이나 모의투자 경험 제공에 그쳤던 기존 시장에, '학습-실전-피드백'이 유기적으로 연결되는 통합적인 교육 모델을 제시한다. 이는 사용자의 실질적인 실력 향상을 유도하며, 데이터 기반의 개인화된 금융 교육이라는 새로운 시장을 개척할 수 있다.
3. **금융 이해력 증진을 통한 사회적 비용 감소:** 검증되지 않은 정보에 의존한 '묻지마 투자'로 인해 발생하는 개인의 금전적 손실과 사회적 비용을 줄이는 데 기여할 수 있다. 사용자들이 Qbit을 통해 체계적으로 학습하고 신중하게 투자하는 경험을 쌓음으로써, 전반적인 금융 이해력 증진에 긍정적인 영향을 미칠 것이다.

본 프로젝트의 핵심 의미는 단기적인 수익률이 아닌, 장기적으로 성공하는 투자자의 '지속 가능한 투자 습관'을 형성하는 데 있다. AI 피드백과 거래 일지는 사용자가 스스로의 투자를 복기하고 성찰하는 '자기주도적 학습 태도'를 길러준다. 이는 투자의 대중화라는 양적 팽창 속에서 소외되었던 '질적 성장'의 필요성을 해결하고자 하는 시도이며, 기술(AI)을 활용하여 개인화된 교육을 제공함으로써 금융 교육의 새로운 방향성을 제시한다.

## 5) 주요 기능 리스트

본 과제는 다음과 같은 핵심 기능으로 구성되어 있다.

### 1. 모의투자 기능

본 서비스는 사용자가 실제 자본 손실의 위험 없이 주식 거래 과정을 경험할 수 있도록 모의투자 환경을 제공한다. 사용자는 종목을 검색하고, 매수·매도 거래를 수행하며, 포트폴리오를 관리할 수 있다. 거래 내역은 자동으로 저장되어 월별 거래 일지 형태로 확인 가능하며, 이를 통해 초보 투자자가 투자 절차와 시장 동작 원리를 체험적으로 이해할 수 있다.

### 2. AI 기반 피드백 리포트 기능

사용자가 수행한 모의투자 기록을 기반으로 투자 피드백 리포트를 자동 생성한다. 거래 로그를 활용해 수익률, 거래 빈도, 변동성 등의 주요 지표를 계산하고, 이를 바탕으로 규칙 기반 분석을 통해 투자 과정에서 나타난 문제점을 도출한다. 이렇게 정량적으로 분석된 결과는 대규모 언어 모델(LLM)을 활용한 요약 과정을 거쳐, 사용자가 이해하기 쉬운 서술형

리포트로 제공된다. 이를 통해 단순 수치가 아닌 투자 습관과 개선 방향이 제시되며, 초보 투자자가 자신의 행동을 객관적으로 성찰할 수 있도록 돕는다.

### 3. RAG 기반 맞춤형 이론 학습 추천 기능

피드백 리포트의 핵심 주제와 태그를 기반으로 관련 학습 자료를 자동 추천한다. AI 리포트에서 추출된 키워드가 검색·선별(RAG) 파이프라인을 거쳐 투자 이론 카드 데이터베이스와 연결되고, 그 결과 사용자는 자신의 투자 경험과 직접적으로 연관된 학습 콘텐츠를 제공받게 된다. 예를 들어, 리포트에서 ‘손절 지연’이 문제로 제시되면 위험 관리 이론 카드가, ‘기술적 지표 활용 부족’이 드러나면 RSI나 이동평균선 관련 이론 카드가 추천된다. 이를 통해 사용자는 추상적인 개념 학습이 아니라 자신의 경험에서 도출된 구체적 필요를 즉시 보완할 수 있다.

### 4. AI 종목 칼럼 생성 기능

복잡한 해외 금융 뉴스를 초보 투자자가 이해할 수 있는 한국어 칼럼으로 자동 변환하여 제공한다. 핵심 종목(S&P 100, Nasdaq 100 종목) 169개를 대상으로 매일 최신 뉴스를 수집하고, 웹 크롤링과 TextRank 요약을 거쳐 핵심 내용을 추출한 후, 대규모 언어 모델(GPT-4o mini)을 활용해 700~1000자 분량의 친절한 설명체 칼럼으로 생성한다. 생성된 칼럼은 사용자의 포트폴리오와 종목 간 상관관계를 기반으로 개인화 추천되며, 직접 뉴스가 없는 종목은 상관종목의 뉴스를 활용하는 2-Pass 전략을 통해 칼럼 생성률을 높인다.

## 3. Project Design & Implementation

### 3.1 요구사항 정의

본 서비스 Qbit은 초보 투자자가 이론 학습과 실전 모의투자를 연계하여 투자 원리를 학습할 수 있도록 설계된 AI 기반 학습 플랫폼이다. 단순한 투자 시뮬레이터를 넘어, 사용자의 매매 결과를 분석하고 맞춤형 학습 콘텐츠를 추천함으로써 ‘실전-분석-학습’의 순환 구조를 형성한다. 서비스의 전체 흐름은 다음 **STEP 1~4 과정**으로 구성된다.

- **STEP 1. 이론 학습**

초보 투자자가 투자 기초 개념을 체계적으로 익힐 수 있도록 카드형 학습 콘텐츠를 제공한다. 각 카드에는 주제별 태그와 난이도 레벨이 부여되어 있으며, 사용자는 자신의 수준에 맞는 학습 경로를 따라갈 수 있다. 또, 매매 분석 리포트와 연동되어, 사용자가 실제 투자에서 부족했던 개념이나 전략이 자동으로 태깅되어 관련 이론 카드가 추천된다. 이 과정을 통해 사용자는 이론 학습을 단순 암기가 아닌 실전 개선의 일부로 체감할 수 있다.

- **STEP 2. 모의투자**

Alpaca API를 이용한 가상 거래 시스템을 통해 사용자는 실제 주식 시장과 동일한 환경에서 투자 연습을 할 수 있다. 실시간 시세는 Polygon API를 통해 수집되며, WebSocket으로 연결된 Redis Stream을 통해 거래 체결 상태와 포트폴리오 변화가 즉시 반영된다.

- **STEP 3. 맞춤 해외 뉴스**

사용자가 보유한 종목을 기반으로 관련성이 높은 해외 뉴스를 선별하여 초보 투자자 친화형 칼럼 형태로 재가공한다. 이를 위해 pandas 상관계수로 보유 종목의 직접·간접 연관 종목을 찾고, Polygon API로 해당 종목 뉴스 URL을 수집한다. BeautifulSoup으로 본문을 크롤링하여 TextRank로 핵심 내용을 요약하고, 이를 기반으로 ChatGPT가 종목 칼럼을 생성한다. 보유 종목 뉴스가 없으면 상관계수가 높은 종목의 뉴스를 대체로 제공한다.

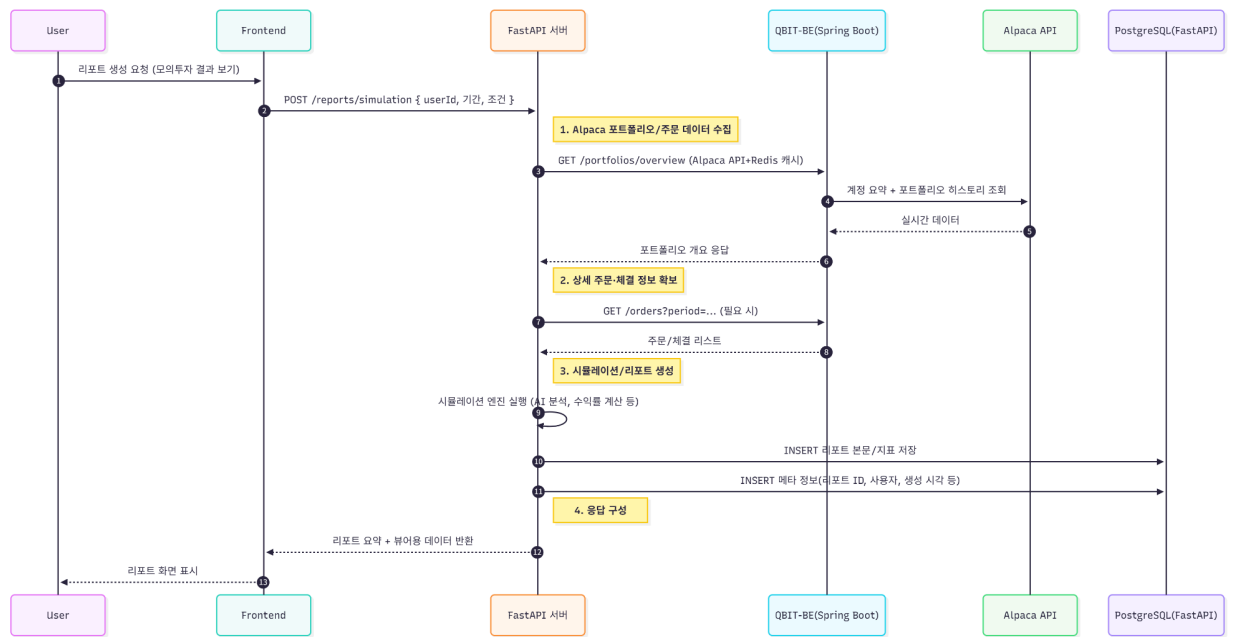
- **STEP 4. AI 피드백 리포트**

사용자의 거래 로그를 기반으로 손익률과 주요 기술 지표를 분석하여 매매 리포트를 자동 생성한다. pandas-ta를 이용해 RI, MACD, 이동평균선, 거래량 등의 지표를 계산하고, 이를 바탕으로 투자 타이밍의 적절성을 판단한다. 이후 ChatGPT API가 분석 결과를 자연어로 요약해 투자 판단 근거와 개선 방향을 제시한다. 생성된 리포트는 Trade Cycle(매수~전량 매도) 단위로 저장되며, 사용자는 이를 통해 자신의 투자 패턴을 복기하고 개선할 수 있다.

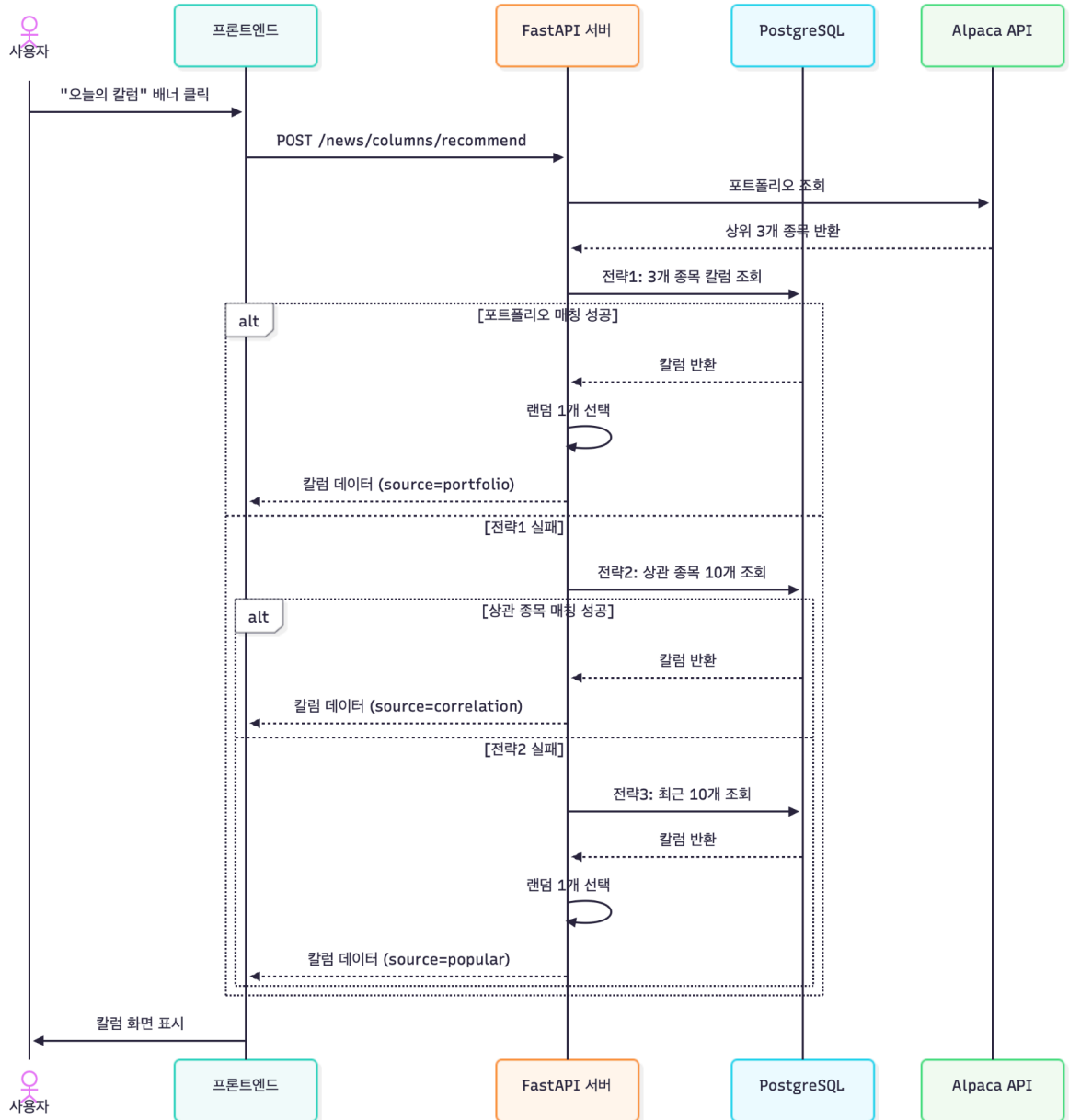
## 3.2 시퀀스 다이어그램

이 순환 구조를 구성하는 기능 중, 가장 중심적인 두 영역(모의투자·리포트 / 컬럼 생성)을 시퀀스 다이어그램으로 정리하였다.

- 모의투자 및 리포트 생성



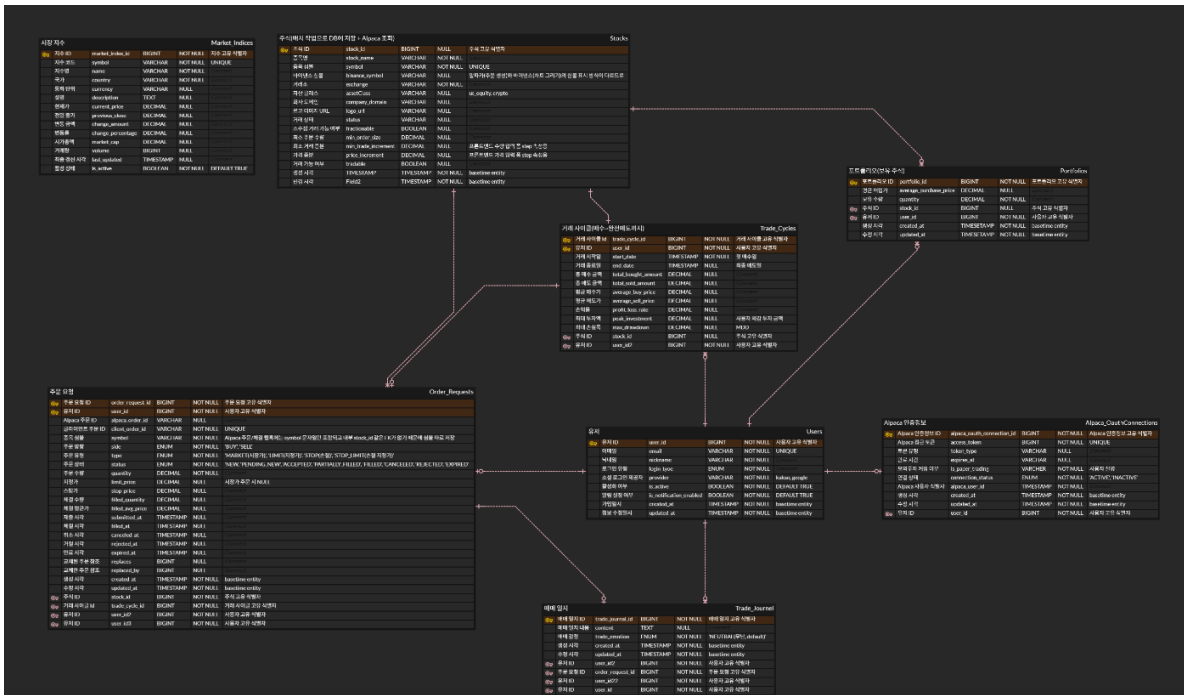
- AI 칼럼 생성 및 추천 기능



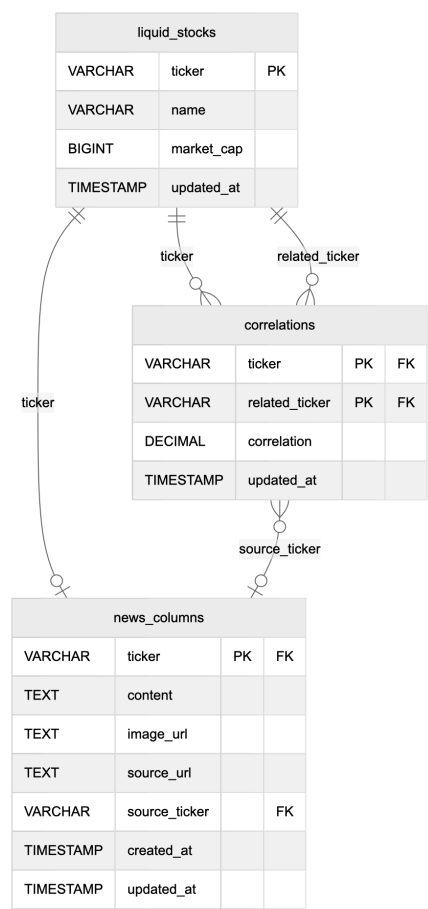
### 3.3 ERD 설계



● 거래 서비스 ERD (Spring Boot)

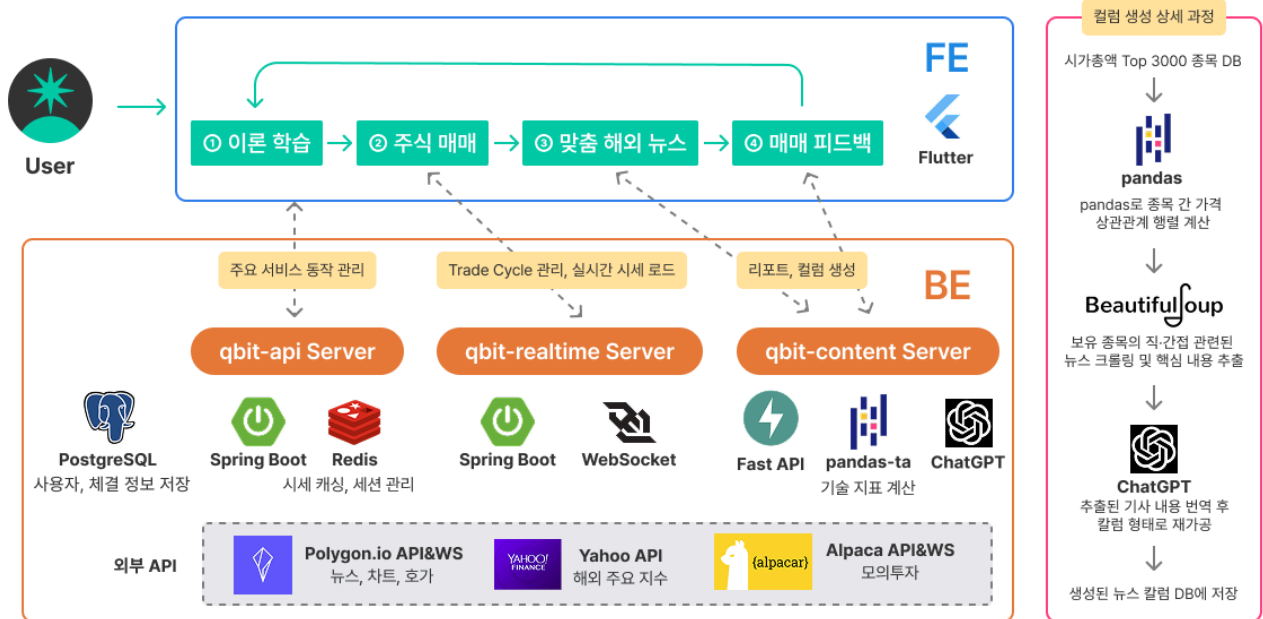


● 콘텐츠 서비스 ERD (Fast API)



### 3.4 전체 시스템 구성

전체 시스템은 사용자, 프론트엔드, 그리고 여러 백엔드 모듈(qbit-api, qbit-realtime, qbit-content)과 데이터베이스로 구성된다. 각 모듈은 역할에 따라 분리되어 있다. qbit-api는 주문 처리와 TradeCycle 관리를 담당하고, qbit-realtime은 Redis Stream 기반의 실시간 체결을 송출하며, qbit-content는 FastAPI를 통해 AI 리포트와 학습 콘텐츠를 생성한다.



#### 1. 사용자 (User)

사용자는 모바일 앱을 통해 투자 학습 및 모의투자를 수행하고, AI가 생성한 매매 리포트와 뉴스 칼럼을 확인한다. 사용자 흐름은 다음과 같다.

- 1) 홈 화면에서 보유 종목 관련 칼럼 조회
- 2) 학습 탭에서 카드형 이론 콘텐츠 학습
- 3) 모의투자 탭에서 종목 검색 및 가상 거래
- 4) 거래 종료 후 자동 생성된 AI 매매 리포트 확인
- 5) 추천 이론 카드 열람

## 2. 프론트엔드 (Flutter)

프론트엔드는 Flutter 기반의 크로스플랫폼 모바일 애플리케이션으로 iOS와 Android 환경을 모두 지원한다. 사용자는 이 앱을 통해 실시간 시세 확인, 주문 체결, 학습 진행, 리포트 조회 등 모든 주요 기능을 수행한다.

### [주요 기능]

- 이론 학습: 태그·레벨을 기반으로 카드형 학습 콘텐츠를 표시한다.
- 모의투자: 실시간 시세 차트, 주문창, 호가창, 거래 내역을 제공한다. 주문 시 백엔드 서버(Spring Boot)와 REST API를 통해 거래 요청을 전달하며, 체결 상태는 WebSocket을 통해 실시간으로 반영한다.
- AI 거래 피드백 리포트: Trade Cycle 단위로 생성된 AI 매매 리포트를 표시한다. 손익률, 기술지표 분석 결과, GPT 기반 피드백 문장을 시각적으로 확인할 수 있다.
- AI 종목 칼럼: 사용자가 보유한 종목과 관련된 해외 뉴스 칼럼을 확인할 수 있으며, 초보 투자자 친화형 요약문과 핵심 포인트를 함께 제공한다.

### [기술 스택]

Flutter, Riverpod(State Management), Dio(REST API 통신), WebSocket(실시간 시세 반영)

## 3. 백엔드 서버 (Spring Boot)

백엔드 서버는 거래 관리 및 시세 데이터 처리의 핵심 허브 역할을 수행한다. 사용자 세션 관리, 모의투자 요청 처리, 실시간 시세 캐싱, AI 서버 간 데이터 연동을 담당한다.

### [주요 기능]

- 사용자 및 포트폴리오 관리: 사용자 로그인, 계좌 정보, 보유 종목 및 거래 내역을 관리한다.
- 거래 처리 및 시세 반영: Alpaca API를 통해 매수·매도 요청을 수행하고, Polygon API의 실시간 시세를 Redis Stream으로 캐싱하여 WebSocket으로 프론트엔드에 전달한다.
- 리포트 및 분석 결과 관리: FastAPI 콘텐츠 서버에서 생성된 매매 리포트 및 뉴스 칼럼 데이터를 수신하고, 이를 PostgreSQL에 저장한다.

### [기술 스택]

Java, Spring Boot, Redis, WebSocket, AWS EC2, PostgreSQL

#### 4. AI 콘텐츠 서버 (FastAPI)

AI 서버는 매매 피드백 리포트 및 뉴스 칼럼 생성을 담당한다.

[주요 기능]

- AI 매매 리포트 생성: Trade Cycle 단위 거래 로그를 분석하여 pandas-ta로 기술지표를 계산하고, ChatGPT API를 통해 투자 판단 근거와 개선 방향을 포함한 자연어 리포트를 자동 생성한다.
- AI 종목 칼럼 생성: pandas 상관계수를 기반으로 보유 종목과 연관된 해외 뉴스를 탐색하고, Polygon API·BeautifulSoup·TextRank·ChatGPT API를 활용해 핵심 내용을 요약한 칼럼 형태로 재가공한다.

[기술 스택]

Python, FastAPI, pandas, pandas-ta, BeautifulSoup, TextRank, OpenAI API

### 3.5 주요 엔진 및 기능 설계

본 프로젝트 Qbit은 초보 투자자의 투자 학습을 돕기 위해, 모의투자 및 AI 리포트 생성 기능과 AI 종목 칼럼 생성 기능을 핵심 소프트웨어 모듈로 구현하였다. 각 엔진은 FastAPI와 Spring Boot를 기반으로 독립적인 처리 파이프라인을 구성하며, 사용자 거래 데이터 및 외부 뉴스 데이터를 분석·가공하여 학습형 콘텐츠를 자동으로 생성한다.

#### 1. 모의투자 및 AI 리포트 생성

##### 1) 목적

모의투자 및 AI 리포트 생성 엔진은 실제 시장 환경과 유사한 조건에서 사용자의 투자 경험을 학습 과정으로 확장하는 것을 목표로 한다. 실시간 시세와 체결 정보를 WebSocket으로 반영해 실제 거래에 가까운 환경을 제공하며, 백엔드 서버에서 거래 사이클 단위로 손익을 계산한다. AI는 계산된 결과를 기반으로 투자 행동을 분석하고, 이해하기 쉬운 피드백과 학습 콘텐츠를 제공함으로써 사용자가 자신의 투자 패턴을 점진적으로 개선하도록 돕는다.

## 2) 구조 개요

Qbit의 모의투자 및 AI 리포트 엔진은 “거래 → 분석 → 피드백 → 학습 추천”의 순환 구조로 구성된다. 사용자는 실시간으로 모의투자를 수행하고, 거래 결과를 기반으로 AI 리포트를 생성하여 자신의 투자 성향과 문제점을 학습 카드와 함께 피드백받는다. 전체 시스템은 실시간 투자 시뮬레이션 파이프라인과 AI 분석 파이프라인의 두 축으로 이루어져 있으며, 두 흐름은 백엔드 모듈 간 비동기 메시징을 통해 긴밀히 연결된다.

실시간 투자 시뮬레이션 파이프라인은 사용자의 주문을 처리하고 시장 데이터를 반영해 체결 결과를 즉시 반영한다. AI 분석 파이프라인은 거래가 완료된 시점에 해당 거래 사이클을 분석하고, 투자 결과에 대한 피드백과 학습 콘텐츠를 자동으로 생성한다.

## 3) 구성 기술

- **Spring Boot (qbit-api, qbit-realtime):** 사용자의 매수·매도 주문을 처리하고 거래 사이클(TradeCycle)을 관리한다. qbit-realtime 모듈은 Redis Stream에서 체결 이벤트를 구독하고, WebSocket을 통해 프론트엔드로 실시간 데이터를 송출한다.
- **Redis Stream:** Alpaca로부터 전달되는 체결 이벤트를 큐잉하며, 비동기적으로 손익 계산 및 리포트 생성 파이프라인에 전달한다.
- **Alpaca API & WebSocket:** 가상 환경에서 주식 주문 및 체결을 수행하고, 계좌 잔고와 포트폴리오를 실시간으로 동기화한다.
- **Binance REST API, Polygon API:** 주식 및 가상자산의 시세, 거래량, 차트 데이터를 수집한다.
- **FastAPI (qbit-content):** 거래 데이터를 분석하고, pandas 및 ChatGPT API를 이용해 AI 리포트를 생성하며, 분석 결과에 따라 학습 콘텐츠를 추천한다.  
**pandas-ta:** 수익률, RSI, MACD, 이동평균선 등 기술적 지표를 계산하고, 거래 패턴을 수치 기반으로 분석한다.
- **OpenAI ChatGPT API:** 정량 분석 결과를 자연어로 해석해 초보 투자자가 이해하기 쉬운 피드백 문장을 생성한다.
- **PostgreSQL:** 주문, 포트폴리오, 거래 사이클, 리포트 데이터를 영속적으로 저장하고, 사용자 단위로 관리한다.
- **Flutter Frontend + WebSocket:** 실시간 체결 정보와 분석 리포트를 시각적으로 표시하며, 사용자의 리포트 요청 이벤트를 백엔드로 전달한다.

#### 4) 내부 모듈 및 역할

##### 1. 모의투자 엔진:

사용자의 주문을 Alpaca Paper Trading으로 전달하고, 체결 결과를 Redis Stream으로 수신한다. 모든 주문 이벤트는 데이터베이스에 기록되며, 동일 종목의 매수와 매도 과정을 하나의 거래 사이클(TradeCycle)로 자동 묶어 관리한다. 이를 통해 사용자는 실전과 유사한 체결 속도와 시장 반응을 경험하며, 반복적인 모의 거래를 통해 매매 패턴을 학습할 수 있다.

##### 2. 거래 사이클 관리 모듈:

매수 시점을 'Cycle Start', 전량 매도 시점을 'Cycle End'로 설정하여 손익률, 매매 횟수, 평균 단가, 투자 기간 등의 핵심 지표를 자동 계산한다. 거래가 종료되면 해당 사이클 데이터를 FastAPI 분석 서버로 전송하여 리포트 생성을 요청한다. 이 과정은 Redis Stream 기반의 이벤트 트리거로 자동화되어 있으며, 거래 종료 시점에 리포트 생성이 비동기적으로 수행된다. 다만 안정성을 위해 backfill용 추가 엔드포인트를 두어, 이벤트 누락이나 분석 지연이 발생한 경우 사용자가 직접 리포트를 재생성할 수 있도록 구현하였다.

##### 3. AI 분석 엔진:

전달받은 거래 데이터를 기반으로 pandas를 통해 수익률 및 주요 지표(RSI, MACD, 이동평균선 등)를 계산한다. 거래 패턴(예: 손절 지연, 과도한 진입, 반등 포착 실패 등)을 규칙 기반으로 감지하고, ChatGPT API를 통해 자연어 피드백을 생성한다. 리포트는 '전체 매매 평가 → 주요 진입 지점별 분석 → 개선 포인트 → 당시 시장 이슈'의 구조로 구성되며, 투자자의 행동적 특징을 요약하는 피드백 문장을 포함한다.

##### 4. 학습 추천 모듈:

AI 분석 엔진의 '전체 매매 평가' 파트에서 추출된 태그 집합을 기반으로 관련 학습 카드를 자동 추천한다. 약 30개의 세분화된 태그가 사전에 정의되어 있으며, 각 태그는 학습 카드 DB와 매핑되어 있다. 추천 알고리즘은 리포트에서 추출한 태그 집합과 학습 카드의 태그 세트를 비교하여 가장 많은 태그가 일치하는 학습 카드를 연결한다. 이 과정은 이미 정교하게 구축된 태그 분류 체계 덕분에 안정적으로 작동하며, 분석 결과와 높은 연관성을 가진 학습 카드가 자동으로 리포트에 포함된다.

##### 5. 리포트 저장 및 조회 모듈:

생성된 리포트는 PostgreSQL에 저장되며, 사용자 ID를 기준으로 분리 관리된다. 사용자는 앱 내에서 실시간으로 생성된 리포트를 확인할 수 있으며, 이후 AI 학습 추천 콘텐츠로 바로 이동할 수 있다.

#### 2. AI 종목 칼럼 생성 및 추천

## 1) 목적

AI 종목 칼럼 생성 엔진은 사용자의 보유 종목과 연관된 해외 뉴스를 자동으로 수집·요약하여 시장 인사이트를 제공하는 것을 목표로 한다. 초보 투자자가 접근하기 어려운 글로벌 금융 뉴스를 AI가 분석·번역·요약함으로써, 사용자는 투자 판단에 도움이 되는 핵심 정보를 빠르게 얻을 수 있다.

## 2) 대상 데이터

본 시스템은 2단계의 종목 데이터를 활용한다.

### 6. Tier 1: 핵심 종목 169개 (칼럼 생성 대상)

AI 칼럼은 핵심 종목 169개를 대상으로 하루 1회 생성된다. 이 핵심 종목은 S&P 100과 NASDAQ 100 지수를 기반으로 선정하였다. 각 종목에서 중복되는 항목들을 제거하여 최종적으로 169개의 종목을 칼럼 생성 대상이 되는 고정 리스트로 설정하였다.

### 7. Tier 2: 유동성 종목 3000개 (상관계수 계산 대상)

Polygon API에서 종목 데이터를 호출하여 시가총액 기준 상위 3000개 종목을 정렬한다. 이 종목들 간의 상관계수를 계산하여 칼럼 조회의 Pass 2 전략에서 관련 종목을 탐색하는 데 활용한다. 이 리스트는 주 1회 갱신된다.

## 3) 도입 기술

- **FastAPI:** 파일 처리와 API 엔드포인트 제공을 담당한다.
- **pandas:** 시가총액 기준 상위 3000개 종목의 상관계수를 계산한다.
- **OpenAI GPT-4o mini:** 계산된 데이터를 기반으로 자연어 형태의 칼럼을 생성한다.
- **BeautifulSoup + httpx:** 웹 페이지를 크롤링해 기사 본문을 수집한다.
- **TextRank:** 크롤링한 기사에서 핵심 문장을 추출해 요약을 구성한다.
- **Polygon.io API:** 뉴스 데이터와 종목 관련 데이터를 수집한다.
- **PostgreSQL:** 생성된 칼럼 데이터를 저장하고 관리한다.

## 4) 전체 시스템 구조

본 기능은 배치 작업과 실시간 작업으로 나뉜다.

1. 배치 작업 (주 1회)
  - Top 3000 유동성 종목 결정
  - 종목 간 상관관계수 계산
2. 배치 작업(매일 새벽 5시)
  - 핵심 종목 169개 대상 칼럼 생성
3. 실시간 작업(사용자 요청 시)
  - 사용자 보유 포트폴리오 기반 추천 칼럼 반환

AI 칼럼 생성 전략: 2-Pass 방식

매일 새벽 5시에 실행되는 칼럼 생성은 '2-Pass 방식'으로 동작한다.

Pass	전략	대상	설명
1	직접 뉴스 검색	핵심 종목 169개	직접 뉴스 일괄 생성 170개 종목별로 최근 1일 이내 핵심 종목의 직접 뉴스가 있는지 확인 (Polygon API의 뉴스 데이터 활용) <ul style="list-style-type: none"> <li>- 존재할 경우, 크롤링-요약-칼럼 생성</li> <li>- 존재하지 않는 경우, 빈칸으로 비워 두고 Pass 2로 이동</li> </ul>
2	간접 뉴스 검색	Pass 1에서 실패한 종목	간접 뉴스로 빈칸 채우기 상관관계수 테이블에서 관련성 높은 종목 20개 조회 상관도 높은 순서대로 Pass 1 성공 여부 확인 <ul style="list-style-type: none"> <li>- 첫 번째로 성공한 칼럼을 복사하여 사용</li> </ul>

Pass 1: 각 종목의 직접 뉴스로 칼럼 생성. 일반적으로 약 120개 성공.

Pass 2: 실패한 종목은 상관도가 높은 관련 종목의 칼럼 재사용. 약 40개 추가 성공.

대형 우량주는 매일 뉴스가 나오지만, 소형주나 거래량이 적은 종목은 최근 뉴스가 없는 경우가 많다. Pass 1만 사용하면 약 50개 종목이 칼럼 없이 빈칸으로 남게 되어 사용자 경험이 저하된다. Pass 2 전략을 도입하면 뉴스가 없는 종목도 상관도가 높은 관련 종목의 칼럼을 활용하여 서비스를 제공할 수 있다. 예를 들어 소형 반도체 기업에 직접 뉴스가 없더라도, 상관도가 높은 NVDA나 AMD의 칼럼을 보여줌으로써 관련 시장 동향을 파악할 수 있다. 또한 Pass 2에서는 새로 칼럼을 생성하지 않고 기존 칼럼을 복사하므로 OpenAI API 비용을 절감할 수 있다. 이를 통해 170개 종목 중 약 95퍼센트에 대해 칼럼을 제공하여 서비스 완성도를 높였다.



## 5) 세부 기능 모듈

AI 종목 칼럼 생성 기능은 (1) 유동성 종목 관리 (2) 상관계수 계산 (3) 기사 크롤링 (4) 텍스트 전처리 및 요약 (5) AI 칼럼 생성 (6) 칼럼 저장 및 조회, 총 여섯 가지 하위 모듈로 구성된다.

### ① 유동성 종목 관리 모듈: LiquidStocksService

- 역할: 시가총액 상위 3000개 종목을 Polygon API에서 가져와 DB 저장
- 사용 기술: [Polygon.io](https://polygon.io) API, PostgreSQL
- 처리:
  - Polygon API에서 약 8000개 티커 목록을 수집한다.
  - 각 티커의 시가총액(market\_cap)을 조회한다. 배치 크기 10개, 병렬 처리로 속도를 최적화한다.
  - 시가총액 기준 내림차순 정렬 후 상위 3000개를 선별한다.
  - PostgreSQL의 liquid\_stocks 테이블에 UPSERT 방식으로 저장한다.
  - 3000개에 포함되지 않는 기존 종목은 삭제한다.
- 출력: PostgreSQL liquid\_stocks 테이블 (ticker, name, market\_cap)
- 갱신 주기: 주 1회

### ② 상관계수 계산 모듈 (CorrealtionService)

- 역할: 3000개 종목 간 상관계수를 계산하여 각 종목당 상위 20개 관련 종목을 저장한다. Pass 2에서 직접 뉴스가 없는 종목을 관련 종목을 찾는 데 활용된다.
- 사용 기술: Polygon.io API, pandas, PostgreSQL
- 처리 내용:
  - 과거 데이터 수집: 각 종목의 과거 90일 종가 데이터를 Polygon API로 수집한다. 세마포어를 사용하여 동시 호출을 8개로 제한하고, Rate Limit 회피를 위해 0.2초 딜레이를 적용한다.
  - 수익률 및 상관계수 계산: pandas DataFrame으로 변환하고(행: 날짜, 열: 티커, 값: 종가), pct\_change() 함수로 일일 수익률을 계산한다. corr(min\_periods=30) 함수로 상관계수를 계산하며, min\_periods=30 설정은 최소 30일의 공통 거래 데이터가 있는 경우에만 계산하도록 하여 상장일 차이나 거래 정지로 인한 날짜 불일치를 대응한다.
  - 필터링 및 저장: 각 종목당 상관계수가 높은 상위 20개만 필터링하여 저장한다.

- 출력: PostgreSQL correlations 테이블 (ticker, related\_ticker, correlation), 총 약 60,000개 (3000 × 20)
- 갱신 주기: 주 1회

### ③ 기사 크롤링 모듈 (ArticleScraper)

- 역할: Polygon API에서 제공한 뉴스 URL에서 기사 본문을 추출한다. Pass 1 칼럼 생성 파이프라인에서 사용된다.
- 사용 기술: httpx, BeautifulSoup
- 입력: 뉴스 기사 URL (Polygon API에서 제공)
- 처리 내용:
  - HTTP 요청 및 파싱: httpx의 AsyncClient로 비동기 HTTP GET 요청을 수행하고(timeout 10초), BeautifulSoup로 HTML을 파싱한다. script, style, nav, header, footer 등 불필요한 태그를 제거한다.
  - 본문 추출 전략: 다중 선택자 전략으로 article, .article-body, .article-content, [itemprop="articleBody"] 등을 순차적으로 시도한다. 실패 시 p 태그 기반 추출을 시도하며, 50자 이상이고 링크가 3개 미만인 단락만 추출한다.
  - 성공 조건 및 실패 처리: 최소 200자 이상 확보 시 성공으로 처리한다. 크롤링 실패 시 Polygon API의 description 필드를 활용한다.
- 출력: 기사 본문 텍스트 (실패 시 None)

### ④ 텍스트 전처리 및 요약 모듈 (TextProcessor)

- 역할: 크롤링한 기사 본문에서 메타 정보를 제거하고 TextRank 알고리즘으로 핵심 문장을 추출한다. ChatGPT API 비용 절감과 요약 품질 향상을 목적으로 한다.
- 사용 기술: summa (TextRank 알고리즘), 정규표현식
- 입력: 크롤링한 기사 본문 텍스트
- 처리 내용:
  - 전처리 단계
    - 정규표현식으로 Read More, Image, Photo, Related 등의 메타 정보 패턴을 제거한다.
    - 연속 공백을 제거한다.
  - 요약 단계

- TextRank 알고리즘으로 핵심 문장을 추출한다. (ratio=0.35)
  - 원문의 35퍼센트 길이로 요약한다.
  - 결과가 150자 미만이면 원문 앞부분 800자를 사용한다.
- 출력: 핵심 문장 요약 (영문, 200~800자)

#### ⑤ AI 칼럼 생성 모듈 (ChatGPTClient)

- 역할: 영문 뉴스 요약을 초보 투자자용 한국어 칼럼으로 변환한다. Pass 1에서만 사용되며, Pass 2에서는 기존 칼럼을 재사용하므로 이 모듈을 호출하지 않는다.
- 사용 기술: OpenAI GPT-4 API, Pydantic
- 입력: ticker(종목 심볼), news\_title(뉴스 제목), key\_sentences(TextRank 요약 결과)
- 처리 내용:
  - System Prompt 구성: 초보 투자자용 경제 칼럼 작가 역할을 부여하고, JSON 스키마(제목, 부제, 4개 섹션)와 분량(700~1000자), 톤(해요체)을 정의한다.
  - User Prompt 구성: 종목 심볼, 뉴스 제목, TextRank로 추출한 핵심 문장을 전달한다.
  - API 호출 및 검증: OpenAI GPT-4 API를 temperature 0.7, response\_format json\_object로 호출하고, 반환된 JSON을 Pydantic의 ColumnContent 스키마로 검증한다.
- 출력: ColumnContent 객체 (title, subtitle, sections 4개)
  - JSON 스키마 구조:
    - title (20~25자)
    - subtitle (25~40자)
    - sections (4개):
      1. 오늘의 투자 한입 뉴스 (200~250자)
      2. 최근 무슨 일이야? (250~300자)
      3. 초보 투자자가 알아두면 좋은 포인트 (3~4개 리스트)
      4. 한줄 요약 (100~150자 + 키워드 3개)

#### ⑥ 칼럼 저장 및 조회 모듈 (NewsColumnRepository)

- 역할: 생성된 칼럼을 PostgreSQL에 저장하고 조회한다. Pass 1과 Pass 2 모두에서 사용되며, Pass 2에서는 source\_ticker 필드로 칼럼의 실제 출처를 추적한다.

- 사용 기술: PostgreSQL, asyncpg
- 입력: Column 객체 (ticker, title, subtitle, sections, image\_url, source\_url, source\_ticker 등)
- 처리 내용:
  - 저장 기능: 칼럼 데이터를 JSON으로 직렬화하여 content 필드에 저장한다. UPSERT 쿼리를 사용하여 ticker 기준으로 중복을 방지하며, 동일한 종목의 칼럼은 업데이트된다. Pass 2에서는 source\_ticker 필드에 실제 뉴스 출처 종목을 기록한다.
  - 조회 기능: 특정 종목의 칼럼을 조회하거나, 전체 칼럼을 updated\_at 내림차순으로 조회한다. Pass 2에서는 조회 기능으로 상관 종목의 칼럼 존재 여부를 확인한다.
- 출력: PostgreSQL news\_columns 테이블 (ticker, content(JSON), image\_url, source\_url, source\_ticker, created\_at, updated\_at)

### 3.6 주요 기능의 구현

#### 1. 모의투자 및 AI 리포트 생성

##### 1) 기능 개요

모의투자 및 AI 리포트 생성 기능은 실시간 거래 데이터를 기반으로 사용자의 매매 흐름을 분석하고, 이를 투자 학습 콘텐츠와 연결하는 핵심 엔진이다. AlpacaOrderSyncService가 Alpaca Paper Trading 이벤트를 OrderRequest·TradeCycle·Portfolio 엔티티로 즉시 정리하고, PortfolioOverviewService와 AiDataService가 계정 요약과 차트 데이터를 묶어 FastAPI 콘텐츠 서버로 전달한다. FastAPI 측의 리포트 생성 파이프라인은 pandas/pandas-ta와 ChatGPT를 이용해 ReportTradeCycleResponseDto 기반의 AI 매매 리포트를 생성하고, 사용자별 학습 추천까지 연결한다.

##### 2) 실행 흐름

##### 1. 보유 종목 조회 (PortfolioOverviewService)

- 앱 실행 시 프론트엔드가 Spring Boot의 PortfolioController로 포트폴리오를 요청한다.
- PortfolioOverviewService는 Alpaca OAuth와 Redis 90초 캐시를 이용해 PortfolioOverviewResponseDto를 반환한다.
- FastAPI의 리포트 엔진은 이를 바탕으로 기본 잔고 및 포지션 정보를 확보한다.

##### 2. 주문 체결 반영 (AlpacaOrderSyncService)

- Alpaca Webhook 이벤트가 Redis Stream을 통해 AlpacaOrderSyncService.consume()으로 전달된다.

- 내부적으로 handleBuy/handleSell 메서드를 호출해 OrderRequestRepository에 주문을 저장하고, PortfolioRepository로 보유 수량과 평균 단가를 갱신한다.
- TradeCycleService가 매수·매도 흐름을 사이클 단위로 묶어 손익과 기간을 관리한다.
- 체결 성공 시 WebSocket 채널로 결과를 broadcast하고, 실패 시 로그 기록 및 재시도를 수행한다.

### 3. TradeCycle 기반 분석 데이터 제공 (AiDataService)

- FastAPI가 AiDataService.getTradeCycleForAi(tradeCycleId, interval)을 호출한다.
- AiDataService는 자산 클래스에 따라 분기 처리한다.
  - Crypto: BinanceMarketService.getKlines() → CandleResponseDto.fromBinance()
  - US Equity: MassiveMarketService.getAggregates() → CandleResponseDto.fromMassive()
- 주문 리스트와 차트 데이터를 결합해 ReportTradeCycleResponseDto를 생성한 뒤 FastAPI로 전달한다.

### 4. AI 리포트 생성 (FastAPI ReportGenerator)

- pandas/pandas-ta로 수익률, RSI, MACD 등 주요 지표를 계산한다.
- ChatGPT(OpenAI) API를 호출해 JSON 스키마와 톤이 정의된 시스템 프롬프트를 적용한 자연어 리포트를 생성한다.
- Pydantic 스키마 검증을 통해 필드 누락이나 형식 오류를 방지하고, PostgreSQL(ai\_reports 등)에 저장한다.
- 생성된 리포트는 요약·세부 분석·추천 학습 카드 구조로 구성되어 Spring Boot로 전달된다.

### 5. 표시 및 추천 (Flutter + FastAPI)

- 프론트엔드가 GET /reports 요청을 보내면 FastAPI가 리포트와 함께 추천 이론 카드 및 관련 뉴스 칼럼을 응답한다.
- 앱 내에서 “리포트”와 “추천 학습 카드” 섹션으로 시각화되어 표시된다.

## 2. AI 종목 칼럼 생성 및 추천

### 1) 기능 개요

AI 종목 칼럼 생성 기능은 핵심 종목 170개를 대상으로 최신 해외 시장 뉴스를 수집하고, 웹 크롤링, TextRank 요약, ChatGPT API를 활용하여 초보 투자자용 한국어 칼럼을 자동 생성하는 기능이다. 매일 새벽 5시에 배치 작업으로 실행되며, 사용자가 앱에서 '오늘의 칼럼' 배너를 클릭하면 보유 포트폴리오 기반으로 관련 칼럼을 추천한다.

## 2) 실행 흐름

### 1. 보유 종목 조회 단계

- 사용자가 앱을 실행하면, 프론트엔드는 사용자의 포트폴리오(보유 종목 리스트)를 Spring Boot 서버로 요청한다.
- 서버는 사용자 계정에 저장된 보유 종목 정보를 조회하여 FastAPI 콘텐츠 서버로 전달한다.

### 2. 뉴스 수집 및 분석 단계

- FastAPI 서버의 *데이터 수집 모듈*이 pandas 상관관계수 행렬을 참조해 연관 종목을 탐색하고, Polygon.io API를 통해 해당 종목의 최신 해외 뉴스 URL을 수집한다.
- 뉴스 크롤링 모듈이 BeautifulSoup으로 본문을 파싱하고, 핵심문장 추출 모듈이 TextRank 알고리즘으로 주요 문장을 선별한다.

### 3. AI 칼럼 생성 단계

- 칼럼 생성 모듈이 ChatGPT API를 호출하여 핵심문장을 기반으로 자연어 칼럼을 생성한다.
- 결과는 제목, 요약, 본문 세 부분으로 구조화되어 JSON 형태로 반환된다.

### 4. 저장 및 표시 단계

- 생성된 칼럼은 PostgreSQL DB의 news\_column 테이블에 저장된다.
- Spring Boot 백엔드가 FastAPI에서 결과를 수신하여 프론트엔드로 전달한다.
- 프론트엔드는 최신 칼럼을 '오늘의 뉴스' 섹션에 표시하고, 사용자가 직접 열람할 수 있도록 한다.

## 3) 모듈 간 데이터 흐름

### 1. 데이터 준비 (주 1회)

Polygon API (8000개 티커) → LiquidStocksService (시가총액 정렬) → liquid\_stocks 테이블 (상위 3000개) → CorrelationService (90일 주가 수집 + pandas 상관관계수 계산) → correlations 테이블 (각 종목당 상위 20개, 총 60,000개)

### 2. 칼럼 생성 (매일 새벽 5시)

[Pass 1]

핵심 종목 169개 → PolygonService (뉴스 검색 + 관련성 필터링) → ArticleScraper (크롤링) → TextProcessor (TextRank 35%) → ChatGPTClient (AI 생성) → news\_columns 테이블 (120개) + failed\_tickers (50개)

[Pass 2]

failed\_tickers → CorrelationService (상관 종목 20개 조회) → NewsColumnRepository (칼럼 조회 및 복사) → news\_columns 테이블 (40개 추가)

### 3. 칼럼 추천 (사용자 요청 시)

[전략 1: 포트폴리오 직접 매칭]

- 입력: 사용자 포트폴리오 3개 종목
- 처리: 각 종목의 칼럼 존재 여부 확인
- 출력: 매칭된 칼럼 중 랜덤 1개 (source="portfolio")

[전략 2: 상관종목 매칭]

- 조건: 전략1에서 매칭 실패
- 처리: 포트폴리오 각 종목의 상관계수 상위 10개 조회 → 상관종목 칼럼 순차 검색
- 출력: 첫 번째 발견된 칼럼 (source="correlation", correlation 값 포함)

[전략 3: 인기 칼럼]

- 조건: 전략 2에서도 매칭 실패
- 처리: 최근 업데이트 칼럼 10개 조회
- 출력: 랜덤 1개 (source="popular")

## 4) 핵심 구현

### ① 2-Pass 전략 알고리즘

- Pass 1: 관련성 필터링
  - Polygon API에서 반환된 뉴스 리스트를 순회하며 tickers 배열의 첫 번째 요소가 검색 종목과 일치하는지 확인한다. 이는 해당 종목이 기사의 주요 주제임을 의미한다. 일치하는 기사가 있으면 칼럼 생성 파이프라인을 실행하고, 없으면 failed\_tickers에 추가한다.

```
for article in news_list:
    if article.get("tickers", [])[0].upper() == ticker.upper():
        news_article = article
        break
```

```
if not news_article:
    failed_tickers.append(ticker)
```

- Pass 2: 칼럼 재사용 로직
  - failed\_tickers를 순회하며 correlations 테이블에서 상관관계수가 높은 상위 20개 종목을 조회한다.  
상관 종목을 순회하며 news\_columns 테이블에서 칼럼 존재 여부를 확인하고, 첫 번째 발견 시 ticker만 변경하여 복사한다. source\_ticker 필드에 실제 칼럼 출처를 기록한다.

```
for ticker in failed_tickers:
    related_tickers = await correlation_service.get_related_tickers(ticker, limit=20)

    for related in related_tickers:
        column = await repository.get_column(related["ticker"])
        if column:
            column_copy = Column(ticker=ticker, source_ticker=related["ticker"], ...)
            await repository.save_column(column_copy)
            break
```

## ② 상관관계수 계산 알고리즘

pandas로 3000개 종목의 90일 주가 데이터를 DataFrame으로 변환한다(행: 날짜, 열: 티커, 값: 종가). pct\_change() 함수로 일일 수익률을 계산하고, corr(min\_periods=30) 함수로 상관관계수를 계산한다. min\_periods=30 설정은 최소 30일의 공통 거래 데이터가 있는 경우에만 계산하여 상장일 차이나 거래 정지로 인한 날짜 불일치를 대응한다. 각 종목별로 자기 자신을 제외하고 상관관계수 내림차순 정렬 후 상위 20개만 필터링한다.

```
df_returns = df_close.pct_change(fill_method=None)
correlation_matrix = df_returns.corr(min_periods=30)
for ticker in tickers:
    ticker_correlations = correlation_matrix[ticker].drop(ticker)
    top_20 = ticker_correlations.nlargest(20)
```

## ③ 3단계 추천 전략

전략 1: 포트폴리오 3개 종목 각각에 대해 칼럼을 조회하고, 칼럼이 있는 종목들을 available\_columns에 추가한다. 리스트가 비어있지 않으면 random.choice()로 랜덤 1개를 선택한다.



전략 2: 각 포트폴리오 종목의 상관 종목 10개를 조회하고, 상관 종목을 순회하며 칼럼이 있는지 확인한다. 첫 번째 발견 시 즉시 반환하고 더 이상 검색하지 않는다.

전략 3: 최근 업데이트된 칼럼 10개를 updated\_at 내림차순으로 조회하고, random.choice()로 랜덤 1개를 선택한다.

```
# 전략 1
for ticker in portfolio_tickers:
    column = await get_column(ticker)
    if column:
        available_columns.append(column)

if available_columns:
    return random.choice(available_columns)

# 전략 2
for ticker in portfolio_tickers:
    related = await get_related_tickers(ticker, limit=10)
    for r in related:
        column = await get_column(r["ticker"])
        if column:
            return column

# 전략 3
popular = await get_all_columns(limit=10)
return random.choice(popular)
```

#### ④ ChatGPT 프롬프팅

1. 시스템 프롬프트: ‘초보 투자자용 경제 칼럼 작가’ 역할을 부여하고, JSON 스키마(title 20~25자, subtitle 25~40자, sections 4개)와 분량(700~1000자), 톤(해요체), 작성 가이드를 상세히 정의한다.

```
SYSTEM_PROMPT = """
당신은 초보 투자자에게 복잡한 금융 뉴스를 쉽게 설명하는 경제 칼럼 작가입니다.

입력받은 영문 뉴스 요약을 기반으로, 초보 투자자가 이해할 수 있도록 한국어 투자 칼럼을 작성하세요.
출력은 반드시 순수 JSON 형태로 반환하고, 불필요한 설명이나 마크다운을 추가하지 마세요.

JSON 구조:
{
  "title": "string (20~25자)",
  "subtitle": "string (25~40자)",
  "sections": [
    {"header": "💡 오늘의 투자 한입 뉴스", "body": "string (200~250자)"},
    {"header": "📈 최근 무슨 일이야?", "body": "string (250~300자)"},
    {"header": "🧠 초보 투자자가 알아두면 좋은 포인트", "list": ["string (40~60자)"]},
```

```
...]],
  {"header": "📝 한줄 요약", "body": "한줄 요약 + 키워드"}
]
}
```

작성 가이드:

#### 📏 분량

- 전체: 700~1,000자 (최대 1,200자)
- 제목: 20~25자 (핵심 키워드 2개 + 의문형/비유형)
- 부제: 25~40자 (시장 톤 요약)
- 오늘의 투자 한입 뉴스: 200~250자 (기사 요약 + 자산 설명, 2~3문장)
- 최근 무슨 일이야?: 250~300자 (구체적 이슈 + 시장 영향, 2~3문단)
- 초보 투자자가 알아두면 좋은 포인트: 각 40~60자 × 3~4개
- 한줄 요약: 100~150자 (한줄 요약 문장 + 키워드 3개)

#### 📌 스타일

- 톤: 따뜻하고 친절한 설명체 (초등 고학년 독해 수준) ~해요, ~예요 같은 어미 사용
- 문장: 짧고 명확하게
- 전문용어: 쉽게 풀어쓰되 괄호로 영문 병기 (예: 스테이블코인(Stablecoin))
- 표현: "~일 수 있습니다", "~할 가능성" 같은 신중한 표현
- 금지: 투자 조언 금지 (정보 전달만)

#### 🎯 섹션별 가이드

1. "💡 오늘의 투자 한입 뉴스":
  - 뉴스의 핵심 내용을 2~3문장으로 요약
  - 관련 자산(코인/주식)이 무엇인지 간단히 설명
2. "📈 최근 무슨 일이야?":
  - 구체적 날짜, 숫자 포함
  - 시장에 미친 영향이나 투자자 반응 설명
3. "🧠 초보 투자자가 알아두면 좋은 포인트":
  - 3~4개 항목 (각 40~60자)
  - 용어 설명, 투자 시 주의점, 핵심 인사이트
  - 번호 형식: "1. ...", "2. ...", "3. ..."
4. "📝 한줄 요약":
  - 형식: "한줄 요약 문장\n📖 함께 보면 좋은 키워드: 키워드1, 키워드2, 키워드3"
  - 한줄 요약: 40~60자 (균형잡힌 시각, 핵심 메시지)
  - 키워드: 3개 (초보 투자자가 학습하면 좋을 투자/주식 개념)  
예시: 펀더멘털 분석, 유동성, 리스크 관리, 포트폴리오 다각화, 배당수익률, 기업가치 평가, 시장 변동성, 손절매, 장기 투자, 자산 배분, 가치 투자, 성장주 등
  - 뉴스 내용의 단순 키워드가 아닌, 이 뉴스를 이해하는데 필요한 투자 개념을 추천

## 2. 유저 프롬프트 구조

```
user_prompt = f"""
종목: {ticker}
뉴스 제목: {news_title}

뉴스 요약 (영문):
{key_sentences}

위 내용을 바탕으로 초보 투자자용 칼럼을 JSON 형식으로 작성해주세요.
"""
```

## 3. API 호출 및 검증

```
response = await self.client.chat.completions.create(
    model="gpt-4o-mini",
    messages=[
        {"role": "system", "content": SYSTEM_PROMPT},
        {"role": "user", "content": user_prompt}
    ],
    temperature=0.7,
    response_format={"type": "json_object"} # JSON 강제 반환
)

# JSON 파싱
content = response.choices[0].message.content
column_dict = json.loads(content)

# Pydantic 검증
column_content = ColumnContent(**column_dict)
```

## 4. 프롬프팅 전략

- System Prompt에서 역할, JSON 스키마, 분량, 톤, 섹션별 가이드를 상세히 정의하여 일관된 품질의 칼럼을 생성한다.
- response\_format을 json\_object로 강제하여 파싱 오류를 방지한다.
- Pydantic 스키마 검증으로 필수 필드 누락이나 형식 오류를 사전에 차단한다.
- temperature 0.7로 설정하여 창의성과 일관성의 균형을 유지한다.