Challey

Build Proposal — Chore Tracking & Rewards App

Prepared: February 11, 2026

Version: 1.0

Classification: Confidential

RECOMMENDATION

Proceed to Build

01 Executive Summary

Challey is a mobile-friendly chore tracking app that lets parents assign chores with rewards and lets teens report completed work for verification. It solves a simple problem: teens do chores but have no easy way to communicate that to their parents.

WHAT WE'RE BUILDING

A web application with two views — one for parents (assign chores, verify work, grant rewards) and one for teens (see chores, mark complete, track rewards). The core loop is: **Assign** → **Complete** → **Verify** → **Reward**, designed to take less than 30 seconds per interaction.

Key Numbers

BUILD INVESTMENT

$3,200

One-time cost to launch MVP

MONTHLY RUNNING COST

$0

At launch (free tiers)

FIRST PAID TIER TRIGGER

500+

Families before costs increase

RECOMMENDATION

Proceed to build. Challey's MVP is small, well-scoped, and technically straightforward. The teen-first design angle is a genuine market gap. The application can launch on entirely free infrastructure tiers and only begins incurring costs at meaningful scale.

02 What You're Getting

The Problem

Teens do chores but have no structured way to report completed work to parents. This leads to forgotten tasks, repeated reminders, disputes about what was done, and a lack of recognition that demotivates responsible behavior.

The Solution

A simple app with a four-step core loop:

1. **Dad assigns a chore** — name, description, reward amount, optional due date
2. **Maggie sees it and does it** — clear list of what's assigned and what each is worth
3. **Maggie marks it complete** — one tap, optional note
4. **Dad verifies and rewards** — checks the work, approves or requests changes

Who Uses It

User	What They Do	Device	Key Need
Parent (Dad)	Assigns chores, verifies work, grants rewards	Phone	Fast — under 30 seconds per action
Teen (Maggie)	Views chores, marks complete, tracks rewards	Phone	Recognition — see effort acknowledged

What's Included in Version 1

Feature	Description	Status
Chore assignment	Create chores with name, description, reward, due date	MVP
Chore board	See all active chores by status (pending, in progress, awaiting review)	MVP
Mark complete	One-tap submission with optional note	MVP
Verify & reward	Approve, reject with feedback, grant reward	MVP
Reward tracking	Earned this week, all time, pending approval, history	MVP

Activity feed	Timeline of assignments, completions, approvals	MVP
Notifications	Alerts for new chores, completions, approvals	MVP
Family login	Simple family code — no complex account setup	MVP

What's NOT in Version 1

These are parked for future versions based on user feedback:

- Recurring chores (auto-assign weekly tasks)
- Photo proof of completion
- Allowance tracking and banking integration
- Calendar sync
- Gamification (badges, levels, streaks)
- Smart home integration

03 How We'll Build It

We're recommending a modern web application that works great on phones without needing to download anything from the App Store. This keeps things simple and fast to launch.

> PLAIN-LANGUAGE SUMMARY
>
> Challey will be a **web app** that looks and feels like a phone app. Users open it in their browser, and it works on any device — iPhone, Android, tablet, or computer. We can add it to the App Store later if needed, using the same codebase.

Technology Choices

What	Our Choice	Why (In Plain English)
App framework	Next.js (React)	Used by Netflix, TikTok, and Notion. Huge developer community means easy to find help and hire talent. Fast to build, fast for users.
Database	Supabase (PostgreSQL)	Like a spreadsheet on steroids. Stores all chores, users, and rewards. Has built-in real-time sync so Dad sees updates the moment Maggie marks a chore done. Free tier is generous.
Login system	Supabase Auth	Comes free with the database. Supports family code login for MVP. Can upgrade to Google/Apple sign-in later.
Hosting	Vercel	Made by the same team as Next.js, so it's the smoothest pairing. Deploys automatically when code is updated. Free tier handles thousands of users.
Notifications	In-app first, push later	Start with notifications inside the app (like a bell icon with a badge). Add phone push notifications in a future version once the core workflow is proven.

Why This Approach?

- **Fast to launch** — this stack has the fastest setup-to-first-feature time of all options evaluated
- **Free to run at launch** — all services have generous free tiers that cover a single family easily
- **Scales affordably** — costs grow gradually with usage, no sudden jumps

- **Future-proof** — can evolve to a native mobile app, add more features, and scale to thousands of families without rewriting

04 Build Investment (One-Time Cost)

This is the one-time cost to build and launch Challey's MVP. After this, the app is live and usable.

Build Phases

Phase 0: Project Setup

Set up the codebase, database, hosting, and development environment. Configure login system.

Deliverable: Empty app that loads, with login working

Phase 1: Database & Data Layer

Create database tables for families, users, chores, and rewards. Set up real-time sync between devices.

Deliverable: Data foundation ready for all features

Phase 2: Parent Experience

Build Dad's view: dashboard, create chore form, verification screen, reward granting.

Deliverable: Parent can assign chores and verify completed work

Phase 3: Teen Experience

Build Maggie's view: chore list, mark complete flow, reward tracker, activity feed.

Deliverable: Full core loop working — assign, complete, verify, reward

Phase 4: Notifications & Polish

Add in-app notifications, empty states, error handling, mobile optimization, and overall polish.

Deliverable: Launch-ready MVP

Phase 5: Testing & Launch

Test with Maggie's family. Fix bugs. Deploy to production.

Deliverable: Live application

Cost Breakdown

Category	Item	Cost
Development	Full-stack development (Phases 0–5)	$2,800

Design	UI/UX design and branding (included in prototype phase)	$0*
Infrastructure Setup	Vercel project, Supabase project, domain name, SSL	$15
Third-Party Setup	No upfront fees — all services are usage-based or free tier	$0
Testing	Included in Phase 5	$0
Total Build Investment		**~$2,815**

*Design is produced during the Crucible discovery and prototyping phases at no additional cost.

WHAT YOU GET FOR THIS INVESTMENT

A fully working, live application that your family can use immediately. It's hosted on the internet with a real URL, works on any phone or computer, and includes the complete assign → complete → verify → reward workflow with notifications.

05 Running Costs (Monthly)

Once built, Challey needs hosting and services to stay online. Here's what that costs — and how costs change as more families use the app.

Service-by-Service Breakdown

Service	What It Does	Free Tier Limit	First Paid Tier
Vercel (hosting)	Serves the app to users	100GB bandwidth/mo	$20/mo after limit
Supabase (database)	Stores all data, handles login, real-time sync	500MB storage, 50K auth users	$25/mo after limit
Domain name	challey.app or similar	—	$12/year (~$1/mo)

Cost at Scale

Here's what running Challey costs as it grows from your family to a real product:

Launch

Families	1–10
Users	2–30
Chores/month	~200

$1/mo

$12/year (domain only)

Growth (6 months)

Families	50–200
Users	150–600
Chores/month	~5,000

$26/mo

$312/year

Scale (12 months)

Families	500–2,000
Users	1,500–6,000
Chores/month	~50,000

$75/mo

$900/year

WHAT TRIGGERS COST INCREASES

- **Database storage exceeds 500MB** → Supabase paid tier ($25/mo). This happens around 200+ active families.
- **Bandwidth exceeds 100GB/mo** → Vercel paid tier ($20/mo). This happens around 500+ active families.
- **Push notifications added** → Future cost of ~$0–$25/mo depending on provider and volume.

Revenue vs. Cost (If Commercialized)

If Challey charges families, here's how the economics look:

Scale	Families	Monthly Revenue (at $2.99/mo)	Monthly Cost	Monthly Profit
Launch	10	$30	$1	+$29
Growth	200	$598	$26	+$572
Scale	2,000	$5,980	$75	+$5,905

Note: Revenue projection assumes $2.99/mo subscription. Alternative models (one-time $9.99, freemium) would differ. These are illustrative, not forecasts.

06 Key Decisions

Decisions Made by Approving This Proposal

Decision	What It Means	Can We Change Later?
Web app first	Users access via browser, no App Store download needed	Yes — can add native app later using same codebase
Family code login	Simple shared code to join a family, not individual email/password	Yes — can upgrade to Google/Apple sign-in later
In-app notifications only	No phone push notifications in v1 — alerts appear inside the app	Yes — push notifications can be added in a future version
Money rewards only	Rewards are dollar amounts, not points or privileges	Yes — can add reward types later

> GOOD NEWS
>
> All decisions above are reversible. The technology choices are designed so that nothing locks you in — every decision can be changed or expanded later without rebuilding from scratch.

Decisions That Need Your Input During Build

- **Domain name** — what URL should the app live at? (e.g., challey.app, getchalley.com)
- **Color scheme & branding** — finalized during prototype phase
- **Default reward amounts** — any suggested defaults when creating chores?
- **Auto-approve setting** — should parents be able to set certain chores to auto-approve?

07 Risks

Risk	Likelihood	Impact	How We Handle It
Dad doesn't adopt — if the parent doesn't use the app, the	Medium	Critical	Make parent actions extremely fast (<30 sec). Strong notifications. Test

teen can't either			with real family first.
Feature creep — requests for calendars, banking, gamification before core is solid	High	High	Frozen MVP scope. "Yes, later" roadmap for future features. Launch simple, iterate from feedback.
Real-time sync issues — dad and teen see different data	Low	High	Supabase has built-in real-time subscriptions. Tested in thousands of production apps.
Competing with texts/sticky notes — "why use an app when I can just tell Dad?"	Medium	Medium	Reward history and recognition features provide value texts can't. Visible progress motivates teens.
Supabase pricing changes — free tier could become paid	Low	Low	Standard PostgreSQL database — can migrate to any provider if needed.

OVERALL RISK ASSESSMENT

Low technical risk, medium adoption risk. The technology is proven and straightforward. The main risk is human — will the family actually use the app consistently? That's why we recommend testing with Maggie's family first before any broader launch.

08 Build Schedule

Here's the order we'll build in, and what's usable after each phase:

Phase	What Gets Built	What's Usable After	Relative Effort
0	Project setup, hosting, login	App loads, users can sign in	Low
1	Database, data layer, real-time sync	Data foundation (not user-visible)	Low
2	Parent: dashboard, create chore, verify, reward	Parent can assign chores and manage them	Medium
3	Teen: chore list, mark complete, rewards, activity feed	**Full core loop working end-to-end**	Medium
4	Notifications, empty states, error handling, mobile polish	Polished, launch-ready experience	Medium
5	Family testing, bug fixes, deployment	**Live application**	Low

INCREMENTAL VALUE

After Phase 3, the app is functionally complete — Maggie's family could start using it. Phases 4–5 add polish, notifications, and reliability, but the core loop works from Phase 3 onward.

09 Recommendation

PROCEED TO BUILD

We recommend building Challey.

The MVP is tightly scoped, technically straightforward, and addresses a genuine gap in the market (teen-first design). The build investment is modest, the running costs start at zero, and every technology decision is reversible.

Conditions

1. Scope stays frozen as defined in this proposal. New feature requests go to the roadmap, not the build.
2. Maggie's family commits to testing the app during and after build — real usage feedback is critical.
3. Domain name is registered before build begins.

Next Steps (If Approved)

1. Client confirms "proceed" — build authorization granted
2. Build Planner creates detailed task breakdown and developer guide
3. Developer begins Phase 0 (project setup)
4. Automated milestone reviews at key checkpoints
5. Family testing at Phase 3 completion
6. Launch after Phase 5
