



README (KOR)



Finance Tree

프로젝트 개요

Finance Tree는 트리 구조를 기반으로 **브랜치**를 생성하고 관리하며, 각 브랜치에서 발생하는 **트랜잭션**을 기록하고 조회할 수 있는 시스템입니다. 사용자는 트랜잭션을 추가, 수정, 삭제할 수 있으며, 거래와 관련된 영수증을 업로드하여 관리할 수 있습니다.

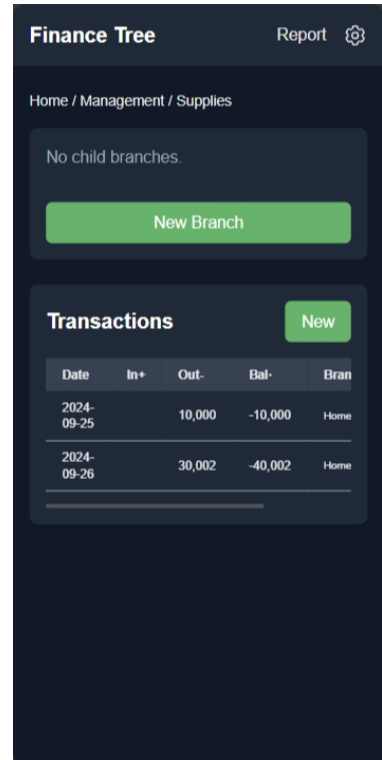
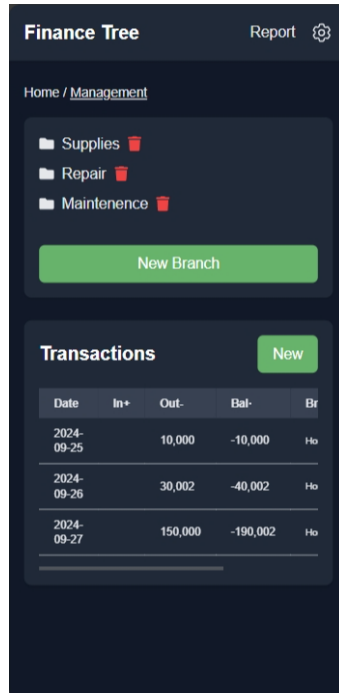
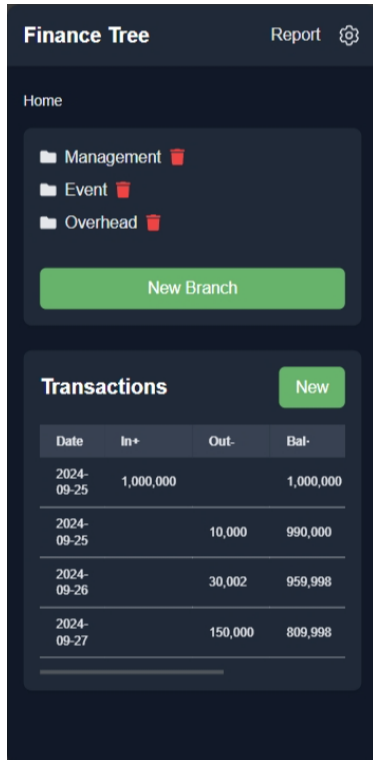
🎯 주요 목표:

- 조직의 복잡한 구조나 개인 재정 관리를 직관적으로 할 수 있도록 도와줍니다.
 - 트리 구조로 **브랜치 관리**와 **트랜잭션 기록**을 손쉽게 처리할 수 있습니다.
-

✨ 주요 기능

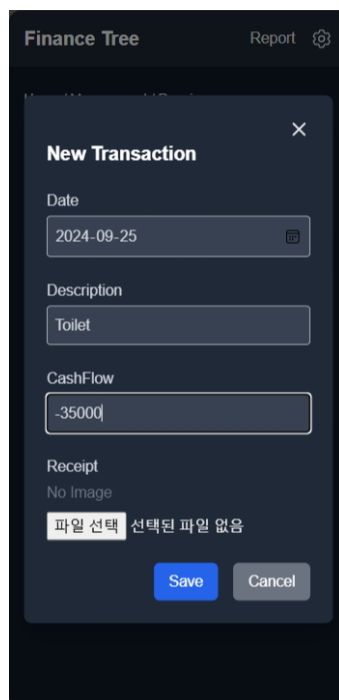
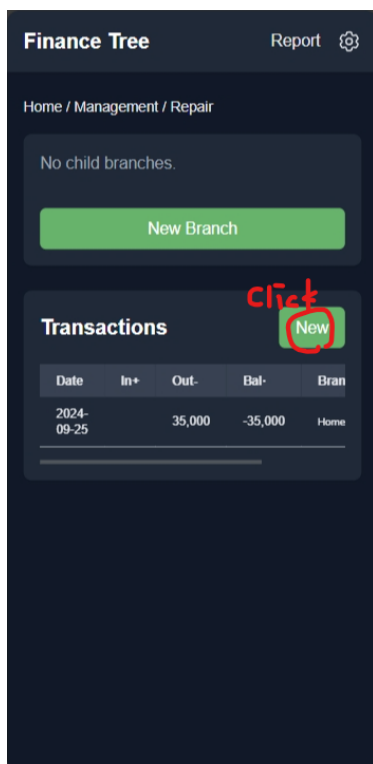
1. 🌿 트리 구조 브랜치 관리

- **브랜치 생성/삭제:** 트리 구조에서 새로운 브랜치를 추가하거나, 불필요한 브랜치를 삭제할 수 있습니다.
- **계층적 구조:** 부모-자식 관계를 명확하게 구분하여 관리할 수 있습니다.



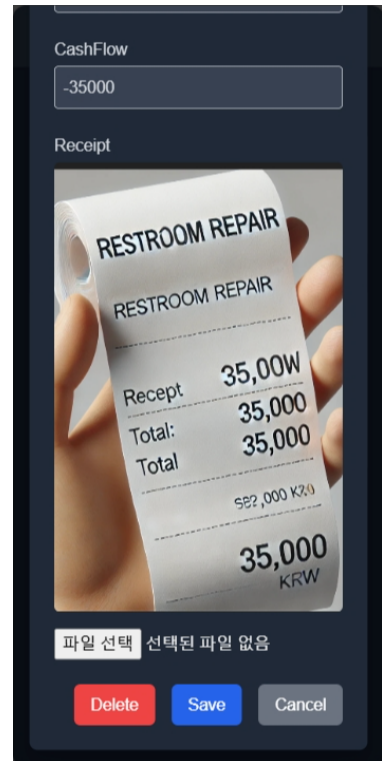
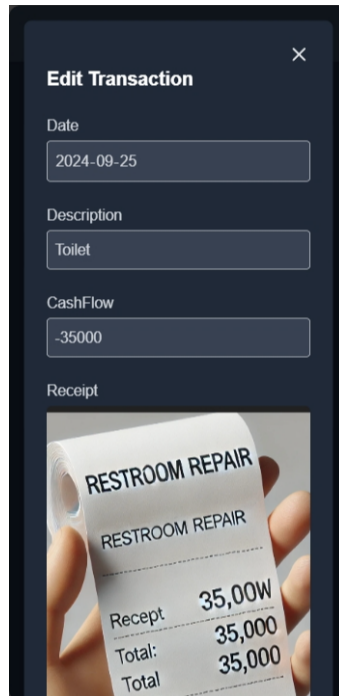
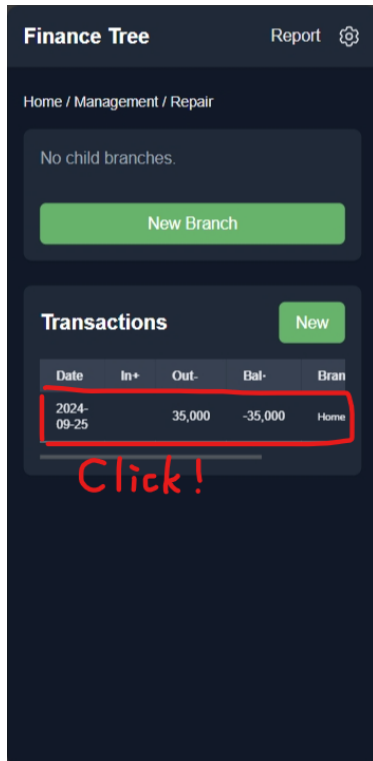
2. 💰 트랜잭션 기록 및 조회

- **트랜잭션 추가:** 각 브랜치에 대한 거래 내역을 추가하고, 설명과 금액을 기록할 수 있습니다.
- **트랜잭션 수정/삭제:** 기존 트랜잭션을 쉽게 수정하거나 삭제할 수 있습니다.
- **잔액 조회:** 각 브랜치의 총 잔액을 확인할 수 있습니다.



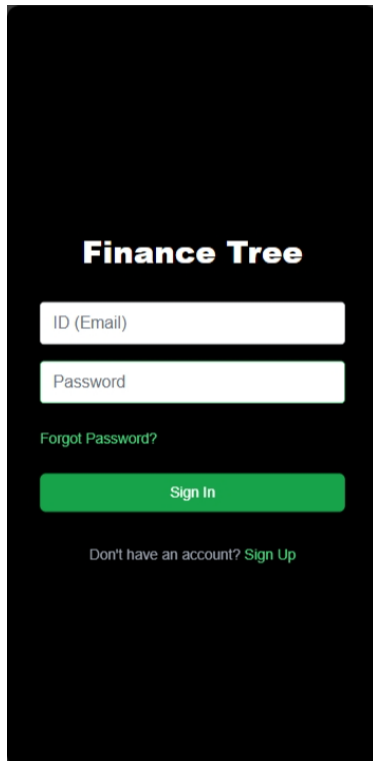
3. 영수증 이미지 관리

- 영수증 업로드: 트랜잭션과 연계된 영수증 이미지를 업로드할 수 있습니다.
- 영수증 조회: 트랜잭션과 관련된 영수증 이미지를 클릭해 확인할 수 있습니다.



4. 🗝️ 사용자 인증 및 관리

- **회원가입/로그인:** 이메일과 비밀번호로 계정을 생성하고 로그인할 수 있습니다.
- **비밀번호 재설정:** 비밀번호를 잊은 사용자는 이메일을 통해 재설정할 수 있습니다.



Finance Tree

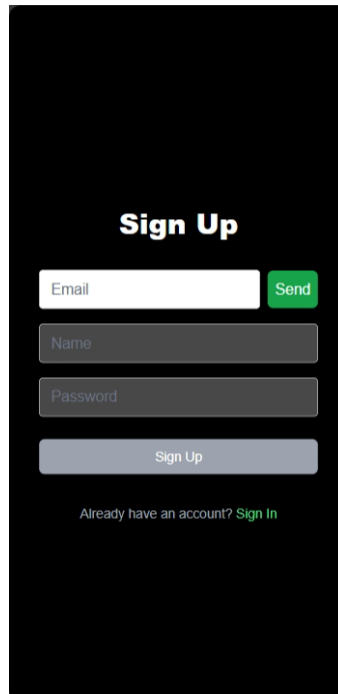
ID (Email)

Password

[Forgot Password?](#)

[Sign In](#)

Don't have an account? [Sign Up](#)



Sign Up

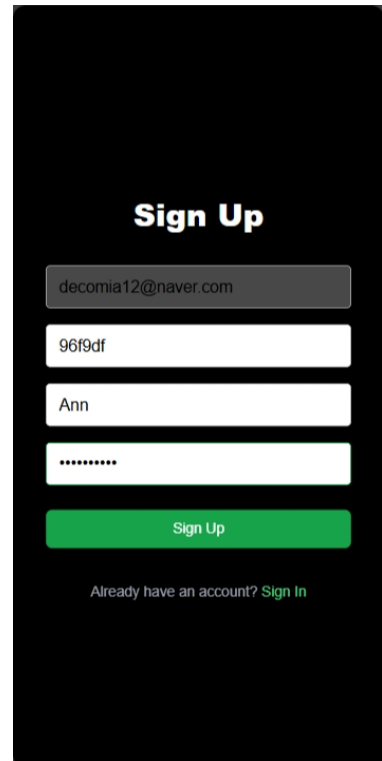
Email [Send](#)

Name

Password

[Sign Up](#)

Already have an account? [Sign In](#)



Sign Up

decomia12@naver.com

96f9df

Ann

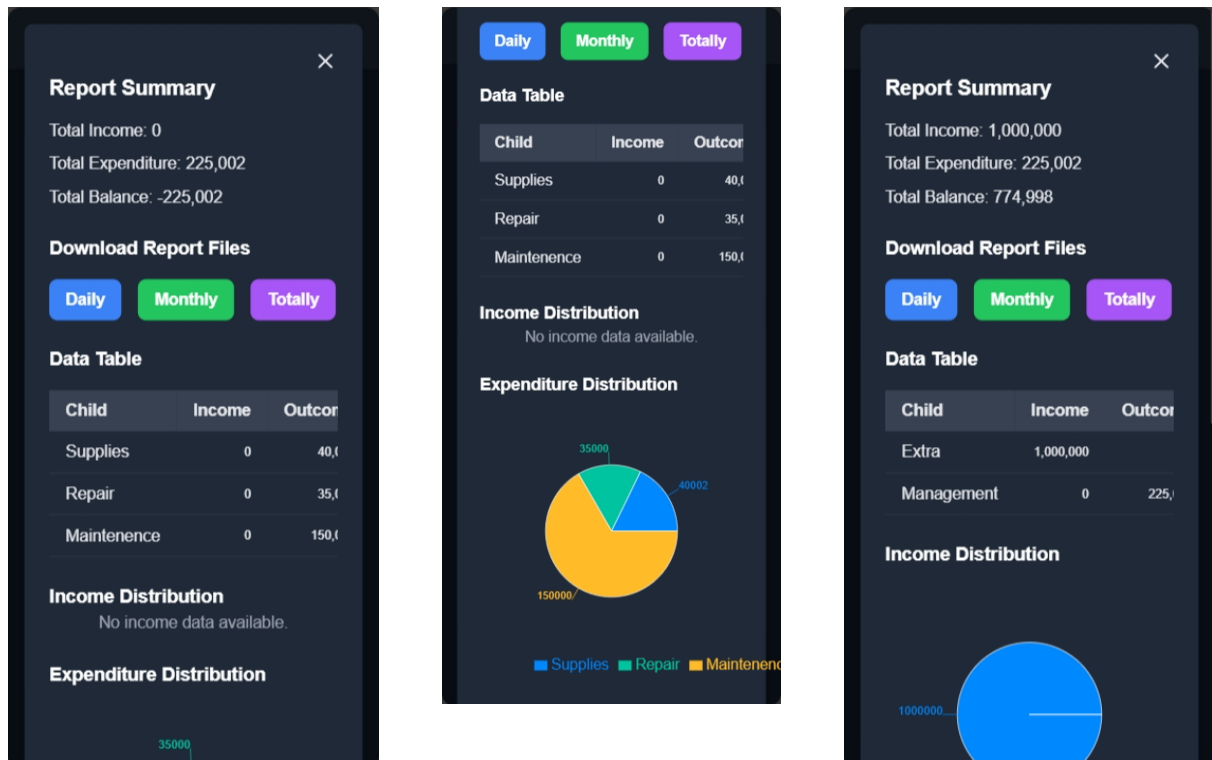
.....

[Sign Up](#)

Already have an account? [Sign In](#)

5. 📊 리포트 생성

- **리포트 생성:** 사용자는 특정 브랜치나 기간별 트랜잭션을 기반으로 재정 리포트를 생성할 수 있습니다.
- **리포트 커스터마이징:** 브랜치나 기간을 선택하여 리포트를 사용자 지정할 수 있습니다.
- **리포트 다운로드 또는 보기:** 리포트를 직접 확인하거나 외부 사용을 위해 다운로드할 수 있습니다 (예: CSV 또는 PDF 형식).



🔧 기술 스택

- **Frontend:** Next.js (React 기반)
- **Backend:** REST API (FastAPI 등으로 구현 가능)
- **Database:** PostgreSQL (또는 다른 관계형 DB)
- **Authentication:** JWT 토큰 기반 인증
- **File Upload:** FormData를 사용한 영수증 이미지 업로드

설치 및 실행 방법

1. 리포지토리 클론 및 설정

```
git clone <https://github.com/username/repository-name.git>
cd repository-name
```

2. 환경 변수 설정

프로젝트 루트 디렉토리에 `.env.local` 파일을 생성하고, 다음 환경 변수를 설정합니다:

```
NEXT_PUBLIC_BASIC_URL=https://your-backend-api-url.com
```

3. 의존성 설치

```
npm install
```

4. 개발 서버 실행

```
npm run dev
```

<http://localhost:3000>에서 실행됩니다.

5. 백엔드 서버 실행

백엔드 서버는 별도로 구현되어 있어야 하며, FastAPI 또는 Express 등을 사용하여 API 서버를 실행해야 합니다.

사용 방법

1. 회원가입

- 이메일과 비밀번호를 통해 새로운 계정을 생성합니다.

2. 로그인

- 로그인 후 브랜치 및 트랜잭션 관리 기능을 사용할 수 있습니다.

3. 브랜치 추가

- 트리 구조에서 브랜치를 추가하거나 삭제할 수 있습니다.

4. 트랜잭션 관리

- 트랜잭션을 추가하고 영수증을 업로드하여 관리할 수 있습니다.

5. 비밀번호 재설정

- 이메일을 통해 비밀번호 재설정 링크를 받을 수 있습니다.

프로젝트 구조

```
/src
├── /app
│   ├── /components      # 공통 컴포넌트
│   ├── /libs            # API 호출 관련 모듈
│   ├── /pages           # 각 페이지 컴포넌트
├── /public              # 정적 파일 (이미지 등)
└── /styles              # 전역 스타일 시트
```

기여 방법

1. 이 프로젝트를 포크합니다.
 2. 새로운 브랜치를 생성합니다 (`git checkout -b feature-branch-name`).
 3. 변경 사항을 커밋합니다 (`git commit -m 'Add some feature'`).
 4. 브랜치에 푸시합니다 (`git push origin feature-branch-name`).
 5. 풀 리퀘스트를 생성합니다.
-