

Simple Budget Tracker

"Simple tools for the busiest hands."

Project Overview

The Simple Budget Tracker is a full-stack, production-ready SaaS platform engineered to reduce the cognitive load of household financial management. Designed specifically for busy families and household managers, the application provides a frictionless, zero-risk trial experience that aggregates financial data into actionable, human-readable intelligence.

Project Documentation Note: This document serves as a professional portfolio overview of the architecture and engineering decisions behind the application. For supplementary historical context and original planning notes, please refer to the original design document, *Simple Budget Track.pdf*, verbatim.

Core Technology Stack

The platform is built upon a robust, custom-engineered WordPress architecture, eschewing heavy visual builders in favor of bespoke PHP, strict MySQL data structures, and a lightweight Vanilla JavaScript frontend.

Backend Engine

Object-Oriented PHP 8.x
Custom MySQL Tables

Frontend UX

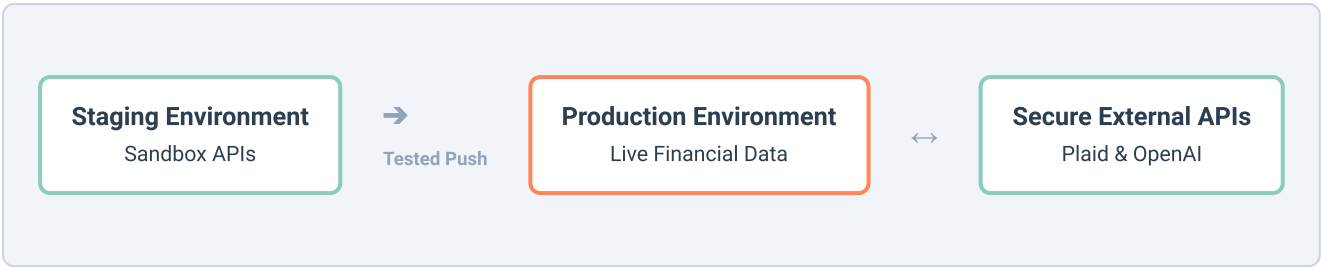
Vanilla JavaScript
AJAX Architecture
Chart.js

External APIs

Plaid (Bank Sync)
OpenAI GPT-4o
Google OAuth 2.0

Cloud Architecture & Deployment

The application utilizes a dual-environment cloud strategy to ensure zero-downtime updates and safe testing of financial API integrations and AI prompt engineering. High performance is maintained via custom caching headers specifically targeted to bypass static caching on dynamic dashboard routes.



Ecosystem Integrations

The core plugin acts as an orchestrator, securely passing data between isolated systems while maintaining a unified interface for the end user. High-specificity CSS guarantees that the bespoke application UI is never corrupted by the underlying host theme.

Authentication Flow (Google OAuth): Hooks intercept the standard login redirect, evaluating the user's subscription and trial status (via WooCommerce logic) to route them intelligently to an onboarding flow or their immediate financial dashboard.

Data Parsing Engine: A custom asynchronous extraction module handles uploaded bank statements. By utilizing specialized regular expressions and a custom OpenAI prompt structure, it parses both structured CSVs and complex, multi-page PDF bank statements directly in the browser.

Security & Intelligence Layers

Handling live financial data requires banking-grade security protocols. The application was built with a "zero-trust" philosophy regarding sensitive keys and user tokens. Furthermore, raw financial data is instantly transformed into actionable insights via an integrated AI Advisor.

AES-256-CBC Encryption

All API keys and Plaid access tokens are encrypted at the application layer before touching the MySQL database. Keys are decrypted on-the-fly in server memory only when actively communicating with external APIs.

Plaid Financial Sync

Integrates the Plaid Link flow to securely exchange public tokens for persistent access. An automated background worker routinely requests 30-day transaction windows, filtering against the database to prevent duplicate entries.

Smart Categorization

Users can define custom categorization rules. The database automatically intercepts incoming bank descriptions and maps them to user-defined expense categories before they render on the frontend.

AI Financial Advisor

Aggregates 90 days of spending velocity alongside demographic data (household size, housing goals). It builds a strictly constrained context prompt for GPT-4o, generating a personalized, highly relevant financial roadmap.

Data Integrity & Rate Limiting

To protect API budgets and ensure database hygiene, a dedicated usage-tracking table acts as a governor for the AI integrations, enforcing a 30-day cooldown on expensive LLM generation. Additionally, custom database repair scripts are exposed to the user to automatically detect, merge, and resolve duplicate category definitions without developer intervention.

Engineering Challenges & Solutions

Technical Challenge	Architectural Solution
State & UI Conflicts	Utilized high-specificity selectors and decoupled the frontend DOM updates from traditional PHP page reloads, relying entirely on asynchronous AJAX handshakes to provide a seamless "app-like" feel.
Expensive LLM API Calls	Engineered a custom rate-limiting architecture via a dedicated SQL table (sbt_ai_usage) to enforce a strict 30-day generation cooldown, protecting backend operating margins.
Bank Data Security	Developed a custom PHP encryption wrapper utilizing openssl_encrypt with AES-256. Plaid tokens and system API keys are never stored in plain text, securing them against database breaches.
PDF Statement Parsing	Implemented a hybrid local/cloud extraction method. Browser-side libraries extract raw text to save server bandwidth, which is then structured via a specialized, prompt-engineered AI endpoint that filters out document noise.

Professional Highlights Summary

- Full-Stack Development:** Engineered a production-ready, highly secure SaaS platform utilizing modern Object-Oriented PHP, strictly typed MySQL, and interactive Vanilla JavaScript.
- Financial Systems Integration:** Successfully navigated complex OAuth handshakes and token-exchange protocols to implement a secure, live data pipeline via the Plaid API.
- AI Implementation:** Integrated cutting-edge LLMs (OpenAI GPT-4o) to transform raw, unstructured user data into valuable, human-readable insights.
- Product Strategy & UX:** Maintained a strict focus on the target demographic ("the mental load of household managers"), ensuring zero-friction onboarding, automated data cleanup, and highly readable reporting interfaces.