
Project Proposals

To: YAAAY365

From: BearPlex

Date: August 22, 2025

Project 1: Accelerated Migration of YAAAY365 Authentication Infrastructure

1. Executive Summary

YAAAY365's strategic pivot to a free-to-use model necessitates an immediate infrastructure change to manage exponential user growth without incurring unsustainable costs from Auth0. Their per-user pricing model directly conflicts with our goal of mass adoption.

This proposal outlines a hyper-efficient migration to a custom authentication infrastructure built on our Supabase backend:

- **Eliminates variable per-user costs** → fixed, predictable expenses.
- **Unifies the technology stack** → simpler development and full ownership of the user journey.
- **Enables scalability** → future-proofing YAAAY365 for rapid growth.

Through a parallelized workstream and deep system familiarity, this complex migration can be delivered efficiently.

Total Effort: 90 hours

Timeline: 2–3 weeks

This is a critical business decision that secures both the financial and technical foundation of YAAAY365.

2. Detailed Scope of Work & Hour Estimation

Phase 1: Backend API Development & Security Hardening (30 Hours)

- **Task 1.1: Database Schema & Security (4 Hours)**
 - Add `password_hash` column to `public.users`.
 - Implement Row-Level Security (RLS).
- **Task 1.2: Secure Registration Endpoint `/auth/register` (8 Hours)**
 - Input validation (name, email, password strength).
 - Password hashing & salting (bcrypt).
 - User record creation in DB.
- **Task 1.3: Secure Login Endpoint `/auth/login` (8 Hours)**
 - Password hash comparison.
 - Generate/sign JWTs (short-lived access + refresh).
- **Task 1.4: Secure Token Refresh Endpoint `/auth/refresh` (4 Hours)**
 - Validate refresh tokens, check blacklist, issue new access token.
- **Task 1.5: Password Reset Flow (6 Hours)**
 - `/auth/forgot-password`: issue secure token.
 - `/auth/reset-password`: validate token, update hash.

Phase 2: Frontend & Mobile Application Integration (32 Hours)

- **Task 2.1: Core Auth Service Module (6 Hours)**
 - Unified cross-platform service for requests, tokens, session handling.
- **Task 2.2: Customer Website (Next.js) Integration (14 Hours)**
 - Remove Auth0 SDK/dependencies.

- Integrate new auth service into login, register, forgot password.
- Implement httpOnly cookie management for refresh tokens.
- Refactor session state + protected routes.
- **Task 2.3: Mobile App (React Native) Integration (12 Hours)**
 - Remove `react-native-auth0`.
 - Integrate new service into all auth screens.
 - Secure storage: Keychain (iOS) / Keystore (Android).

Phase 3: Seamless & Secure User Migration (14 Hours)

- **Task 3.1: Data Export & Mapping Script (4 Hours)**
 - Export profiles (no passwords) from Auth0 API.
- **Task 3.2: Lazy Migration Logic (8 Hours)**
 - If no password hash exists → validate against Auth0 one final time.
 - On success → hash with bcrypt, save locally.
 - Ensures frictionless migration.
- **Task 3.3: Dry-Run & Validation (2 Hours)**
 - Full test import in staging environment.

Phase 4: QA, Deployment & Decommissioning (14 Hours)

- **Task 4.1: Testing & Security Review (8 Hours)**
 - Validate all flows (register, login, reset password, API access).
 - Security audit of all endpoints.
- **Task 4.2: Production Deployment (4 Hours)**

- Execute deployment checklist.
 - Run user import script.
 - Monitor stability.
 - **Task 4.3: Decommission Auth0 (2 Hours)**
 - Remove integrations & cancel subscription.
-

3. Estimated Project Timeline & Total Effort

Phase	Estimated Hours
Phase 1: Backend Development & Security	30
Phase 2: Frontend & Mobile Integration	32
Phase 3: User Migration Strategy	14
Phase 4: QA & Decommissioning	14
Total Effort	90 Hours

Timeline: 2–3 weeks (dedicated team).

4. Benefits of This Migration

- **Immediate Cost Elimination** – End Auth0 billing.
 - **Full Ownership & Control** – Direct auth management.
 - **Architectural Simplification** – Unified stack reduces technical debt.
-

Project 2: Introducing "Pay with Cash" Option

1. Executive Summary

YAA365 currently only supports in-app wallet payments, excluding customers who prefer cash. This limits adoption and results in incomplete financial tracking.

This proposal introduces “**Pay with Cash**” as an additional option:

- Greater flexibility → increases redemption success.
- Logged transactions → complete financial transparency.

Effort: 50 hours

Timeline: 1–2 weeks

2. Background & Problem Statement

- **Payment Inflexibility:** Wallet-only limits adoption.
 - **Incomplete Tracking:** Cash transactions unlogged → skewed GTV.
-

3. Proposed Solution: A Seamless "Pay with Cash" Flow

- **User Story:** Members pay in cash at stores and redeem seamlessly.
 - **Partner Story:** Cash redemptions logged for transparency.
-

4. Scope of Work & Estimation

Phase 1: Backend Enhancements (14 Hours)

- Add `payment_method` column.
- Refactor redemption logic.

Phase 2: Mobile App UX (24 Hours)

- Add payment selection UI.
- Build “Pay with Cash” flow.
- API integration.

Phase 3: Partner & Admin Ledger Enhancements (6 Hours)

- Add payment method column & filters.

Phase 4: QA & Deployment (6 Hours)

- Test all flows.
- Deploy updates.

5. Estimated Project Timeline & Total Effort

Phase	Hours
Backend Enhancements	14
Mobile App UX	24
Ledger Enhancements	6
QA & Deployment	6
Total Effort	50

Timeline: 1–2 weeks

6. Benefits

- **Increased Transactions** – Capture cash users.
- **Better UX** – Flexible payment options.

- **Accurate Reporting** – Complete financial data.
 - **Partner Trust** – Transparent ledgers.
-

Project 3: Single Member Upload Functionality (Accelerated)

1. Executive Summary

Current CSV imports are inefficient for single entries. This project adds a “**Single Member Add**” modal for faster onboarding.

Effort: 15 hours

Timeline: ~1 week

2. Background & Problem Statement

- **Inefficient for One-Offs:** CSV required for single additions.
 - **Missed Opportunities:** Delays onboarding.
 - **Barrier to Adoption:** Complex for non-technical partners.
-

3. Proposed Solution

- Add “Add Member” button.
 - Simple modal with Name, Email, Phone.
 - Direct API integration → instant group membership.
-

4. Scope of Work & Estimation

Phase 1: Backend API Enhancement (7 Hours)

- Adapt existing bulk import endpoint.
- Ensure schema compatibility.

Phase 2: Partner Portal Frontend (4 Hours)

- Build responsive modal.
- Add “Add Member” button.
- API integration & feedback states.

Phase 3: QA & Deployment (4 Hours)

- Test all cases (new user, existing user, invalid email).
- Deploy.

5. Estimated Project Timeline & Total Effort

Phase	Hours
Backend Enhancement	7
Frontend Development	4
QA & Deployment	4
Total Effort	15

Timeline: ~1 week

6. Benefits

- **Efficiency** – Quick onboarding.
 - **Partner-Friendly** – Intuitive workflows.
 - **Faster Adoption** – Easier for partners to use portal.
-