

# Pop Tree

Treeples A

김세은, 박승규, 심은수

## 1. 프로젝트 개요

- 프로젝트명 및 팀명
- 개발 기간 및 인원 구성
- 개발 배경 및 문제 인식
- 주요 목표 및 서비스 개요
- 기대효과

## 2. 시장 및 사용자 분석

- 시장 현황
- 사용자 페르소나 및 문제 인식
- 주요 사용자 니즈
- 본 서비스의 시장 내 위치

## 3. 요구사항 정의

- 기능적 요구사항 (FR)
- 비기능적 요구사항 (NFR)
- 서비스 흐름 요약

## 4. 시스템 설계

- 시스템 구성 개요
- 전체 아키텍처
- 주요 앱 및 책임 분리
- ERD (데이터베이스 설계 개요)
- 시스템 아키텍처 다이어그램
- Flow Chart
- 데이터 흐름 예시
- API 상세 명세

## 5. 주요 기능 설명

- 소비 내역 저장 및 요약 분석
- 소비 기반 트리 성장 시각화
- 사용자 점수 및 랭킹 시스템
- 금융상품 정보 제공
- 사용자 인증 및 계정 관리

## 6. 개발 환경 및 기술 스택

- 프로젝트 폴더 구조
- 프론트엔드 환경 (Vue.js 기반 SPA)
- 백엔드 환경 (Django REST API 구조)
- 데이터베이스
- 외부 데이터 연동
- 개발 지원 도구

## 7. 테스트 및 검증

- 기능 구현 중 발생한 오류 및 검증 절차
- Git 병합 및 배포 중 발생한 충돌 처리
- 검증 체크리스트

## 8. 생성형 AI 활용 내역

- 더미 데이터 자동 생성 및 분류 지원
- 추천 알고리즘 구조 설계 및 보조 구현
- 코드 기반 성능 개선 및 구조 최적화
- AI 기반 금융상품 추천 기능 구현
- 생성형 AI 활용 결과 요약

## 9. 성과 및 기대효과

- 주요 성과
- 기대효과

## **10.향후 발전 방향**

- 소비 기반 행동 인식 시스템 고도화
- 금융 데이터 기반 복합 추천 로직 구축

## **11.개발후기**

## 1. 프로젝트 개요

### 1.1 Pop Tree & Treeple A

프로젝트명 Pop Tree는 'Pop'이라는 단어가 지닌 경쾌하고 터지는 듯한 이미지와 'Tree'를 활용한 시각적 성장 메타포를 결합한 명칭이다. 이는 사용자의 소비 습관 변화를 유도하고, 이를 나무의 성장으로 직관적이고 흥미롭게 시각화하는 금융 플랫폼이라는 프로젝트의 핵심방향성을 반영한다. Pop Tree는 단순한 지출 관리에서 나아가, 사용자에게 재미와 동기부여를 동시에 제공하는 새로운 형태의 소비 경험을 목표로 한다.

팀명 Treeple A는 팀원 3인을 의미하는 'Triple'과 최고 신용등급을 나타내는 'AAA'에서 착안하여 구성된 명칭이다. 이는 세명의 팀원이 하나로 뭉쳐 최고의 결과를 만들어내겠다는 의지를 담고 있으며, 프로젝트 수행에 있어 신뢰성과 완성도를 최우선 가치로 삼겠다는 의미를 내포하고 있다.

### 1.2 인원 구성 및 개발 기간

본 프로젝트는 총 3인으로 구성된 개발팀에 의해 수행되었다.

프론트엔드와 백엔드로 역할을 분담하여 프로젝트에 기여하였다.

- 프론트엔드: Vue.js를 활용한 사용자 인터페이스(UI) 개발
- 백엔드: Django 및 Django REST Framework(DRF)를 기반으로 한 API 서버 구축 및 데이터 처리

프로젝트는 2025년 4월 28일부터 5월 28일까지 약 5주간 진행되었으며, 단기간 내에 기획, 설계, 구현, 테스트를 모두 완료하는 것을 목표로 체계적인 협업을 통해 개발이 이루어졌다.



#### 1주차

서비스 기획 및 요구사항 분석을 통해 프로젝트 방향성과 핵심 기능을 정의하고, 사용자 페르소나를 바탕으로 문제 해결 아이디어를 수립하였습니다. UI/UX 흐름에 따라 화면 mockup을 작성하고

전체 구조를 설계하였다.

## 2주차

백엔드 구조 설계 및 모델링을 바탕으로 핵심 API 구현을 시작하고, 사용자 인증, 소비 내역 저장, 점수 계산 로직 개발하였습니다. 프론트엔드 기본 틀을 구성하고 화면 전환 및 API 연동을 준비하였다.

## 3주차

소비 점수에 따른 트리 성장 로직과 사용자 랭킹 시스템을 완성하고, 금융상품수집 기능을 연동하였다. GPT API를 활용한 금융상품 추천 기능도 구조를 설계하고 테스트 기반으로 구현을 시작하였다.

## 4주차

전체 기능 통합 및 실제 데이터 흐름 테스트를 수행하며, 소비 내역 → 점수 → 트리 → 랭킹으로 이어지는 흐름을 안정화 하였다. 발생한 버그를 수정하고, 응답 속도 및 API 응답 구조를 정비하였다.

## 5주차

프로젝트 최종 결과물을 정리하고 보고서와 발표 자료를 작성하였다. README, ERD, Swagger 문서 및 시연 자료를 마무리 하였다.

## 1.3 개발 배경 및 문제 인식

현대인들은 바쁜 일상에서 가계부를 매일 기록하기 어렵고, 대부분 은행 앱의 잔고만 확인하며 소비를 판단하는 경향이 있다.

그러나 잔고에는 반영되지 않은 '보이지 않는 지출'(예: 카드 청구액, 자동이체, 고지서 등)이 많기 때문에, 실제 재정 상황을 오판하여 불필요한 소비를 하거나 예기치 못한 경제적 곤란을 겪는 사례가 많다.

대표 페르소나 '김민준(34세, 직장인)'은 이런 문제를 극명하게 보여준다.

가계부를 쓰지 않던 그는 카드 값과 관리비가 빠져나가며 마이너스 잔고를 경험했고, 이후 소비 관리의 필요성을 절감했다. 하지만 매일의 기록은 너무 번거롭고 지속하기 어려웠다.

## 1.4 주요 목표 및 서비스 개요

Pop Tree는 다음과 같은 방식으로 문제를 해결하고자 한다.

- 자동 소비 데이터 수집 및 분석: 카드사 또는 은행 데이터를 기반으로 소비 내역을 자동으로 저장하고 분류
- 소비 습관 시각화: 누적 소비 점수를 기반으로 '나무'가 성장하는 형태로 시각 피드백 제공
- 게이미피케이션 요소 도입: 좋은 소비 습관을 유도하는 랭킹 시스템 탑재
- 맞춤형 금융상품 추천: 사용자 조건(나이, 가입 방식, 예금/적금 선호 등)을 반영해 최적 상품 추천

## 1.5 기대효과

본 프로젝트를 통해 기대되는 주요 효과는 다음과 같다.

- 기록에 대한 부담 없이 소비를 인식하고 개선할 수 있는 습관 형성: 자동화된 데이터 기반 분석을 통해 사용자가 별도의 수기 입력 없이도 자신의 소비 패턴을 인식하고 개선 방향을 모색할 수 있다.
- 시각적 재미를 통한 지속적인 동기 부여: 소비 점수에 따라 나무가 성장하는 시각적 요소를 통해 사용자가 꾸준히 서비스를 이용할 수 있도록 유도하고, 즐거운 사용자 경험을 제공한다.
- 합리적 금융 의사 결정 지원: 소비 내역과 지출 패턴의 시각화 및 점수화를 통해 사용자가 보다 명확한 재정 상태를 파악하고, 이를 바탕으로 현명하고 계획적인 소비 결정을 내릴 수 있도록 돕는다.

## 2. 시장 및 사용자 분석

### 2.1 시장 현황

최근 몇 년간 디지털 금융 및 소비 관리 시장은 빠르게 성장하고 있다. 특히 가계부 애플리케이션, 소비 통계 시각화 도구, 금융 상품 비교 플랫폼 등이 다수 출시되었으며, 이는 사용자들이 보다 손쉽게 개인 재정을 관리하고자 하는 수요 증가를 반영한다.

비대면 자산관리 수단에 대한 관심이 커지는 가운데, 사용자들은 단순한 잔고 확인을 넘어 소비 패턴 분석, 예상 지출 관리, 자동 금융상품 추천까지 통합된 서비스를 요구하고 있다. 또한 배지, 성장 피드백 등 게임적 요소를 결합한 게이미피케이션 기반 재무관리 서비스에 대한 관심도 증가하고 있다.

다만, 대부분의 서비스는 시각화나 비교 기능 중 하나에만 초점이 맞춰져 있으며, 일상 속 소비의 재미와 금융 정보의 실용성을 동시에 충족하는 서비스는 부족한 상황이다.

## 2.2 사용자 페르소나 및 문제 인식

### 페르소나 - 김민준 (34세, 직장인)

김민준은 바쁜 일정 속에서 은행 앱을 통해 잔고만 확인하며 소비 여부를 판단하는 전형적인 직장인이다. 그는 가계부를 따로 쓰지 않으며, 실제 지출이 반영되지 않은 잔고를 보고 소비 결정을 내리다가 예상치 못한 마이너스 잔고를 경험한 적이 있다.

지출 내역을 일일이 기록하는 데는 번거로움과 심리적 부담이 커서, 장기적인 소비 관리가 어렵다. 그는 다음과 같은 문제를 가지고 있다.

- **문제 1:** 잔고에 보이지 않는 지출(카드 청구, 자동이체 등)이 많아 실제 사용 가능 금액을 파악하기 어려움
- **문제 2:** 가계부 기록에 시간과 에너지를 지속적으로 투자하기 어렵고, 흥미도 부족함
- **문제 3:** 시각적으로 인식 가능한 소비 통계나 동기 부여 요소가 부족함

## 2.3 주요 사용자 니즈

본 프로젝트는 김민준과 같은 사용자들의 니즈를 다음과 같이 구조화하여 반영하였다.

분류	사용자 니즈
실용성	실제 사용 가능한 잔액을 빠르게 파악하고 싶다
편의성	별도 기록 없이 자동으로 소비 내역이 정리되길 원한다
지속성	재미 요소 또는 동기 부여 요소가 있어야 꾸준히 사용할 수 있다
이해도	단순한 숫자보다 시각적 요소(그래프, 성장도 등)를 통해 소비 패턴을 파악하고 싶다
확장성	내 소비 성향에 맞는 금융상품 추천까지 통합되길 원한다

## 2.4 본 서비스의 시장 내 위치

Pop Tree는 다음과 같은 요소를 통해 경쟁 서비스와 차별화된다:

- 자동 소비 내역 처리 + 시각화 기반 트리 성장이라는 독자적 UX 구조



- 사용자의 행동을 긍정적으로 유도하는 게이미피케이션 구조
- 소비 행동과 금융상품 추천을 유기적으로 연결하는 데이터 기반 추천 구조

이는 단순한 소비 통계 제공을 넘어서, 사용자로 하여금 스스로의 소비 습관을 "직관적으로 인지하고 개선할 수 있게 만드는 점"에서 경쟁력을 가진다.

### 3. 요구사항 정의

#### 3.1 기능적 요구사항 (Functional Requirements)

Pop Tree는 사용자의 소비 데이터를 기반으로 다음과 같은 기능을 제공한다. 모든 기능은 RESTful API를 통해 프론트엔드(Vue.js)와 통신하며, 사용자 맞춤형 시각 피드백 및 금융정보를 연동하는 구조를 따른다.

분류	기능	설명
사용자 인증	회원가입 및 로그인	JWT 기반 인증 시스템을 사용하며, 사용자 이름, 이메일, 비밀번호, 닉네임을 등록
사용자 정보	유저 정보 조회 및 수정	사용자의 기본 정보 확인 및 일부 필드 수정 기능
소비 관리	소비 내역 입력 및 저장	사용자의 일별 소비 내역을 기록하며, 금액, 카테고리, 날짜 정보를 포함
소비 분석	월간 소비 요약	사용자의 월별 소비 총액 및 요약 통계를 계산하여 응답
시각화	나무 성장 시각화	소비 점수를 누적하여 시각화 요소(트리 단계)로 변환하고 자동 반영
점수 시스템	소비 점수 관리	일일 소비 내역 기반으로 점수를 산정하여 저장하고, 누적 점수에 따라 트리 레벨을 갱신
랭킹 시스템	일간 용자 점수 순위	누적 용자 점수를 기준으로 글로벌 랭킹 기록을 생성
금융 추천	예적금 상품 제공	금융감독원 API를 통해 예금 및 적금 상품을 수집하여 사용자에게 제공
금융 분석	금리 계산 및 요약	수집된 상품의 기본금리 및 우대금리를 정제 및 요약 후 응답

#### 3.2 비기능적 요구사항 (Non-Functional Requirements)

항목	요구사항
보안	모든 사용자 정보 및 API 통신은 JWT 인증 기반으로 처리되어야 하며, 비밀번호는 암호화 저장

성능	API 응답 속도는 1초 이내, 금융상품 수집은 서버 시작 시 자동 처리되며 주기적 갱신이 가능해야 함
확장성	향후 ETF 등 추가 금융 상품 연동을 고려한 모델 설계
유지보수	백엔드 모듈은 apps 단위로 분리되어 있으며, views, serializers, services 모듈화 기준을 따름

### 3.3 서비스 흐름 요약

1. 사용자는 회원가입 후 로그인하여 서비스를 이용할 수 있다.
2. 로그인시 소비 내역이 업데이트되고 데이터가 저장되고 소비 점수가 자동 계산된다.
3. 누적 소비 점수에 따라 트리 성장 시각화가 변경된다.
4. 점수는 랭킹 시스템에 반영되며, 전체 사용자 중 자신의 위치를 확인할 수 있다.
5. 사용자는 예금/적금 금융상품 정보를 확인하고, 조건에 맞는 상품을 비교 및 탐색할 수 있다.

## 4. 시스템 설계

### 4.1 시스템 구성 개요

본 시스템은 Django 기반 백엔드와 Vue.js 기반 프론트엔드로 구성된 단일 페이지 애플리케이션 (SPA) 구조를 따르며, AWS 기반으로 배포된다. 백엔드는 RESTful API를 제공하며, 소비 입력 및 분석, 금융상품 추천, 점수 기반 시각화 기능을 통합적으로 처리한다.

### 4.2 전체 아키텍처

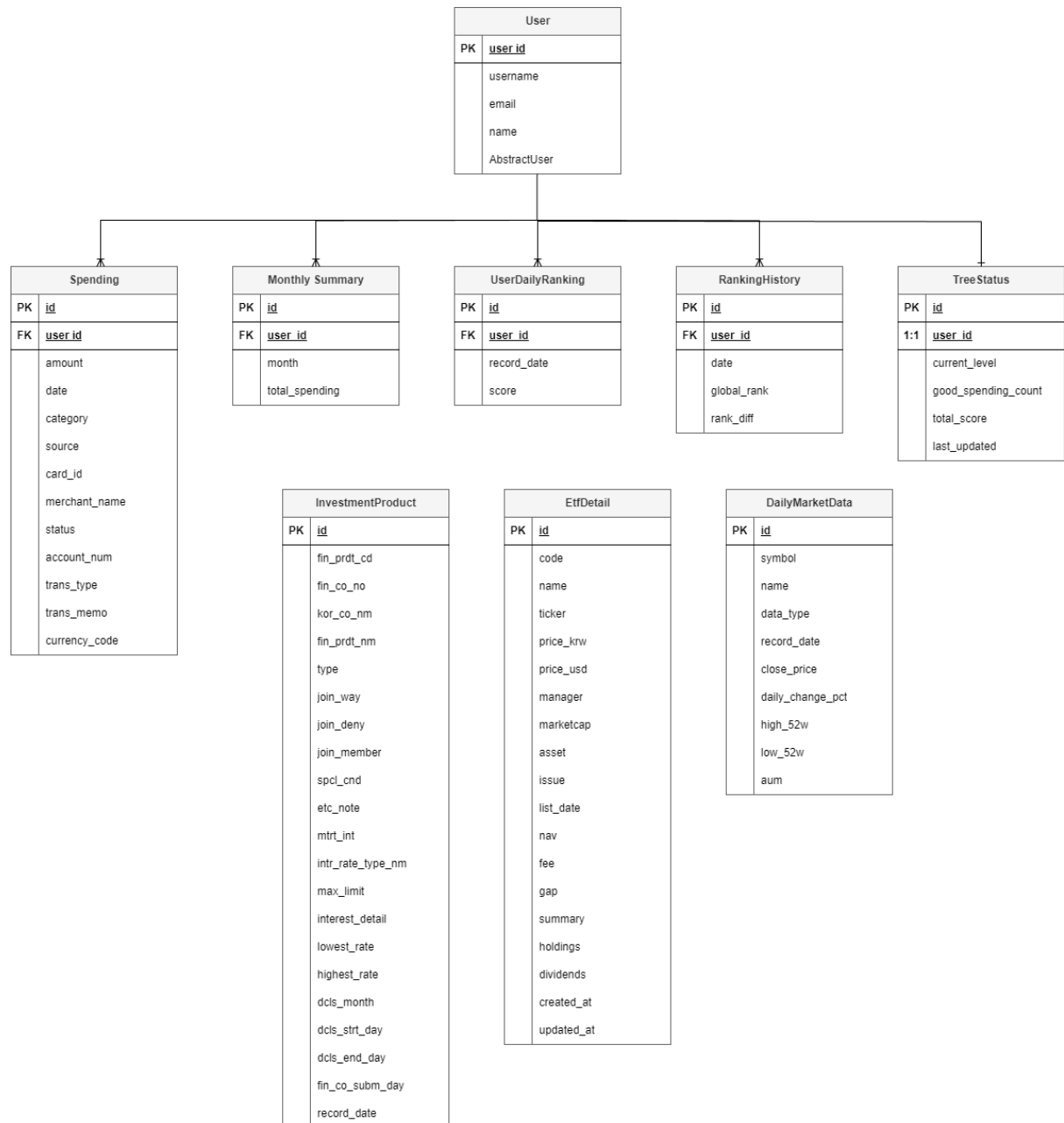
- 프론트엔드: Vue.js
  - 사용자 인터페이스 제공
  - Axios를 이용한 비동기 API 요청 처리
- 백엔드: Django + DRF
  - 사용자 관리, 소비 기록 처리, 금융정보 연동, 트리 성장, 랭킹 관리
- 데이터베이스: SQLite (개발용) / 추후 MySQL 등 관계형 DB 이관 가능
- 외부 API:

- 금융감독원 오픈 API (예적금 상품 수집)
- yfinance (ETF 가격 데이터 수집)

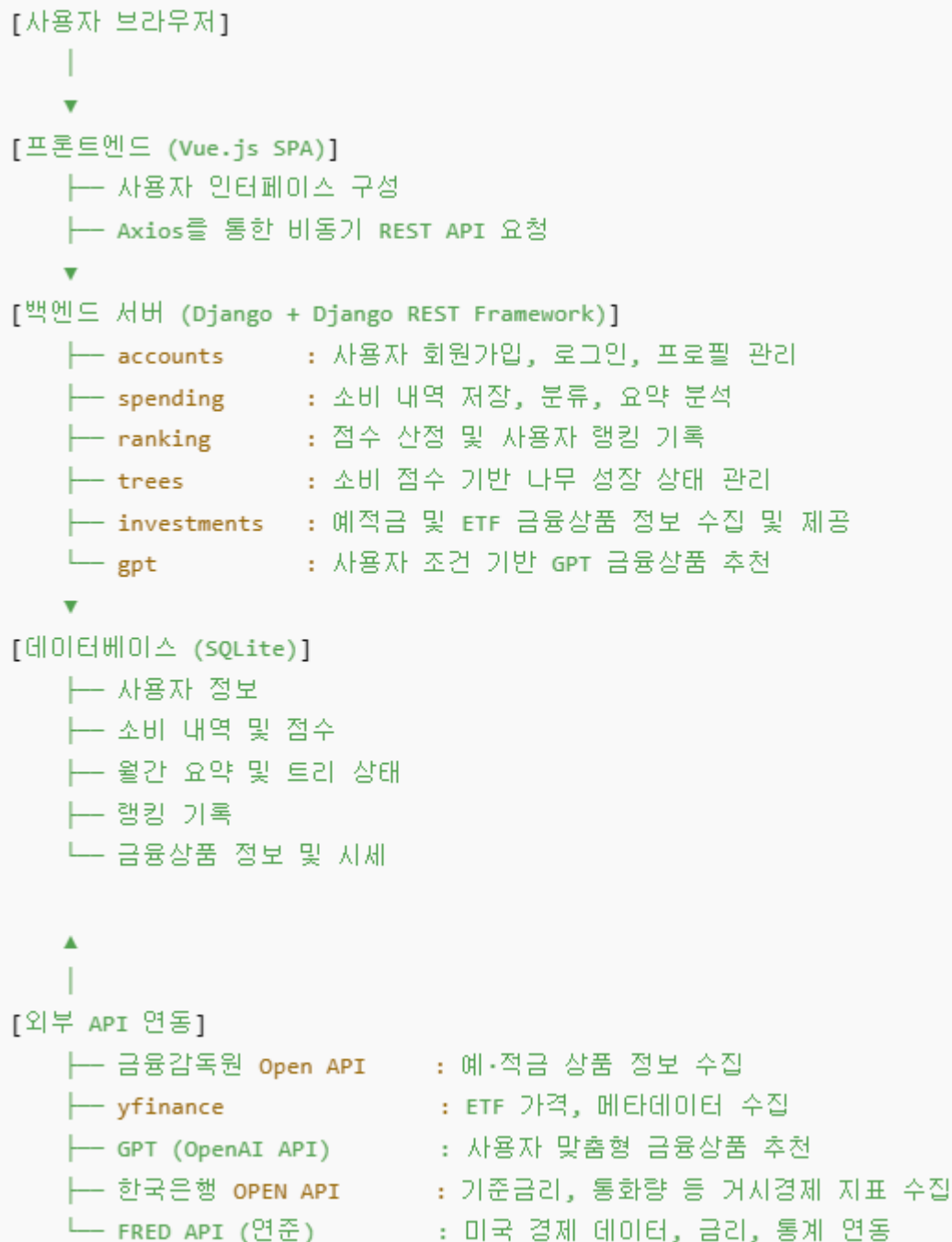
#### 4.3 주요 앱 및 책임 분리

앱 이름	주요 역할
<b>Accounts</b>	사용자 인증 및 프로필 관리 (회원가입, 로그인, 로그아웃 등)
<b>Spending</b>	소비 내역 저장, 자동 분류, 월별 요약 계산
<b>Ranking</b>	사용자 점수 및 일간 랭킹 기록 관리
<b>Trees</b>	소비 점수에 기반한 시각적 트리 성장 처리
<b>Investments</b>	예적금 상품 및 ETF 가격 정보 수집 및 제공
<b>GPT</b>	GPT 기반 금융 추천 연동 영역

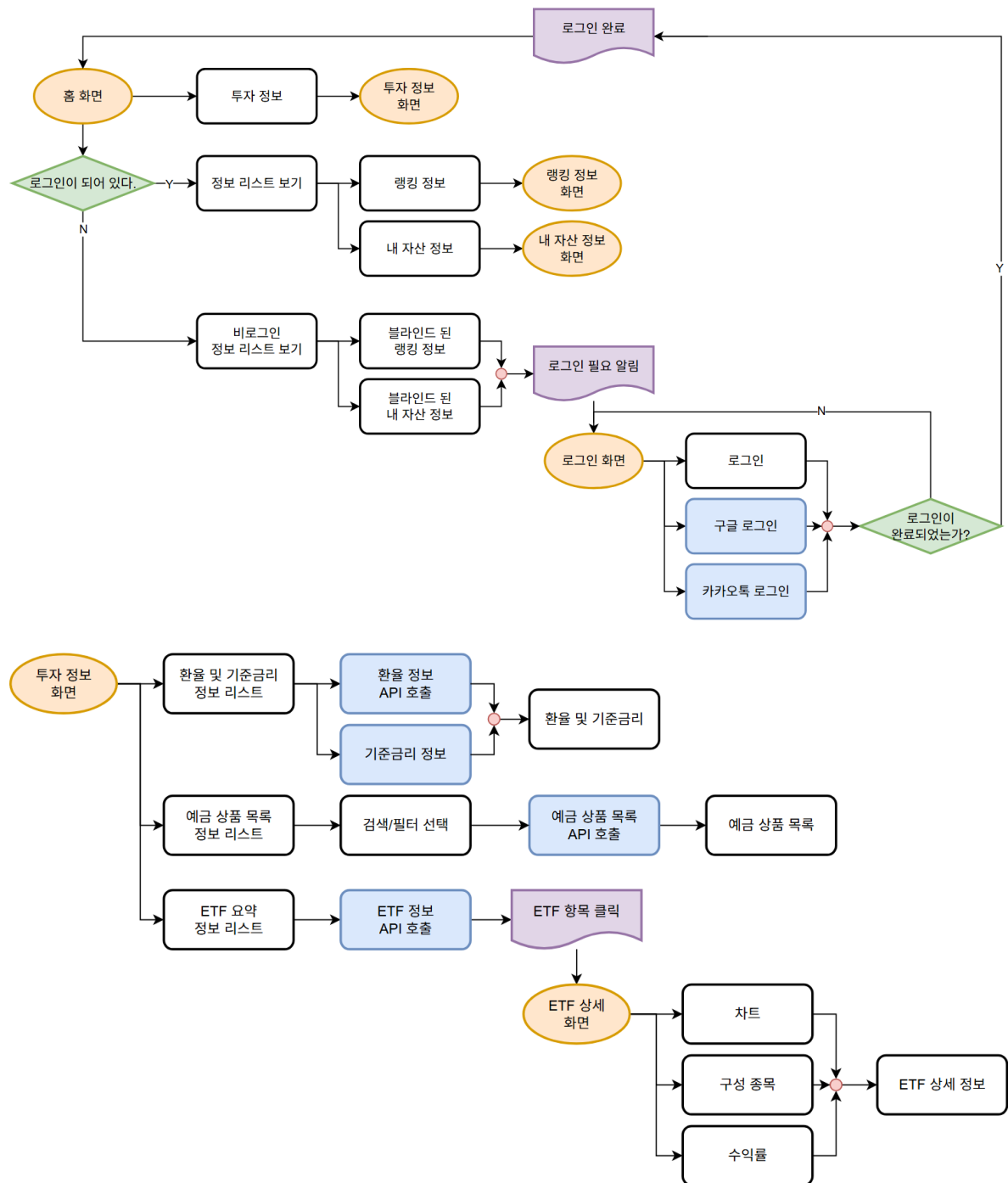
## 4.4 ERD (데이터베이스 설계 개요)

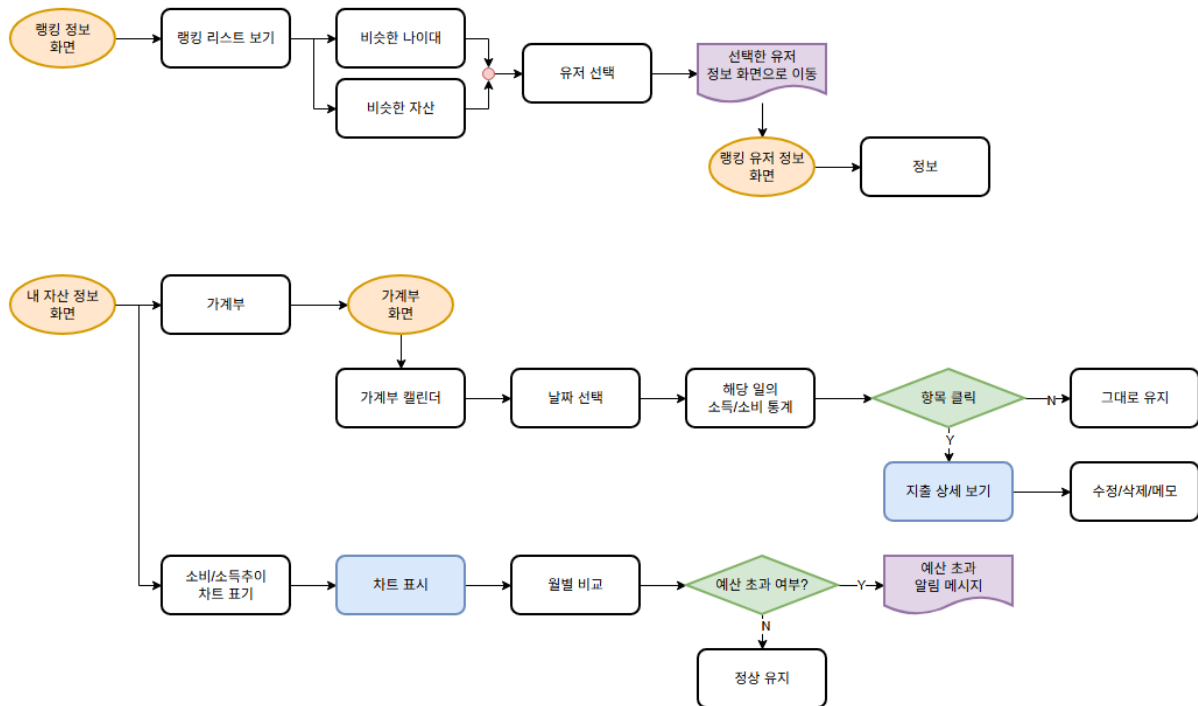


## 4.5 시스템 아키텍처 다이어그램



## 4.6 Flow Chart





## 4.7 데이터 흐름 예시

1. 사용자가 소비 내역을 기록하면:

- Spending 모델에 저장
- summary\_updater.py에서 월별 요약 자동 갱신
- score\_updater.py를 통해 점수 산정
- 트리 성장 로직(trees/utils.py) 트리거 → TreeStatus 갱신
- 점수는 UserDailyRanking에 저장됨

2. ETF 및 예적금 상품은 서버 시작 시 자동 수집되어 DB에 저장되며,

- API 호출 시 프론트엔드에 카드 형태로 전달됨

## 4.8 API 상세 명세

공통 사항:

- 인증 방식: JWT 토큰 (헤더: Authorization: Bearer <token>)
- 요청/응답 포맷: JSON
- 버전링은 미사용 상태 (향후 /api/v1/ 구조 고려 가능)

## 1. 사용자 인증 및 계정 관리 (accounts)

accounts				^
POST	/accounts/auth/login/		accounts_auth_login_create	✓ 🔒
GET	/accounts/auth/logout/	Calls Django logout method and delete the Token object assigned to the current User object.	accounts_auth_logout_list	✓ 🔒
POST	/accounts/auth/logout/	Calls Django logout method and delete the Token object assigned to the current User object.	accounts_auth_logout_create	✓ 🔒
POST	/accounts/auth/password/change/	Calls Django Auth SetPasswordForm save method.	accounts_auth_password_change_create	✓ 🔒
POST	/accounts/auth/password/reset/	Calls Django Auth PasswordResetForm save method.	accounts_auth_password_reset_create	✓ 🔒
POST	/accounts/auth/password/reset/confirm/	Password reset e-mail link is confirmed, therefore this resets the user's password.	accounts_auth_password_reset_confirm_create	✓ 🔒
POST	/accounts/auth/signup/	Registers a new user.	accounts_auth_signup_create	✓ 🔒
POST	/accounts/auth/signup/resend-email/	Resends another email to an unverified email.	accounts_auth_signup_resend-email_create	✓ 🔒
POST	/accounts/auth/signup/verify-email/	Verifies the email associated with the provided key.	accounts_auth_signup_verify-email_create	✓ 🔒
GET	/accounts/auth/user/	Reads and updates UserModel fields Accepts GET, PUT, PATCH methods.	accounts_auth_user_read	✓ 🔒
PUT	/accounts/auth/user/	Reads and updates UserModel fields Accepts GET, PUT, PATCH methods.	accounts_auth_user_update	✓ 🔒
PATCH	/accounts/auth/user/	Reads and updates UserModel fields Accepts GET, PUT, PATCH methods.	accounts_auth_user_partial_update	✓ 🔒
POST	/accounts/delete/		accounts_delete_create	✓ 🔒
PUT	/accounts/update/{username}/		accounts_update_update	✓ 🔒
GET	/accounts/user info/{username}/		accounts_user info_read	✓ 🔒

## 2. 소비 관리 (spending)

spending				^
GET	/spending/by-category/		spending_by-category_list	✓ 🔒
GET	/spending/calendar/		spending_calendar_list	✓ 🔒
GET	/spending/daily/		spending_daily_list	✓ 🔒
GET	/spending/statistics/		spending_statistics_list	✓ 🔒

## 3. 트리 성장 시각화 (trees)

trees				^
GET	/trees/status/		trees_status_list	✓ 🔒

## 4. 점수 및 랭킹 시스템 (ranking)



ranking			^
GET	/ranking/history/	ranking_history_list	▼ 🔒
GET	/ranking/leaderboard/	ranking_leaderboard_list	▼ 🔒
GET	/ranking/score/	ranking_score_list	▼ 🔒
GET	/ranking/today/	ranking_today_list	▼ 🔒

## 5. 금융상품 정보 제공 (investments)

investments			^
GET	/investments/exchange-rate/usd/	investments_exchange-rate_usd_list	▼ 🔒
POST	/investments/gpt/answer/	investments_gpt_answer_create	▼ 🔒
GET	/investments/kr-base-rate/	investments_kr-base-rate_list	▼ 🔒
GET	/investments/market/list/	investments_market_list_list	▼ 🔒
GET	/investments/products/list/	investments_products_list_list	▼ 🔒
GET	/investments/us-base-rate/	investments_us-base-rate_list	▼ 🔒

## 5. 주요 기능 설명

Pop Tree

홈

투자정보

랭킹

로그아웃

회원탈퇴



# 나의 소비 습관을 키워보세요!

소비를 관리하고, 나무처럼 자산을 키우는 Pop Tree

### 🏆 랭킹 정보

전체 랭킹에서  
나의 위치를 확  
인하고 도전하세  
요.

### 🔥 소비 지출 정보

나의 소비 지출  
정보를 확인하고  
관리하세요.

### 📊 투자정보

금융상품 정보를  
비교하고 추천  
금융 상품을 확  
인하세요.

© 2025 Pop Tree. All rights reserved.

본 프로젝트는 사용자의 소비 활동을 중심으로, 데이터를 수집하고 분석하여 시각적으로 피드백을 제공하며, 금융상품 추천 기능까지 통합한 서비스이다. 다음은 구현된 핵심 기능들의 개요이다.

## 5.1 소비 내역 입력 및 요약 분석

### Pop Tree

홈 투자정보 랭킹 [로그아웃](#) [회원탈퇴](#)

카테고리별 소비금액(막대그래프)



5월 2025						
일	월	화	수	목	금	토
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

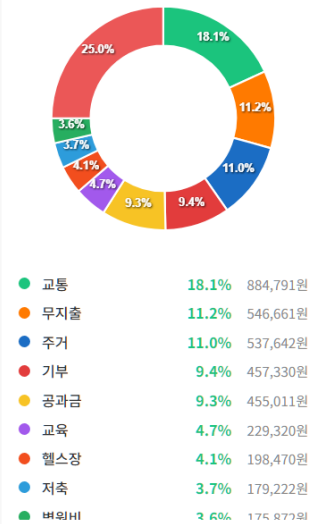
2025-05-01  
76,556 원  
공과금: 60,603 인터넷: 15,953

2025-05-02  
40,667 원  
반려동물: 28,589 기타: 12,078

2025-05-03  
127,160 원  
공과금: 68,099 편의점: 59,061

2025-05-04  
152,951 원

카테고리별 비중



- 기능 개요: 사용자가 직접 입력한 소비 데이터를 저장하고, 자동 분류 및 월별 요약 분석을 수행한다.
- 입력 방식: 소비 일자, 금액, 카테고리, 결제수단 등
- 백엔드 처리:
  - Spending 모델에 저장
  - 저장 시 summary\_updater 모듈을 통해 MonthlySummary 자동 갱신
- 시각화:
  - 월별 소비 총합, 카테고리별 비중 등은 프론트에서 통계 그래프 형태로 제공 예정

## 5.2 소비 기반 트리 성장 시각화

- 기능 개요: 누적 소비 점수에 따라 사용자의 트리가 성장하며, 해당 상태를 시각적으로 제공한다.

- 점수 산정 방식:
  - 소비 내역 기록 시, 금액과 카테고리에 따라 점수 부여
  - score\_updater.py를 통해 일일 점수 저장
- 트리 성장 조건:
  - 누적 점수 기준에 따라 TreeStatus.current\_level이 상승
  - 각 레벨에 대응하는 정적 이미지가 trees/static/에 존재하며 프론트에서 단계별 렌더링
- 사용자 경험:
  - 자신의 소비 습관 개선 정도를 게임 요소처럼 시각적으로 인식 가능

### 5.3 사용자 점수 및 랭킹 시스템

#### 내 랭킹



testuser2

1위 점수: 49 (-0)

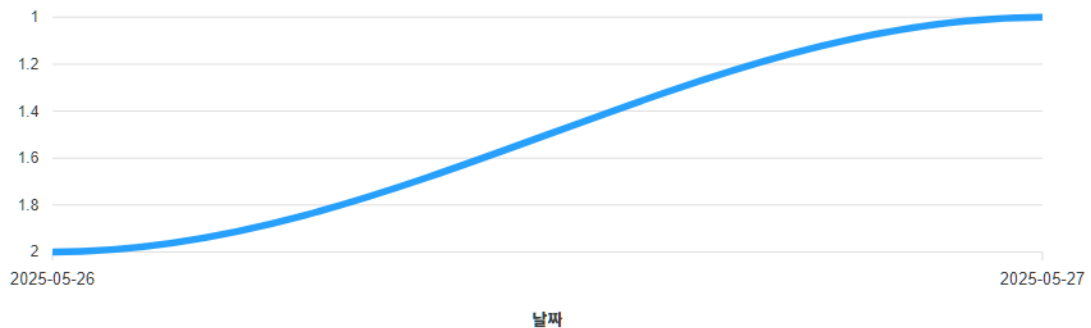
#### 전체 랭킹 TOP 10

순위	닉네임	점수	변화
1	testuser2	49	-0
2	test1	6	-0

#### 내 랭킹 변화 (최근 7일)

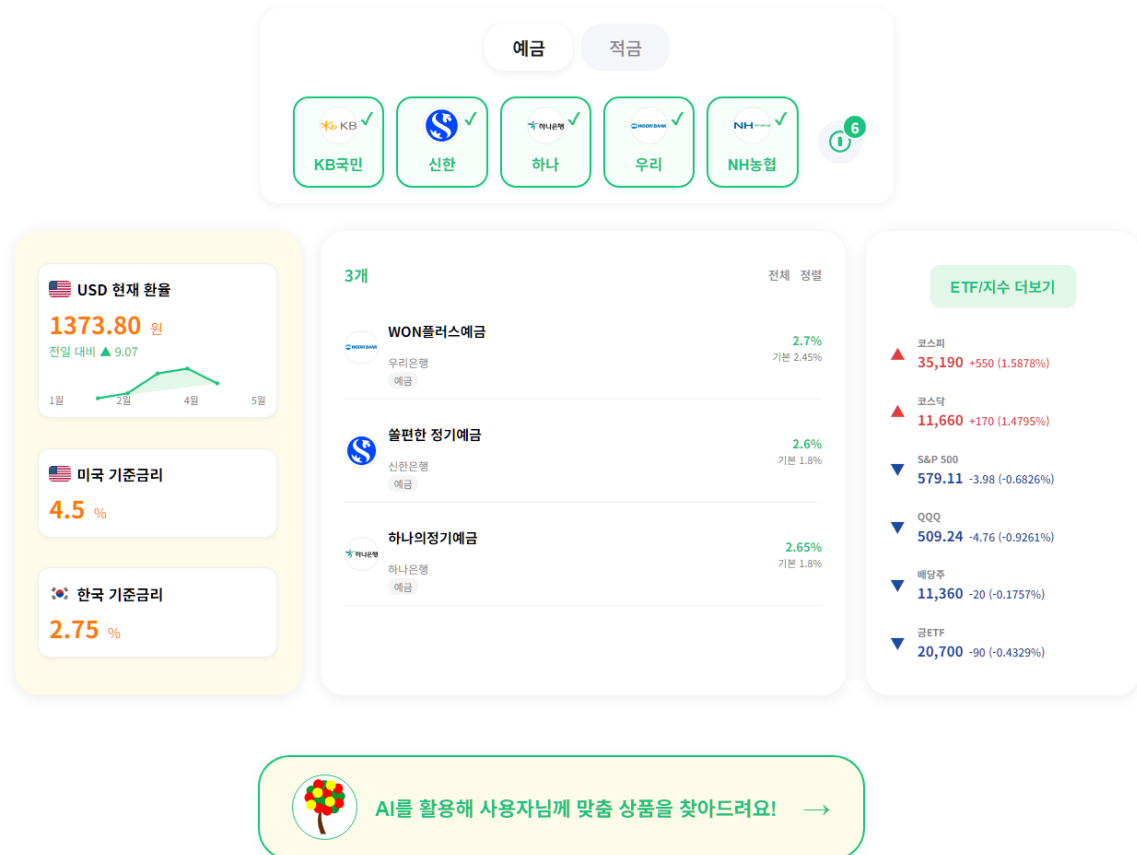
그래프는 최근 7일간의 내 순위 변화를 보여줍니다.

1위에 가까울수록 그래프가 위로 올라갑니다.



- 기능 개요: 매일 사용자별 점수를 집계하고, 전체 사용자와 비교해 랭킹을 제공한다.
- 백엔드 처리:
  - UserDailyRanking 모델에 점수 기록
  - update\_ranking.py 커맨드 또는 소비 변경 시 자동 반영
- 활용 목적:
  - 타 사용자와의 비교를 통해 소비 개선 동기 유도
  - 향후 개인 기록 추적 및 순위 기반 리워드 도입 가능

## 5.4 금융상품 정보 제공 및 AI 기반 추천



- 기능 개요: 사용자에게 금융감독원 API와 yfinance 데이터를 기반으로 예적금 및 ETF 금융상품 정보를 제공할 뿐만 아니라, 소비 및 투자 성향을 바탕으로 GPT를 활용한 맞춤형 금융상품 추천 기능을 함께 제공한다.
- 예적금 정보 제공:
  - 금융감독원 Open API를 활용하여 예금 및 적금 상품 정보를 수집하고, InvestmentProduct 모델에 저장한다.
  - 각 상품의 기본금리(lowest\_rate) 및 우대금리(highest\_rate)를 파싱 및 계산하여 저장하고 응답 시 제공한다.
- ETF 정보 제공
  - Yfinance 라이브러리를 통해 ETF의 실시간 시세, 국가 정보, 메타데이터 등을 수집하고, ETFDetail 모델에 저장한다.
  - 주요 ETF는 정렬 및 필터 기능을 적용한 형태로 프론트엔드에 전달된다.
- GPT 기반 맞춤형 금융상품 추천

Pop Tree

홈
투자정보
랭킹
로그아웃
회원탈퇴

예금
적금

KB국민

신한

하나

우리

NH농협

6

**AI 맞춤 추천 결과**
최준혁님을 위한 추천

**### 사용자 소비 성향 분석**
1. **\*\*소비 패턴\*\***: 주어진 정보를 바탕으로 사용자는 전반적으로 **\*\*안정 추구형\*\***으로 분석됩니다. 금융 상품으로 단기 자금시장에 투자하는 ETF를 선호하거나, 보수적으로 CD금리에 투자하는 것도 소비 성향의 특징을 반영합니다.

**### 잠재적인 투자 성향 유추**
2. **\*\*투자 성향\*\***: 사용자는 위험을 최소화하며 안정적인 수익을 추구하는 것으로 보이며, 장기 투자는 선호하지 않을 가능성이 높습니다. 이런 성향은 단기 금융 상품이나 안전한 채권과 관련된 ETF로 나타납니다.

**### 금융 상품 추천**
3. **\*\*추천 금융 상품\*\***:
1. **\*\*KODEX 머니마켓액티브 (ETF 코드: 488770)\*\***

닫기

**AI를 활용해 사용자님께 맞춤 상품을 찾아드려요!**
→

사용자의 소비 내역(최근 3 개월 소비 금액 및 주요 카테고리 비중)과 프로필 정보를 데이터베이스에서 조회한 후, 해당 내용을 기반으로 프롬프트를 구성한다. 이 프롬프트는 사용자의 투자 관심도, 성향, 소비 패턴 등을 문장 형태로 요약하며, GPT-4o-mini 모델에 입력된다.

추천 로직은 다음과 같은 절차로 구성된다:

1. 사용자 질문 정제: 입력된 질문을 GPT 를 통해 명확하고 구체적으로 변환
2. 텍스트 임베딩 생성: 사용자 질문 + 소비/프로필 데이터를 하나의 텍스트로 구성하여 임베딩 생성
3. 금융상품 데이터 임베딩: 예적금 및 ETF 데이터를 각각 임베딩하여 FAISS 벡터 검색 인덱스에 등록
4. FAISS 유사도 검색: 사용자 임베딩과 가장 유사한 금융상품 텍스트 3 개를 추출
5. GPT 응답 생성: 유사도가 높은 금융상품 텍스트와 함께 사용자 질문을 다시 GPT 에 입력하여, 우선순위 기반 추천 응답을 생성

추천 결과는 사용자의 소비 및 투자 성향을 해석한 설명과 함께 최대 3 개의 금융상품을 우선순위에 따라 제안하며, 각 상품이 사용자에게 적합한 이유를 간단히 기술한다.

이 기능은 단순한 키워드 필터링 방식이 아닌, 사용자 데이터를 기반으로 금융상품 임베딩 및 GPT 해석을 결합한 통합 추천 시스템으로 작동하며, 사용자의 신뢰도 높은 의사결정을 지원하는 데 목적이 있다.

## 5.5 사용자 인증 및 계정 관리

- 기능 개요: 회원가입, 로그인, 사용자 정보 수정 및 탈퇴까지의 전 과정 제공
- 기술 스택: Django Custom User 모델 + JWT 인증
- API 구조:
  - dj-rest-auth 기반 기본 인증 구조에 커스텀 필드(name)를 추가하여 확장
- 기능적 특성:
  - 회원가입 시 date\_joined를 기준으로 이후 소비 데이터만 분석 대상에 포함
  - 사용자별 점수 및 트리 상태는 계정에 종속되어 1:1로 연결



## 6. 개발 환경 및 기술 스택

본 프로젝트는 Vue.js 기반의 싱글 페이지 프론트엔드와 Django REST Framework 기반의 백엔드로 구성되었으며, 다양한 외부 API 및 개발 도구를 연동하여 기능적 완성도와 확장 가능성을 확보하였다.

### 6.1 프로젝트 폴더 구조

```
poptree_project/
├── poptree/
│   ├── backend/
│   │   ├── accounts/
│   │   ├── spending/
│   │   ├── ranking/
│   │   ├── trees/
│   │   ├── investments/
│   │   ├── gpt/
│   │   ├── config/
│   │   └── manage.py
│   ├── frontend/
│   │   ├── public/
│   │   └── src/
│   │       ├── assets/
│   │       ├── components/
│   │       ├── views/
│   │       ├── router/
│   │       └── App.vue, main.js 등
│   └── README.md
├── etf_price.py
├── inst_rate.py
└── .editorconfig
```

### 6.2 프론트엔드

- 프레임워크: Vue.js
- 구성 구조: Single Page Applications (SPA)
- 통신 방식: Axios를 활용한 REST API 비동기 요청
- 구현 방식: 컴포넌트 기반 인터페이스 및 사용자 흐름 구현

### 6.3 백엔드

- 언어 및 프레임워크: Python 3.9, Django, Django REST Framework
- 구조 구성: 앱 단위 분리 (accounts, spending, ranking, trees, investments, gpt)

- 인증 방식: JWT 기반 인증(dj-rest-auth 활용)
- 핵심 처리 로직: 사용자 인증, 소비 내역 저장 및 점수 산정, 트리 성장, 랭킹 기록, 금융 상품 정보 제공

## 6.4 데이터베이스

- DB 엔진: SQLite (개발 환경 기준)
- 주요 테이블: 사용자, 소비 내역, 점수 및 트리 상태, 금융상품 정보 등
- ERD 설계: 1:1 및 1:N 관계 기반 모델링, JSONField 활용 포함

## 6.5 외부 데이터 연동

다양한 외부 데이터 제공처와 연동하여 실시간 금융정보를 수집하고 추천 기능에 활용하였다.

- 금융감독원 Open API: 예·적금 상품 정보 수집 및 금리 데이터 정제
- yifinance: ETF 가격 및 메타 정보 수집
- FRED API (연준): 미국 경제 지표, 금리 정보 수집
- 한국은행 Open API: 국내 기준금리, 통화량 등 거시경제 정보 수집
- GPT PI (OpenAI): 사용자 조건 기반 금융상품 추천 프롬프트 실행

## 6.6 개발 지원 도구

- GitLab: 형상 관리 및 협업
- Notion: 요구사항 정리 및 일정 관리
- Draw.io: ERD 및 아키텍처 다이어그램 시각화
- OpenAI: 코드 설계, 로직 설계, 더미데이터 생성, 추천 로직 작성 지원

## 7. 테스트 및 검증

본 프로젝트에서는 개발 초기 단계에서 유닛 테스트 코드를 체계적으로 구성하지는 않았으나, 개발 및 통합 과정에서 발생한 다양한 오류를 수동 디버깅과 로그 기반 분석을 통해 해결하였다. 아래는 주요 검증 사례와 그 과정을 정리한 것이다.

### 7.1 기능 구현 중 발생한 오류 및 검증 절차

#### 1. Django admin 설정 오류 (admin.E108)

- 문제: InvestmentProduct 모델에서 rate 필드를 삭제했음에도 admin.py에서 list\_display에 여전히 포함되어 있어 오류 발생
- 검증 및 해결:
  - admin.py에서 list\_display 필드 수정
  - Django 시스템 체크 통과 및 관리자 페이지 정상 렌더링 확인

## 2. apps.py의 순환 참조 문제

- 문제: apps.py에서 services.py의 함수를 임포트하는 과정에서 ImportError 발생
- 검증 및 해결:
  - services.py 내 실행 코드 제거
  - apps.py 내 try-except 구문으로 래핑하여 초기화 시점에서 로그로 오류 확인
  - 서버 실행 후 정상 동작 로그 확인

## 3. DRF 템플릿 오류 (TemplateDoesNotExist)

- 문제: 브라우저로 API 접근 시 rest\_framework/api.html 템플릿을 찾지 못해 오류 발생
- 검증 및 해결:
  - settings.py에 DEFAULT\_RENDERER\_CLASSES를 JSONRenderer로 명시
  - 이후 브라우저 요청 시 JSON 응답 정상 출력 확인

## 4. 모델 필드 리네이밍 과정의 마이그레이션 오류

- 문제: rate → lowest\_rate, highest\_rate로 필드 변경 시 Django가 필드 리네임으로 인식
- 검증 및 해결:
  - makemigrations 진행 중 "N"을 선택하여 새 필드로 처리
  - migrate 후 정상 반영 확인

## 5. JWT 인증 오류 및 설정 누락

- 문제: CSRF 오류, 잘못된 인증 클래스 참조, username 필수 요구 등 인증 흐름에서 여러 문제 발생
- 검증 및 해결:
  - JWT 설정 (REST\_USE\_JWT, DEFAULT\_AUTHENTICATION\_CLASSES) 반영

- username/email 기반 로그인 흐름 설정
- Postman을 통한 access/refresh 토큰 발급 및 인증 확인

## 7.2 Git 병합 및 배포 중 발생한 충돌 처리

- 문제: .pyc, \_\_pycache\_\_ 디렉토리 충돌로 병합 실패
- 검증 및 해결:
  - 문제 파일 제거 후 .gitignore에 예외 규칙 추가
  - Git conflict 해소 후 재커밋 및 정상 푸시 확인

## 7.3 검증 체크리스트

항목	확인 여부
관리자 페이지 오류 제거	○
소비 기록 → 점수 반영 흐름 정상 작동	○
트리 레벨 자동 갱신 확인	○
랭킹 기록 및 점수 산정 갱신 정상 작동	○
예적금/ETF API 응답 및 DB 저장 정상 확인	○
JWT 로그인 및 인증 토큰 발급 확인	○

## 8. 생성형 AI 활용 내역

Pop Tree 프로젝트에서는 생성형 인공지능 도구(ChatGPT)를 단순 참고 수준을 넘어, 데이터 구성 자동화, 추천 알고리즘 보조 구현, 서비스 품질 개선, AI 기반 서비스 기능 자체의 구현에 활용하였다. 다음은 구체적인 활용 사례이다.

### 8.1 더미 데이터 자동 생성 및 분류 지원

- 활용 목적: 테스트용 소비 기록 더미 데이터를 반복 생성하고, 카테고리별 소비 항목 분류기를 사전 구성하기 위함
- 활용 방식
  - 다양한 실제 소비 항목 데이터를 ChatGPT를 통해 생성
  - 도서, 식비, 건강 등 카테고리별 키워드 목록 자동 수집

- 소비 상점명을 기준으로 카테고리를 자동 분류하는 로직의 조건 테이블 생성에 도움
- 성과
  - 소비 카테고리 분류 정확도 향상
  - 테스트 자동화 및 초기 분류기 성능 확보

## 8.2 추천 알고리즘 구조 설계 및 보조 구현

- 활용 목적: 예·적금 및 ETF 데이터를 바탕으로 사용자 조건 기반 추천 알고리즘 설계
- 활용 방식
  - 사용자 나이, 가입 방식, 가입 금액 등의 조건에 따라 추천 우선순위를 설계
  - 조건 분기, 정렬 로직 등의 구조적 설계에 대해 AI 자문 활용
  - 추천 후보군 필터링 기준과 모델 필드 정의 최적화
- 성과:
  - 추천 로직의 가독성과 조건 명시성이 향상
  - 추천 결과의 사용성 개선

## 8.3 코드 기반 성능 개선 및 구조 최적화

- 활용 목적: 반복 로직, 응답 속도, 필드 처리 등의 효율성 개선
- 활용 방식
  - 복잡한 뷰 함수와 serializer 내 중복 로직 제거를 위한 구조 개선 자문
  - services.py, utils.py 분리 기준 재설계에 참고
  - DB 조회 최소화 및 캐시 전략 제안 검토
- 성과
  - API 응답 처리 속도 안정화
  - 모듈 간 의존도 감소 및 유지보수성 향상

## 8.4 AI 기반 금융상품 추천 기능 구현

- 활용 목적: 실제 사용자 조건을 기반으로 GPT가 직접 금융상품을 추천하는 기능 구현
- 구현 방식
  - /gpt/ 엔드포인트를 구성하여 사용자 요청값을 기반으로 OpenAI API 호출
  - 사용자의 연령, 선호 상품군, 위험 성향 등을 질의로 구성하여 GPT가 가장 적합한 예적금 또는 ETF 상품을 추천
  - 응답 결과를 카드 형태로 프론트에 제공
- 성과
  - 타 추천 알고리즘과 차별화된 설명 가능한 추천 결과 제공
  - 실제 서비스 기능으로서 GPT 활용 구현 완료

## 8.5 생성형 AI 활용 결과 요약

활용 분야	기여 내용
데이터 자동화	소비 카테고리 분류 기반 더미 데이터 생성
추천 알고리즘	사용자 조건 기반 금융상품 추천 구조 설계 보조
코드 개선	API 응답 최적화, 코드 중복 제거, 모듈 구조 개선
서비스 기능	GPT를 통한 금융상품 실시간 추천 기능 구현

Pop Tree는 생성형 AI를 단순 보조가 아닌 기능적 핵심 컴포넌트로 직접 활용한 사례이며, 실제 사용자 서비스를 통해 실현 가능한 형태로 통합하였다.

## 9. 성과 및 기대효과

본 프로젝트는 소비 기록과 금융정보를 통합적으로 분석하고 시각화하는 것을 목표로 하였으며, 핵심 기능이 실제로 구현되어 서비스 구조를 완성하였다. 이를 통해 다음과 같은 성과와 기대효과를 도출할 수 있다.

### 9.1 주요 성과

#### 1. 소비 기반 시각화 구조 완성

- 사용자의 소비 내역에 따라 트리가 성장하는 구조를 설계하고, 실제 점수 산정 로직과 이미지 기반 단계 시스템을 구현하였다.

- 소비 습관을 데이터가 아닌 시각적인 변화로 인식하게 하여, 사용자 행동 개선을 유도할 수 있는 기반을 마련하였다.

## 2. 금융상품 추천 시스템 구현

- 금융감독원 Open API와 yfinance를 연동하여, 예.적금 및 ETF 정보를 실시간으로 수집하고 구조화 하였다.
- 사용자 조건을 반영한 필터 기반 추천 외에도, GPT를 활용한 설명형 금융 추천 기능을 별도 구현함으로써 개인화된 금융 의사결정을 지원할 수 있게 되었다.

## 3. 점수 및 랭킹 시스템 정합성 확보

- 소비 내역 기반 점수 계산 → 사용자 트리 성장 → 글로벌 랭킹 반영이라는 일련의 흐름이 전체적으로 연동되어 정상 작동하도록 구현하였다.
- 일간 점수 기록, 전일 대비 순위 변화 계산 등 데이터 기반 피드백이 가능한 구조를 완성하였다.

## 4. 인증 및 사용자 관리 안정화

- JWT 기반 회원가입 및 로그인 기능을 구현하고, 사용자 정보를 안전하게 관리할 수 있는 인증 체계를 구축하였다.
- name 필드 확장 등 커스텀 유저 구조를 적용하여, 향후 다양한 사용자 속성 반영이 용이한 구조를 확보하였다.

## 9.2 기대효과

분야	기대효과
사용자 경험	사용자가 직접 기록하지 않아도 소비 습관을 직관적으로 파악할 수 있는 시각적 피드백 제공
지속적 사용 유도	트리 성장, 랭킹 상승 등 게임적 요소를 통해 반복 사용 유도
금융 이해도 향상	맞춤형 금융정보 제공을 통해 재무 인식 개선 및 합리적 선택 지원
기술 확장성	데이터 기반 구조와 API 중심 설계로 향후 기능 확장 및 연결 용이성 확보
AI 실전 적용	데이터 기반 구조와 API 중심 설계로 향후 기능 확장 및 연결 용이성 확보

## 10. 향후 발전 방향

Pop Tree는 사용자의 소비 데이터를 기반으로 점수를 산정하고, 그 점수를 시각화하며, 금융상품 추천까지 연결하는 구조를 완성도 있게 구현한 프로젝트이다. 본 프로젝트는 단순 소비 통계를 넘어서, 행동 데이터 기반 피드백 + 금융 의사결정 지원이라는 두 축을 유기적으로 결합했다는 점에서 고유한 확장 가능성을 지닌다. 다음은 Pop Tree만의 구조적 특성과 기술 흐름에 기반한 실질적인 발전 방향이다.

### 10.1 소비 기반 행동 인식 시스템 고도화

- 사용자의 소비 점수를 정량화하여 시각적 피드백으로 전달하는 구조를 행동 리포트 형태로 확장할 수 있다.
- 예: 소비 유형별 점수 히스토리, 특정 소비 습관 반복 시 리마인더, '가장 이상적인 소비일' 분석 등
- 점수 변화 추이에 따라 성장하는 나무 외에도 다양한 시각적 메타포 적용이 가능하다 (예: 감정 그래프, 소비 일지 등).

### 10.2 금융 데이터 기반 복합 추천 로직 구축

- 금융감독원 API와 yfinance 데이터를 통합 처리하는 구조가 완비되어 있어, 향후 이를 기반으로 한 다중 조건 필터링 추천, 자동 필드 클러스터링, 조건부 비교 리포트 출력 등 고도화된 금융 분석 기능으로의 확장이 용이하다.
- 단순 리스트 제공을 넘어, "내 소비성향과 가장 유사한 사용자들이 선택한 상품"과 같은 행동 기반 추천 로직 개발 가능성을 내포한다.

Pop Tree는 단순한 소비 기록 앱이나 금융상품 비교 서비스가 아니라, 행동을 분석하고, 의미를 부여하며, 금융 결정을 돕는 방향으로 작동하도록 완성된 구조이다. 이 구조를 바탕으로 한 향후 확장은 기능적 추가보다 사용자 경험 중심의 진화로 이어질 수 있다.

## 11. 개발후기

### 김세은

백엔드, 그 안에서도 기능 구현을 위주로 담당했습니다. 외부 API, 크롤링 등을 통해 데이터를 불러오는 부분, 그리고 GPT 추천 기능을 구현하는 과정에서 API 구조에 대한 이해를 비롯한 많은 공부が必要했었습니다. 다른 금융데이터와 달리 ETF 상세 정보 부분에서 원하는



API가 없어 크롤링으로 가져오게 되는 등 시행착오도 많았지만, 최종적으로 GPT 추천 서비스까지 기획했던 기능들을 모두 구현할 수 있어 뿌듯했습니다.

## 박승규

프론트엔드를 담당하며 화면 구성뿐만 아니라 파일 구조와 데이터 흐름을 고려해 백엔드와의 연결을 매끄럽게 구현해야 했습니다. Vue3, Django, 다양한 API와 라이브러리를 활용하면서 코드 작성부터 화면 출력까지 전체 과정을 예상하고 설계하는 것이 가장 어려웠습니다. 생성형 AI와 공식문서 검색, 이전 학습 결과물과 유튜브 강의를 통해 높은 학습곡선을 극복하고 맡은 부분을 기한 내에 완성할 수 있었습니다. 이론뿐만 아니라 실제 동작하는 결과물을 만들어보는 소중한 경험이 되었습니다.

## 심은수

수업 시간에 배운 Django, DRF, Vue 등의 기술을 실제 서비스 구현에 적용하며, 배움을 실전에 연결하는 과정을 흥미롭게 경험할 수 있었습니다. 저는 소비 점수 산정 로직과 트리 성장 알고리즘을 직접 설계하고 구현하며, 백엔드 로직이 사용자 경험과 어떻게 맞물리는지를 구체적으로 체감할 수 있었습니다. 또한 금융감독원 OpenAPI와 yfinance 등 외부 데이터를 수집하여 API 형태로 가공하는 작업을 통해 데이터 통신 구조에 대한 이해도 함께 키울 수 있었습니다. 데이터 흐름을 명확히 기록하고, 예외 상황을 정교하게 처리하는 개발 습관의 중요성을 배웠습니다. 예상치 못한 오류를 해결해 나가면서 문제 해결 능력 또한 실질적으로 향상되었다고 느꼈습니다. 무엇보다도, 각자 맡은 역할에 최선을 다하며 신뢰 속에서 협업했던 팀원들에게 깊은 감사를 전하고 싶습니다. 이번 프로젝트는 단순한 기능 구현을 넘어서, 개발자로서의 성장 방향을 스스로 발견하는 값진 기회였습니다.