TECH STACK DOCUMENT

Profit-First Ad Manager (PFAM)

Student Build Edition · Budget-Optimised · Two Options

v1.0 | February 2026 | Internal

Field	Value
Team	College student group
Philosophy	Free tiers first → student credits → pay only when you have paying customers
Stack A	The JS Monorepo — Next.js + Supabase (one language, fastest to ship)
Stack B	The Python Route — Next.js + FastAPI + Neon (better for data/ML, slightly more setup)
Estimated cost to first customer	$0 on either stack

1. Claim These Credits First — Do This Today

Before writing a single line of code, claim every credit below. Combined value: $500+ in free infra. Some require a .edu email or proof of enrollment.

Program	What You Get	How to Claim
GitHub Student Developer Pack	Free Vercel Pro, Figma Pro, Sentry Team, Namecheap domain, 30+ tools	education.github.com → verify with .edu email (1–7 day approval)
AWS Educate	$100 AWS credits, no credit card required	aws.amazon.com/education/awseducate → .edu email
Google Cloud Free Trial	$300 credits (90 days) + free tier that never expires	console.cloud.google.com → free trial, no student ID needed
Vercel (via GitHub Pack)	Free Pro plan (normally $20/mo) — better bandwidth + team features	Redeem inside GitHub Student Pack dashboard
Railway Student	Extra $10/mo compute credits on top of the $5/mo free tier	railway.app → connect GitHub → claim Student Pack benefit
Sentry (via GitHub Pack)	Free Team plan (normally $26/mo) — error tracking + replays	Redeem inside GitHub Student Pack dashboard

Permanently Free Tiers (No Expiry, No Card)

Service	Free Tier	What It Covers for PFAM
Supabase	2 projects, 500MB DB, 50K auth users, 1GB storage	Database + Auth + Realtime for Stack A — covers you to ~200 users
Neon	0.5GB Postgres, scales to zero (no idle cost)	Database for Stack B — no charges when dev DB is inactive
Upstash	10K Redis requests/day, 256MB	Caching + rate limiting on either stack
Clerk	10,000 monthly active users	Auth for Stack B — generous enough for all of Year 1
Cloudflare R2	10GB storage, 1M ops/month, zero egress fees	File storage (CSV exports, reports) — free effectively forever
Resend	3,000 emails/month	Notification emails — covers ~100 active users

Service	Free Tier	What It Covers for PFAM
PostHog	1M events/month	Product analytics — free forever at student scale
Stripe	No monthly fee — 2.9% + $0.30 per transaction only	Payments — you pay nothing until you earn money

2. Stack A vs Stack B — Pick One and Commit

Stack A
The JS Monorepo
Next.js + Supabase

Stack B
The Python Route
Next.js + FastAPI + Neon

Stack A: Everything in TypeScript/JavaScript — one language front-to-back. Best if your whole team knows JS/React. Fastest to ship.

Stack B: JS frontend + Python backend. More setup, but Python is purpose-built for the data pipeline (attribution engine, profit calc, ML). Best if anyone on the team knows Python.

	Stack A — JS Monorepo	Stack B — Python Route	Edge
Languages	TypeScript only	TypeScript (frontend) + Python (backend)	**A**
Ship speed	Fast — one codebase	Slower — two repos	**A**
Data pipeline	Doable but JS not ideal	Python pandas/polars is built for this	**B**
ML model (Tier 5 attribution)	Hard — needs Python service anyway	Native — scikit-learn + XGBoost run directly	**B**
DevOps complexity	Near-zero (Vercel + Supabase)	Low (Railway reduces pain)	**A**
Monthly cost at launch	$0	$0–$5	**A**
Industry alignment	Good	Excellent	**B**
Choose if your team knows…	JavaScript / React / Node	Python + at least one JS dev	—

Recommendation: Stack B if anyone knows Python — the data processing is the hardest part of PFAM and Python makes it significantly easier. Stack A if you're a JS-only team — it's perfectly capable of building a great v1 and will ship faster.

3. Stack A — Full Specification (JS Monorepo)

One language (TypeScript), one repo, minimal DevOps. Supabase handles database, auth, realtime, and storage. Vercel handles hosting. You focus on the product.

Layer	Tool	Cost	Why
Frontend + API	Next.js 14 (App Router)	Free (open source)	React SSR + API routes in one repo. Industry-standard SaaS frontend.
Hosting	Vercel	Free → Pro via GitHub Pack	Zero-config deploys. Preview URL per PR. Perfect for Next.js.
UI Components	shadcn/ui + Tailwind CSS	Free (open source)	Copy-paste components, no lock-in, fully customisable.
Database	Supabase (PostgreSQL 15)	Free — 500MB, 2 projects	Managed Postgres with dashboard. Row-level security built in.
Auth	Supabase Auth	Free — 50K MAU	OAuth, magic links, JWTs — all included in Supabase. Zero setup.
Realtime	Supabase Realtime	Free — 200 connections	Live dashboard updates (sync status, automation feed) via WebSockets.
File Storage	Supabase Storage	Free — 1GB	CSV exports, PDF reports. S3-compatible API.
Background Jobs	Inngest	Free — 50K runs/month	Durable background jobs in TypeScript. No separate worker server needed. Handles sync jobs, profit calc, rule evaluation.
Cache / Rate Limiting	Upstash Redis	Free — 10K req/day	Serverless Redis. Dashboard cache, API rate limiting, session state.
Email	Resend	Free — 3K emails/month	Modern transactional email. React Email templates.
ORM	Drizzle ORM	Free (open source)	Type-safe SQL for TypeScript. Works perfectly with Supabase/Postgres.
Error Tracking	Sentry	Free via GitHub Pack	Catch and debug production errors with full stack traces.
Payments	Stripe	No monthly fee	2.9% + $0.30/txn only. Subscription management + Customer Portal.
CI/CD	GitHub Actions	Free — 2K min/month	Auto-run tests + deploy on every PR.
Analytics	PostHog	Free — 1M events/month	User funnels, feature flags, session recordings.

Stack A — Repo Structure (Simplified)

```
pfam/ ├── src/app/ │   ├── (dashboard)/      → campaigns/, automation/, settings/ │   └── api/
→ connect/ (OAuth), webhooks/, stripe/ ├── src/lib/ │   ├── connectors/       → Shopify, Meta,
Google API clients │   ├── attribution/      → Attribution engine (TS, Tiers 1–4) │   ├── profit/
→ Profit calculation engine (TS) │   └── rules/           → Rules evaluation engine (TS) ├──
src/inngest/          → Sync jobs, profit calc, rule eval workers └── supabase/migrations/   → SQL
schema migration files
```

4. Stack B — Full Specification (Python Route)

Next.js for the frontend. Python FastAPI for all data logic. Neon for serverless Postgres. Deployed on Railway. Best alignment with the SRS architecture — Python is purpose-built for data pipelines and ML.

Layer	Tool	Cost	Why
Frontend	Next.js 14 (App Router)	Free (open source)	Same as Stack A. No reason to change the frontend.
Frontend Hosting	Vercel	Free → Pro via GitHub Pack	Same as Stack A.
UI Components	shadcn/ui + Tailwind CSS	Free (open source)	Same as Stack A.
Auth	Clerk	Free — 10K MAU forever	Best auth DX available. Drop-in Next.js middleware. Verifies JWTs in FastAPI too.
Database	Neon (Serverless Postgres)	Free — 0.5GB, scales to zero	No idle cost — dev DB charges you nothing when inactive. DB branching per PR is a great dev feature.
Backend API	FastAPI (Python 3.12)	Free (open source)	Async Python API. Auto-generates OpenAPI docs. Industry standard for data-heavy SaaS backends.
Backend Hosting	Railway	$0 with Student Pack credits	Git push → deployed. Supports always-on API + Celery workers + cron — all in one platform.
Background Workers	Celery + Upstash Redis (as broker)	Free tier covers dev + beta	Distributed task queue for data sync, profit calc, rule evaluation. Celery = Python's standard worker system.
File Storage	Cloudflare R2	Free — 10GB, zero egress	S3-compatible. CSV/PDF storage. Zero egress fees — unlike AWS S3.
Data Processing	Pandas + Polars	Free (open source)	Pandas for data wrangling. Polars as faster alternative for large orders datasets.
ML (Attribution Tier 5)	scikit-learn + XGBoost	Free (open source)	Attribution ML model. scikit-learn for preprocessing, XGBoost for gradient boosting.
ORM	SQLAlchemy 2.0 + Alembic	Free (open source)	Industry-standard Python ORM. Alembic manages schema migrations.
Email	Resend	Free — 3K emails/month	Same as Stack A. Python SDK available.

Layer	Tool	Cost	Why
Error Tracking	Sentry	Free via GitHub Pack	Python SDK available. Same setup as Stack A.
Payments	Stripe	No monthly fee	Same as Stack A.
CI/CD	GitHub Actions	Free — 2K min/month	Two workflows: one for Vercel (frontend), one for Railway (backend).

Stack B — Repo Structure (Simplified)

```
pfam-frontend/  (→ Vercel) ├── src/app/            → All Next.js pages └── src/lib/api.ts      → Typed client that calls the FastAPI backend  pfam-backend/   (→ Railway) ├── app/routers/       → connectors.py, campaigns.py, rules.py, billing.py ├── app/services/ │   ├── attribution/    → 5-tier attribution engine │   ├── profit/         → Profit calculation engine │   ├── rules/          → Rules evaluation engine │   └── connectors/     → Shopify, Meta, Google clients ├── app/workers/ → Celery tasks (sync_shopify, calc_profit, eval_rules) ├── alembic/           → Database schema migrations Procfile              → web=uvicorn  worker=celery  beat=celery-beat
```

5. Same on Both Stacks — Platform APIs & Shared Tools

External Platform APIs

API	Auth	Free Rate Limit	Key Tip
Shopify Admin API	OAuth + access token	2 req/sec (burst: 40)	Use webhooks to avoid polling. Store tokens AES-256 encrypted.
Meta Marketing API	OAuth + access token	200 calls/hr per app	Batch requests to group API calls. Cache insights — they change slowly.
Google Ads API	OAuth2 + developer token	15,000 ops/day	Apply for Standard Access post-launch. Use GAQL for efficient bulk reads.
Stripe API	Secret key	No documented limit	Always use webhook events. Add idempotency keys to all POST requests.
Exchange Rates (FX)	API key (exchangeratesapi.io)	1,500 req/month free	Cache daily — no need for real-time rates.

Shared Dev Tools (Both Stacks)

Tool	Purpose	Cost
GitHub + GitHub Actions	Version control, PR reviews, automated tests + deploys	Free (Student Pack = unlimited private repos + CI minutes)
Figma	UI design and component handoff	Free via GitHub Student Pack → Professional plan
Notion	Internal wiki, roadmap, meeting notes	Free with .edu email (Education plan)
Discord	Team communication + GitHub notifications bot	Free forever
Excalidraw	Quick system design diagrams	Free, open source, no login needed
Postman / Bruno	API testing and documentation	Postman free tier / Bruno is fully open source

6. Monthly Cost Projection

Stage	Users	Stack A Cost	Stack B Cost	What Kicks In
Dev	0	$0	$0	Everything on free tiers. No real traffic.
Beta	1–50	$0	$0–$5	Supabase/Neon free handles this. Railway Student credits cover workers.
Early Traction	50–200	$0–$30	$20–$50	Supabase Pro ($25/mo) or Neon Launch ($19/mo). Inngest Team ($25/mo).
Growth	200–500	$50–$100	$60–$120	All core services now paid but under $120/mo. You have MRR to cover it.
Scale	500–1,500	$100–$300	$150–$400	At this MRR ($50K+/mo), infra is < 1% of revenue. Not a problem.

💰 At 500 paying customers (~$120K ARR), your infra bill is ~$100–150/month — less than 0.1% of revenue. The only cost that scales linearly with revenue is Stripe's 2.9% transaction fee, which is just a cost of doing business. You will never be "too broke for infra" before you're profitable.

7. Quick Decision Guide — Which Stack?

If this describes your team...	Choose
Everyone knows JavaScript/TypeScript. Nobody knows Python.	Stack A
At least one person is comfortable with Python (data scripts, Flask, Django, pandas).	Stack B
You need to ship an MVP in less than 6 weeks.	Stack A
You want to build the ML attribution model (Tier 5) in Year 1.	Stack B
You want the simplest possible deployment — no DevOps whatsoever.	Stack A
You plan to hire data engineers or ML engineers later.	Stack B — your codebase will already be in Python
You're not sure — team is split.	Stack A to start, add Python microservice for data pipeline in Phase 2

⚠ The wrong choice is debating for two weeks instead of starting. Both stacks can build PFAM fully. Pick based on what your team already knows, not what sounds most impressive.

Revision History

Version	Date	Notes
1.0	February 2026	Initial tech stack doc. Two stack options, student credits guide, cost projection, decision guide.