최영한

데이터 엔지니어 지망 · 백엔드/데이터 파이프라인

leorld@naver.com
010-6888-8178
github.com/leorld

두손 (dooson.cc) 예비부부 결혼 준비 웹앱

1인 개발 (기획~배포) · dooson.cc 실서비스 운영 중

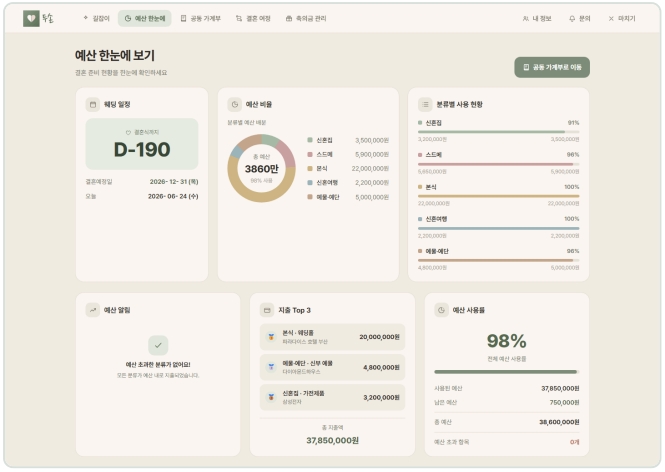
예산/지출 관리, 웨딩 로드맵, 축의금 관리, 짝꿍(파트너) 연결을 한 곳에서 — Supabase 기반 SSR 웹앱

배경 — 결혼을 준비하며 여자친구와 비용을 정리할 때 누가 얼마를 썼는지 불분명하고 정리가 번거롭다는 문제를 직접 겪고, 이를 해결하기 위해 제작해 dooson.cc로 실배포했습니다.

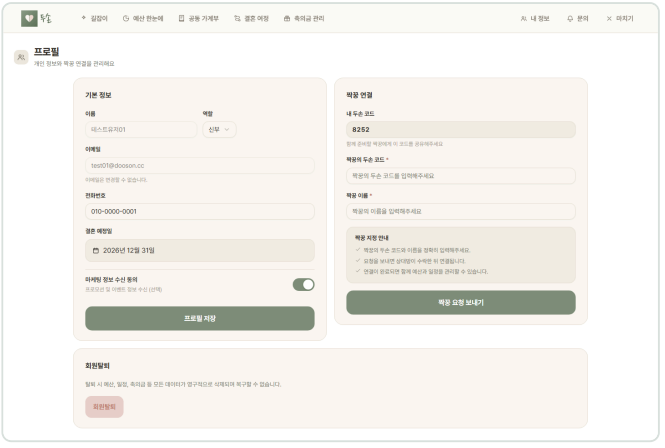
- **짝꿍(Partnership) 데이터 모델** — 신랑·신부 두 계정을 코드 공유로 연결하는 partnership 테이블을 핵심 관계로 설계, 예산/일정 등 대부분 기능을 개인 단위가 아닌 partnership 단위로 스코프
- **Supabase RLS 보안 설계** — 본인 행만 접근해야 하는 테이블은 강하게 제한하고, 파트너십으로 연결된 두 사용자가 서로의 예산/지출을 봐야 하는 테이블은 partnership 단위로 정책을 풀어주는 식으로 — 어떤 테이블에 RLS를 걸고 어떤 테이블을 풀어야 하는지 직접 판단하고 정책을 적용
- 기능 모듈 단위(schema → queries → mutations → pages)로 일관된 아키텍처 적용, React Router v7 SSR + Drizzle ORM, 카카오 소셜 로그인 · 영수증 카메라 업로드 등 세부 UX 구현

React 19 · React Router v7 (SSR) · TypeScript · Drizzle ORM · Supabase (Auth/Postgres) · TailwindCSS 4 · Vercel

성과 — dooson.cc 실서비스 배포 및 운영 중(현재 본인 사용 단계). 이후 모바일 앱(dooson-app)으로 서비스 범위를 확장.



웨딩 예산 한눈에 보기



프로필 / 짝꿍 연결

두손 모바일 앱 (dooson-app) Expo WebView 앱 · 출시 도전

1인 개발 · Google Play 제출 준비 단계

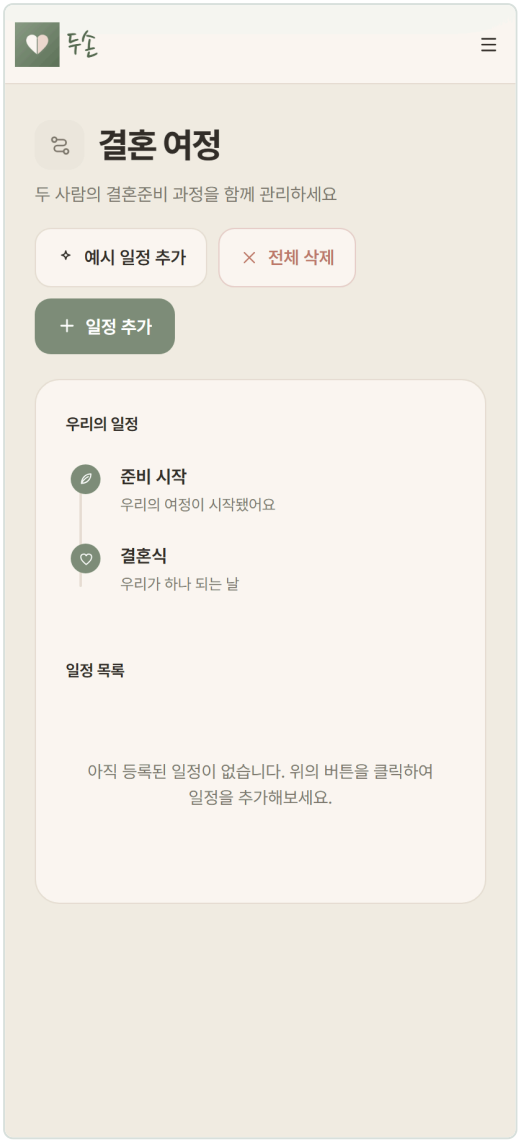
이미 운영 중인 dooson.cc를 앱스토어에도 출시해보고 싶어서 시작한 도전 프로젝트

배경 — 꼭 필요해서 시작한 프로젝트는 아니었습니다. 웹서비스를 실제 앱스토어 출시 파이프라인까지 끝까지 끌고 가보고 싶다는 도전 의식으로 시작했고, 그 과정에서 웹과는 다른 종류의 문제들을 직접 부딪히며 풀었습니다.

- **이종 인증 메커니즘 연동(native-bridge)** — 네이티브 앱(SecureStore 토큰)과 WebView 안의 웹서비스(쿠키 기반 Supabase 세션)는 서로 다른 인증 메커니즘을 쓰기 때문에 자동으로 공유되지 않는 문제를, 토큰을 웹 쪽에 전달해 쿠키를 재발급받는 브릿지로 해결. 네트워크 실패와 인증 실패(토큰 만료)를 구분해 각각 다른 복구 전략을 적용
- Expo Push Token 등록/딥링크 이동, 영수증 카메라 네이티브 연동, Android 뒤로가기·세션 만료 자동 복구 등 WebView 앱에서 흔히 겪는 디테일 처리
- EAS Android 빌드, 스토어 등록정보·그래픽 자산·데이터 안전 섹션 등 Google Play 제출 자료 준비

Expo / React Native | TypeScript | Supabase | EAS Build

성과 — 실기기 빌드로 로그인/푸시/카메라/딥링크 흐름 검증 완료, Google Play 제출 준비 단계. 기능의 완성도보다 "웹 서비스를 모바일 배포 파이프라인까지 끝까지 끌고 가본 경험" 자체에 의의를 둔 프로젝트입니다.



실기기(Android) 캡쳐 — 웨딩 로드맵

실기기(Android) 캡쳐 — 축의금 관리

제로뿌 가계경영 대시보드 개인 가계부 데이터 플랫폼

1인 개발 (기획~배포) · 2026.06 · 자가 호스팅 운영 중

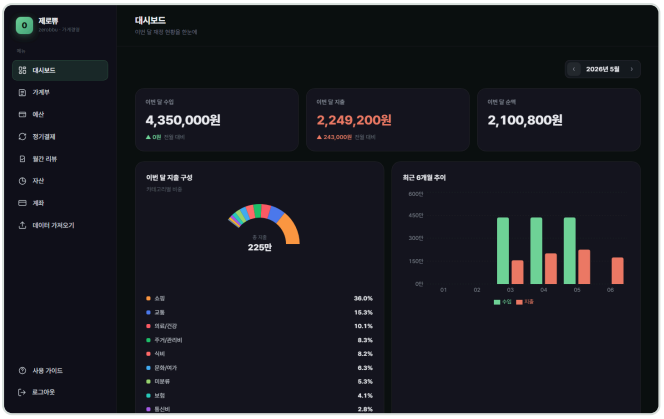
은행·카드 거래내역 CSV를 업로드하면 자동 분류 → 정기결제 탐지 → 예산 추천까지 이어지는 부부 전용 가계부

배경 — 결혼 후 둘이 가계부를 정리하면서, 매달 반복되는 고정비·구독료를 손으로 추적하는 게 번거로워 우리 두 사람의 소비 패턴에 맞춘 전용 대시보드를 직접 만들었습니다.

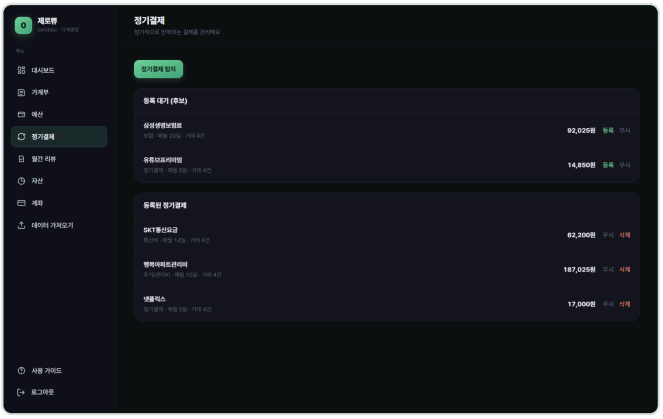
- **파서 레지스트리 설계** — 은행/카드사별로 CSV 포맷이 제각각인 문제를, 공통 인터페이스(base parser) + 은행별 구현(KB카드 전용 파서, 범용 CSV 매핑 파서)으로 분리해 신규 금융사 추가 시 파서만 추가하면 되는 구조로 설계
- **학습형 자동 분류 엔진** — 키워드 규칙 기반으로 거래를 자동 분류하고, 사용자가 직접 카테고리를 수정하면 그 피드백을 우선순위가 높은 규칙으로 승격(upsert)시켜 이후 동일 가맹점은 자동으로 분류되도록 구현
- **정기결제 자동 탐지 알고리즘** — 거래 설명을 정규화한 뒤 발생 횟수·금액 변동률·결제일 변동을 기준으로 패턴을 클러스터링해 정기결제 후보를 자동 탐지 (등록대기 → 등록 상태로 관리)
- **예산 추천 & 월간 리뷰** — 최근 3개월 평균 지출 기반 예산 추천, 전월 대비 증감·예산 초과· 순자산 변동을 자동 요약하는 월간 리뷰 생성 로직

Python FastAPI SQLAlchemy Alembic Pandas React TypeScript Vite SQLite Docker / Caddy

성과 — 본인과 배우자가 실제로 매월 사용 중. CSV 업로드 한 번으로 분류·정기결제·예산 점검까지 자동화되어 기존에 수기로 하던 월말 가계부 정리 과정을 대체.

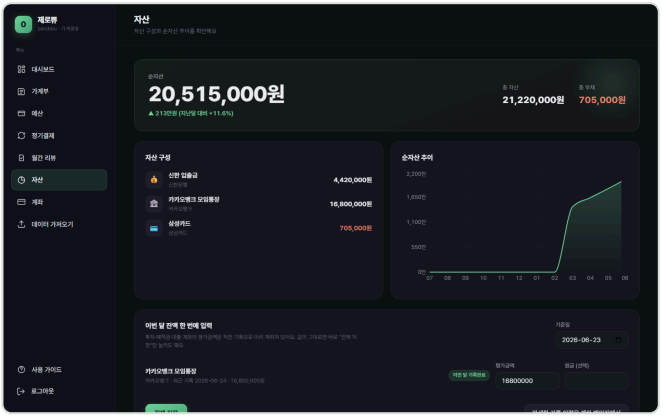


월별 수입/지출 대시보드



정기결제 자동 탐지

카테고리별 예산 추적



순자산 추이

공통 역량 — 1인 기획~배포 풀스택 경험(웹/모바일/백엔드), 데이터 모델링 및 보안 정책(RLS) 설계, 이종 시스템(네이티브 앱 ↔ 웹, 은행사별 데이터 포맷) 간 연동, 규칙 기반 분류/탐지 로직 설계.