

배



Skatton Project
우리 가게부 앱 개발

KB IT's Your Life 풀스택 개발 과정
성공하면 혁신!

2025. 4. 11. 금

목차

1. 주제 및 구성원
2. 구성도 및 사용 기술
3. 주요 기능
4. 주요 화면 설계
5. 컴포넌트 설계
6. 구현 결과 화면
7. 후기

1. 주제 및 구성원

주제

- Vue.js를 활용하여 온라인 가계부 웹 서비스 제작

목적

- 배운 기술 요소를 실제 서비스에 접목해 봄으로써 각 **개인의 프로그래밍 능력 향상**
- 서비스 개발 과정 중 기획, 설계, 코딩, 테스트, 배포의 각 단계를 **경험**
- 팀 프로젝트 진행 시 협력 및 충돌 경험으로써 **의사소통 능력 향상**



1. 주제 및 구성원

구성원

- 박사은: 통계 페이지 화면 구현 및 시각화
- 박채연: 통계 페이지 데이터 흐름 및 통계 계산
- 송우진: 엑셀 내보내기, 예산 페이지
- 신유리: 카테고리 필터, 메모 검색
- 안다윤: 전체 UI 디자인, 레이아웃, 페이지 네이션
- 오혁신: 거래 내역 목록 조회/추가/삭제
- 정선웅: 시작페이지, 로그인 페이지, 전체 라우팅 설정 및 화면 연결
- 최유정: 거래 내역 캘린더 조회/탭 메뉴

2. 구성도 및 사용 기술

구성도

```
- src
├── layout
│   ├── TheHeader.vue (헤더)
│   ├── TheSidebar.vue (사이드바)
│   └── TheFooter.vue (푸터)
├── pages
│   ├── StartPage.vue (시작)
│   ├── LoginPage.vue (로그인)
│   ├── RegisterPage.vue (회원가입)
│   ├── HomePage.vue (홈)
│   ├── TransactionPage.vue (수입/지출 관리)
│   ├── PopupPage.vue (팝업 화면)
│   ├── PopupListPage.vue (팝업 화면)
│   ├── StatisticsPage.vue (요약 통계)
│   ├── ExportExcelPage.vue (엑셀 내보내기)
│   └── BudgetPage.vue (예산 관리)
├── components
│   ├── SummaryCard.vue (홈화면 지출/수입/예산 컴포넌트)
│   ├── LoginForm.vue (로그인 폼 컴포넌트)
│   ├── SearchByMemo.vue (검색 버튼 컴포넌트)
│   ├── FilterCategory.vue (카테고리 필터 포넌트)
│   ├── TransactionFilter.vue (수입/지출 필터)
│   ├── TransactionContent.vue (지출/수입 목록)
│   ├── CalendarContent.vue (캘린더 폼)
│   ├── AddListBtn.vue (지출/수입 빠른 추가)
│   ├── MonthlyPieChart.vue (파이차트)
│   ├── MonthlyLineChart.vue (라인차트)
│   ├── NetProfit.vue (순이익)
│   ├── ExcelFilter.vue (엑셀 기간 조회 기능)
│   ├── Excellist.vue (조회된 리스트 불러오기)
│   ├── BudgetProgress.vue (예산 그래프)
│   └── BudgetSetting.vue (예산 설정)
├── stores
│   ├── TransactionStore.js (지출/수입 관련 함수)
│   ├── statisticsStore.js (통계 및 기간 조회, 계산용)
│   ├── MonthSelector.js (월별 선택자)
│   ├── useBudgetProgress.js (예산 합계 그래프 적용)
│   └── UseBudgetStore.js (예산 조회용)
```



2. 구성도 및 사용 기술

사용 기술

- 언어

- HTML5, JavaScript

- **프론트엔드**

- vite project, Composition API, vue-router, axios, components, pinia, 이벤트, 스타일 (bs5, fontawesome)

- **백엔드**

- json server (db server)

- **협업 도구**

- Github, Notion, Slack, figma

3. 주요 기능

- 수입/지출 등록, 기존 거래 수정 및 삭제: 날짜, 금액, 카테고리, 메모 정보를 입력하여 기록
- 거래 내역 조회: 필터 기능을 통해 특정 조건에 맞는 거래 내역 조회
- 월별 재정 요약: 각 월의 수입, 지출, 순이익 요약 표기
- 데이터 저장: json-server에 저장됨

4. 주요 화면 설계

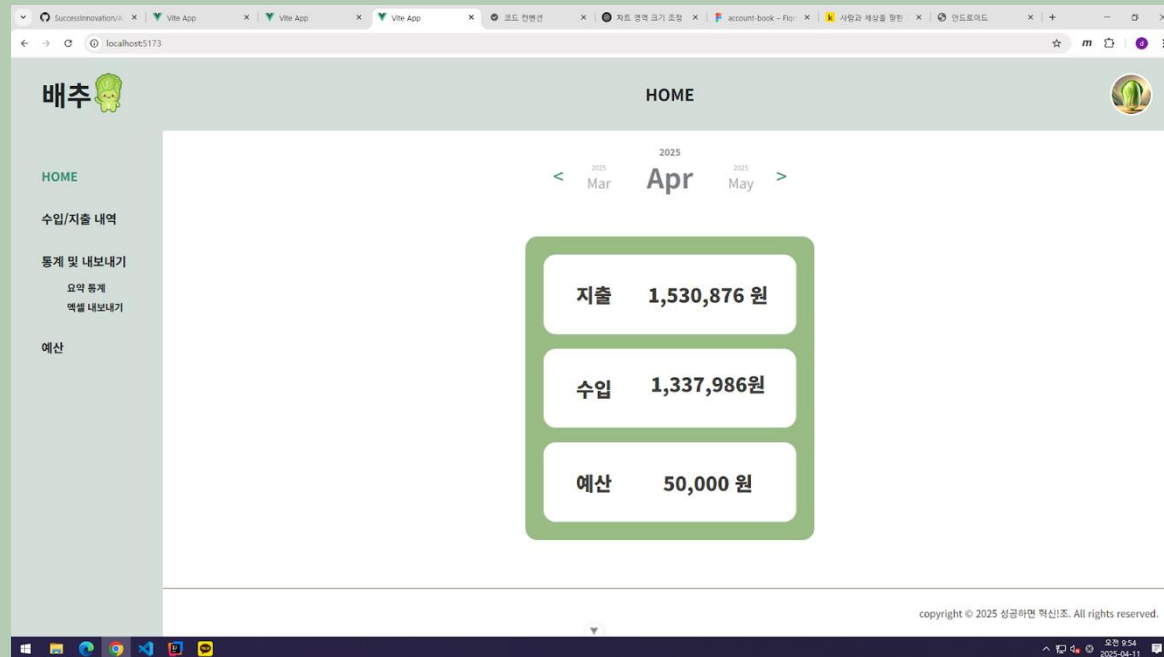
기본 화면 레이아웃

- Header, Side Bar, Footer
- 요약 카드: 이번 달 총 수입, 총 지출, 순수익을 보여주는 카드
- 최근 거래 목록: 날짜와 함께 표시되는 최근 거래 목록
- 빠른 추가 버튼: 고정된 버튼으로 신속하게 새로운 거래 추가 가능

4. 주요 화면 설계

홈 화면

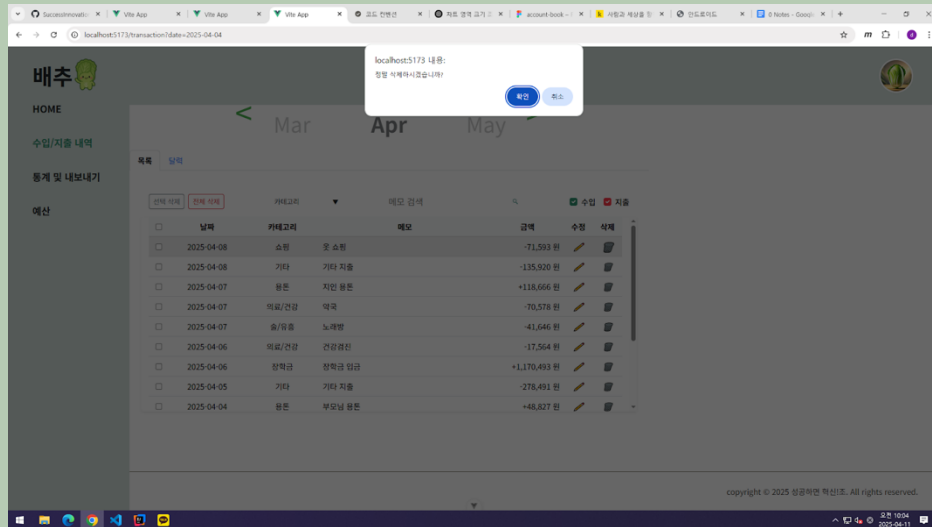
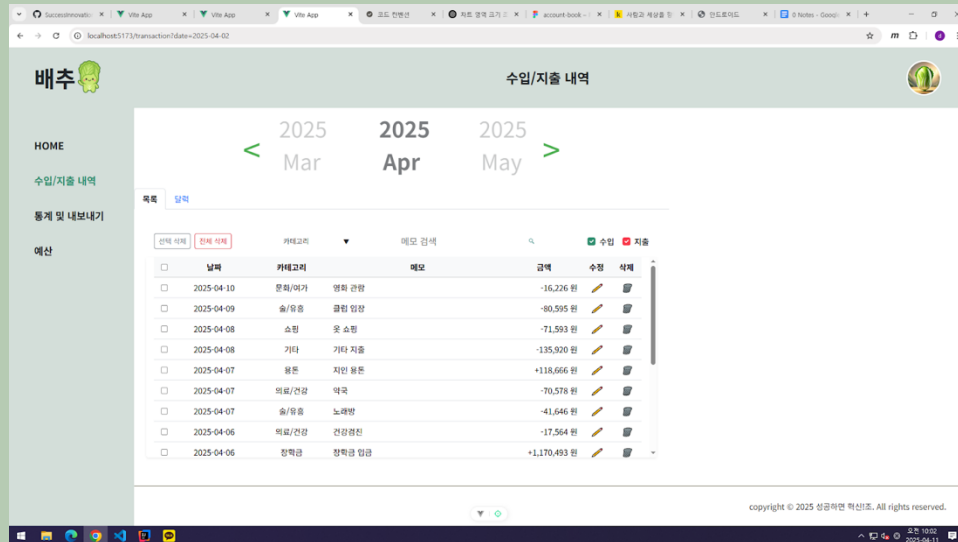
- 요약 카드: 이번 달 총 수입, 총 지출, 순수익을 보여주는 카드



4. 주요 화면 설계

수입/지출 내역 - 목록

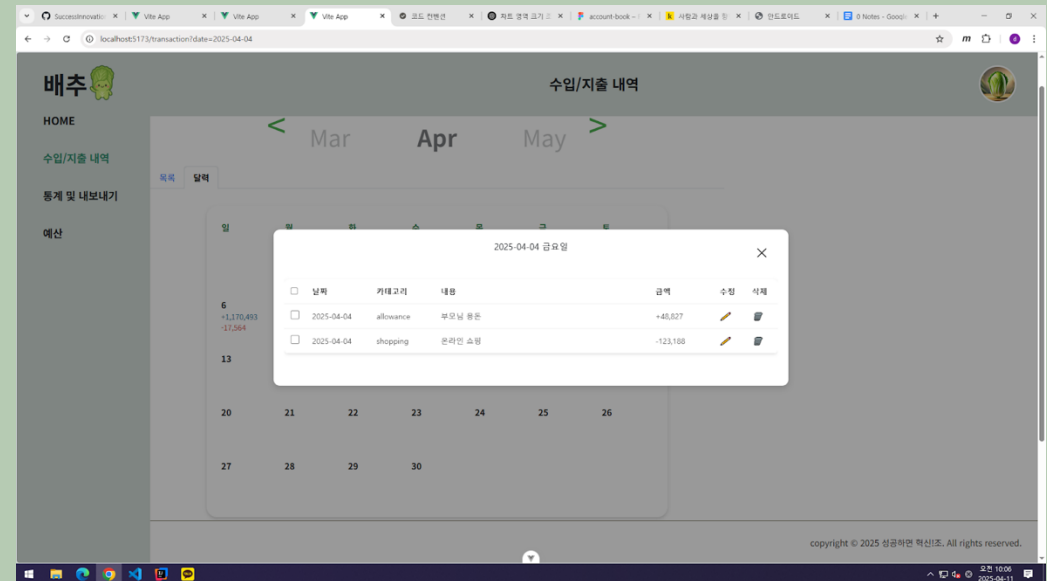
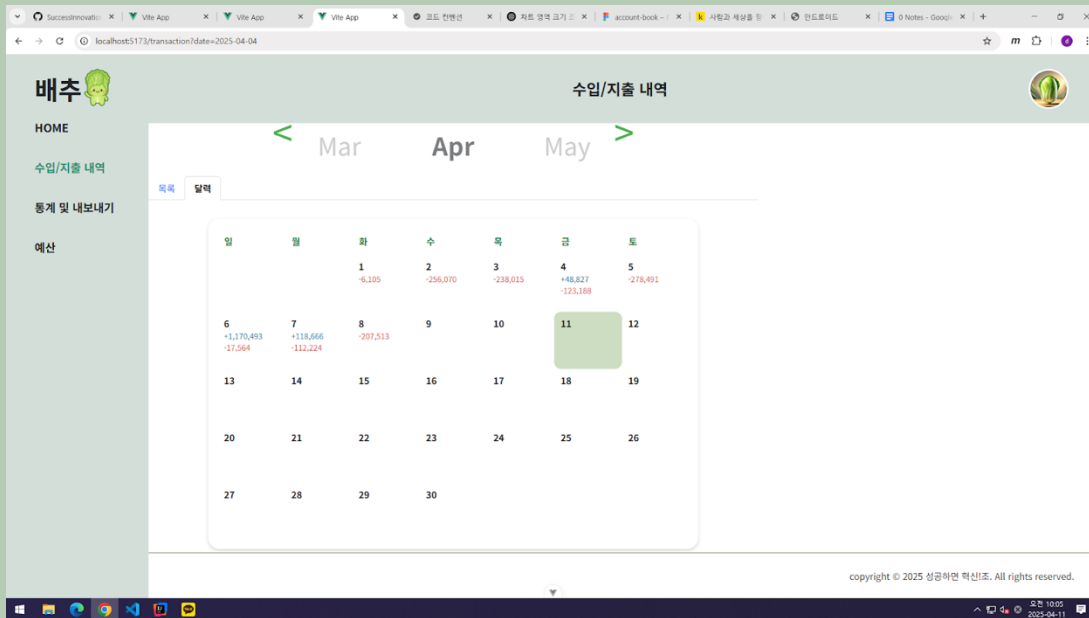
- 거래 목록: 날짜와 함께 표시되는 거래 목록
- 필터링 기능: 카테고리, 메모별로 거래 내역 가능
- 수정/삭제 기능: 해당 거래 내역 클릭 후 수정/삭제 클릭 시 수정/삭제 가능



4. 주요 화면 설계

수입/지출 내역 - 달력

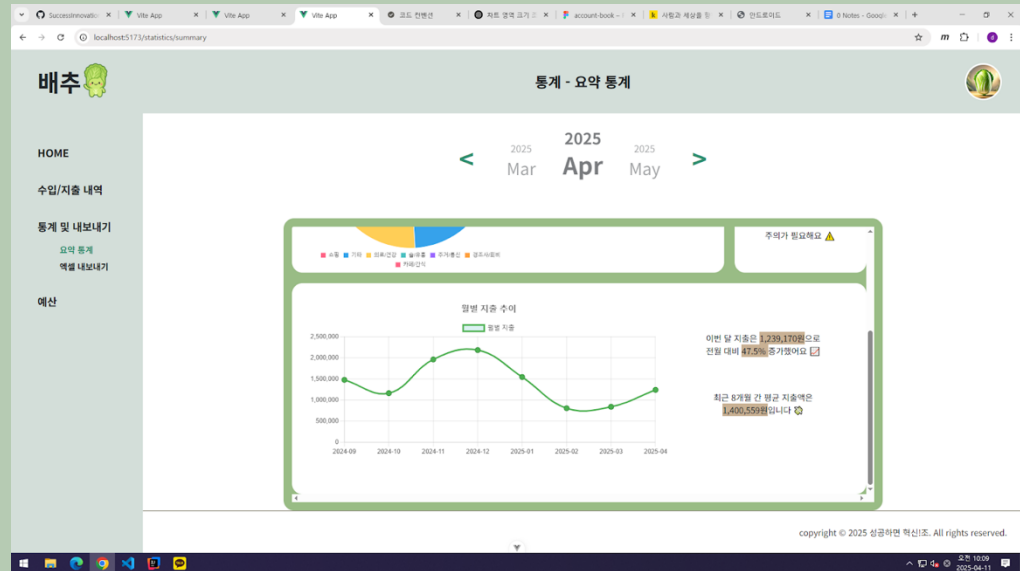
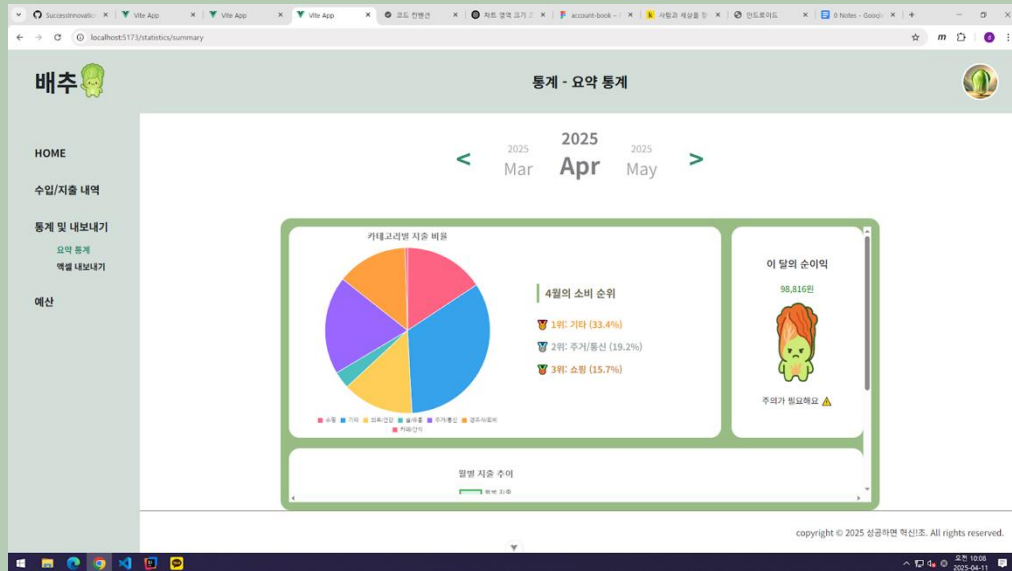
- 달력: 날짜와 함께 표시되는 거래 내역
- 수정/삭제 기능: 해당 날짜 칸 클릭 후 수정/삭제 클릭 시 수정/삭제 가능



4. 주요 화면 설계

통계 - 요약 통계

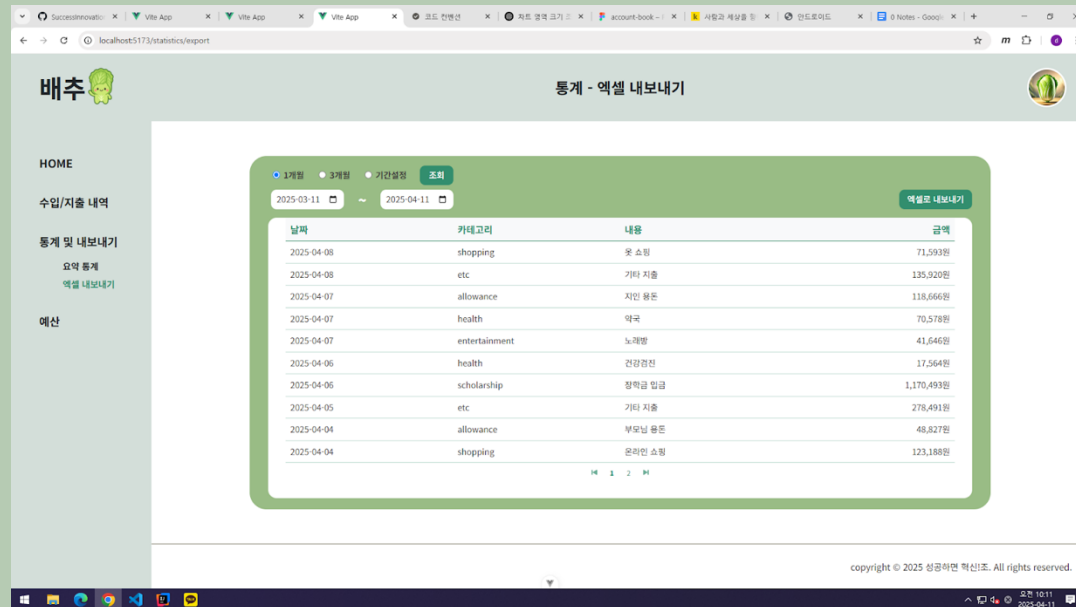
- 필터링 기능: 월별 수입 및 지출 요약
- 데이터 시각화: DB 조회하여 차트 데이터 시각화 제공



4. 주요 화면 설계

통계 - 엑셀 내보내기

- 필터링 기능: 기간별로 거래 내역 조회 가능
- 페이지네이션: 각 페이지마다 10개 거래 내역 조회
- 엑셀 내보내기: 필터링된 거래 내역 엑셀로 내보내기 가능

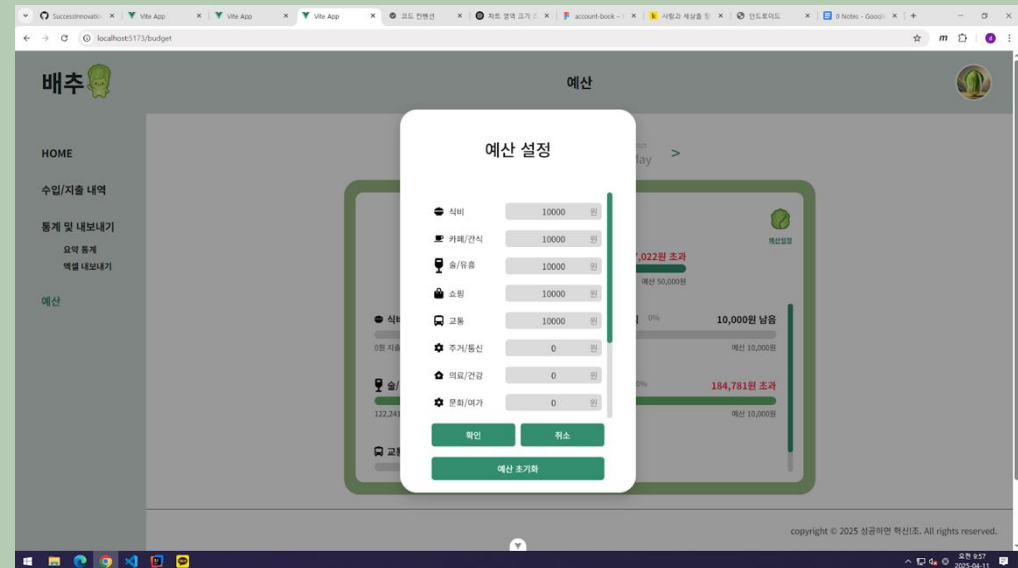
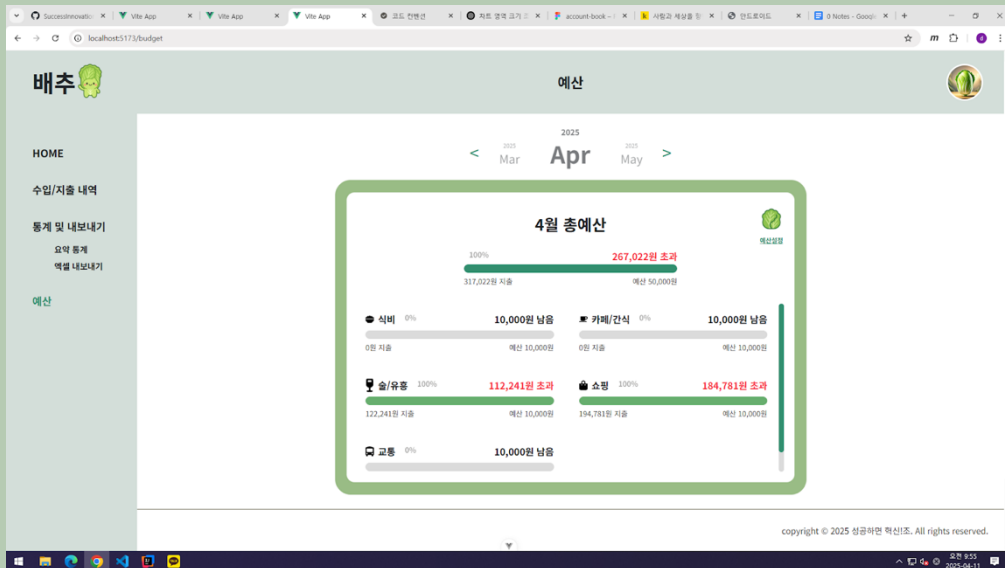


날짜	카테고리	내용	금액
2025-04-08	shopping	옷 쇼핑	71,593원
2025-04-08	etc	가타 지출	135,920원
2025-04-07	allowance	자신 용돈	118,666원
2025-04-07	health	약국	70,578원
2025-04-07	entertainment	노래방	41,646원
2025-04-06	health	건강검진	17,564원
2025-04-06	scholarship	장학금 입금	1,170,493원
2025-04-05	etc	가타 지출	278,491원
2025-04-04	allowance	부모님 용돈	48,827원
2025-04-04	shopping	온라인 쇼핑	123,188원

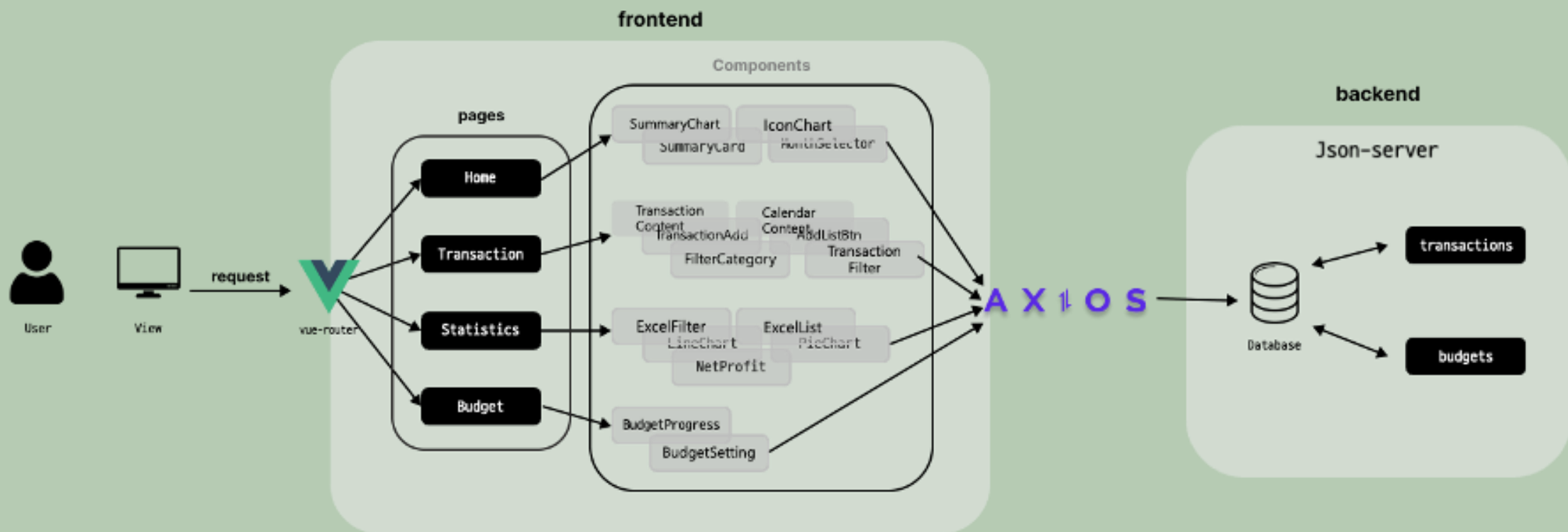
4. 주요 화면 설계

예산

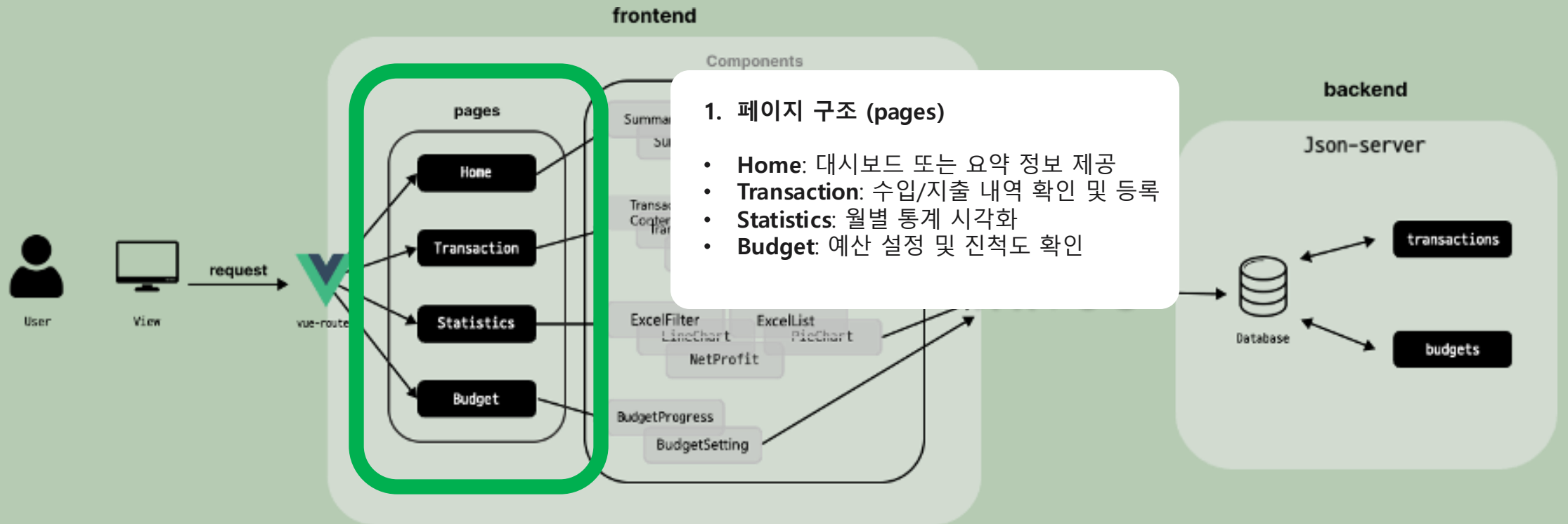
- 입력폼: 사용자 입력으로 새로운 예산 추가 및 수정 가능
- 필터링 기능: 월별 총 예산 조회 가능
- 데이터 시각화: 사용자 입력으로 실시간 데이터 시각화 제공



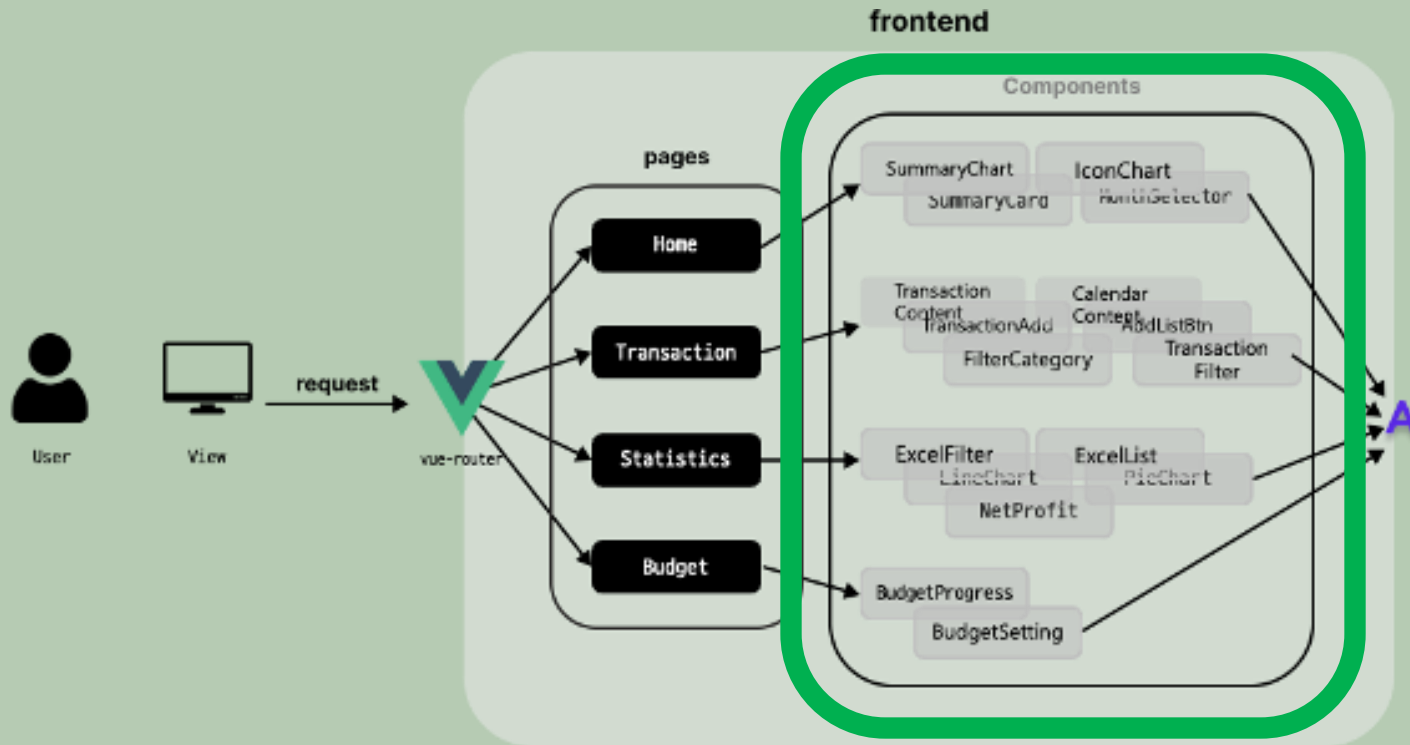
5. 컴포넌트 설계



5. 컴포넌트 설계



5. 컴포넌트 설계

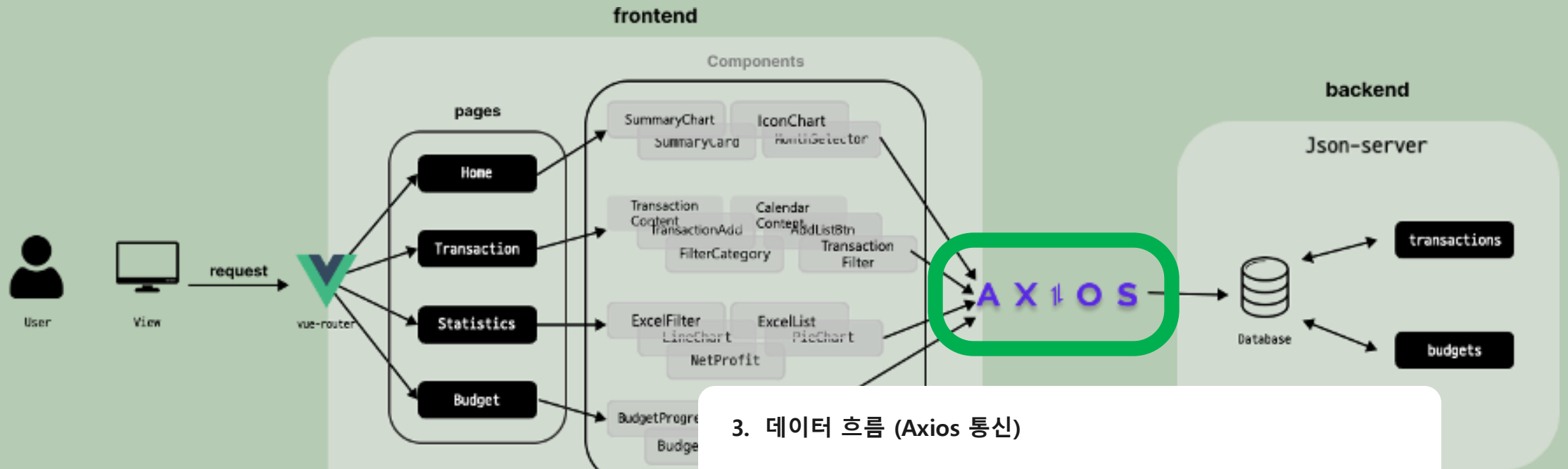


2. 컴포넌트 (Components)

예시:

- Transaction 페이지: **TransactionAdd**, **TransactionFilter** 등을 사용하여 데이터 입력과 필터링 제공
- Statistics 페이지: **PieChart**, **NetProfit**, **ExcelList** 등으로 시각화 기능 수행
- Budget 페이지: **BudgetProgress**, **BudgetSetting** 등을 통해 예산 관리 UI 제공

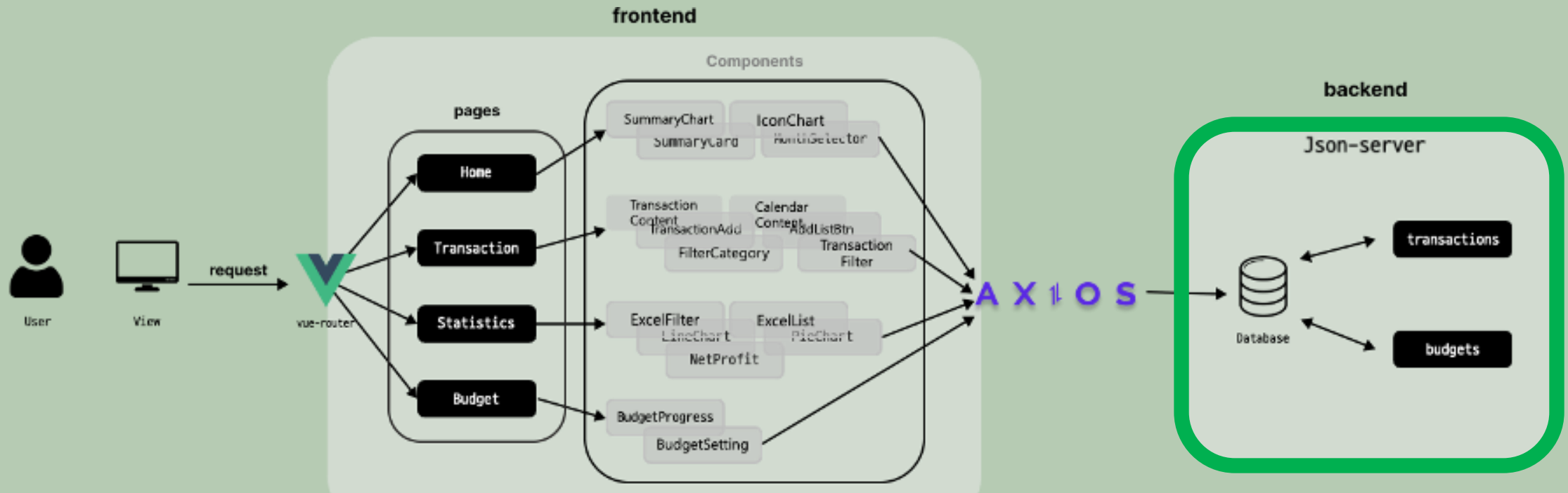
5. 컴포넌트 설계



3. 데이터 흐름 (Axios 통신)

- 모든 페이지 및 컴포넌트는 **Axios**를 통해 json-server와 통신
- (json-server는 로컬 DB로서 transactions, budgets 엔드포인트 제공)
- 이 API로부터 데이터를 가져오고 갱신하며, Vue의 상태관리(Pinia 등)를 통해 앱 전체에서 공유

5. 컴포넌트 설계



4. 백엔드 대체 - Json-server

- 복잡한 서버 없이 json-server를 통해 간단한 RESTful API 구현
- transactions와 budgets를 저장하고 관리하는 역할 수행

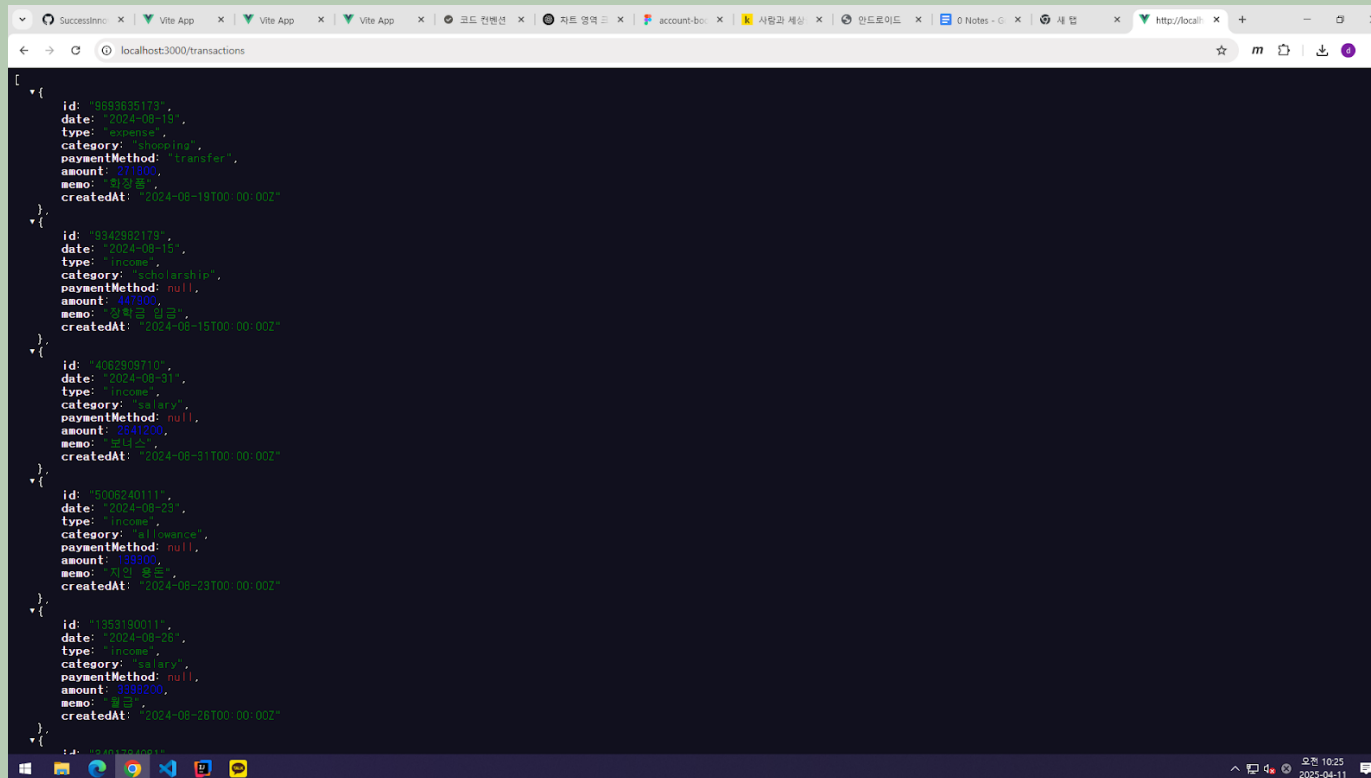


6. 구현 결과

json-server

```
[
  {
    id: "9693635173",
    date: "2024-08-19",
    type: "expense",
    category: "shopping",
    paymentMethod: "transfer",
    amount: 271800,
    memo: "화장품",
    createdAt: "2024-08-19T00:00:00Z"
  },
  {

```



6. 구현 결과

화면