Challey

Build Plan — How We'll Build Your App

Prepared: February 11, 2026

Version: 1.0

Tech Stack: Next.js + Supabase + Vercel

Classification: Confidential

01 Building Materials & Services

Think of this like a construction project. Before we start building, here's everything we need — the tools, materials, and services — and what each one costs.

FOUNDATION

Next.js 15

The framework we build the app with. Like the framing of a house — everything attaches to this.

Free (open source)

STORAGE

Supabase (PostgreSQL)

Where all your data lives — chores, users, rewards. Also handles login and real-time updates between devices.

Free up to 500MB / $25/mo after

HOSTING

Vercel

Puts the app on the internet so anyone with the URL can use it. Handles traffic automatically.

Free up to 100GB bandwidth / $20/mo after

ADDRESS

Domain Name

Your app's address on the internet (e.g., challey.app or getchalley.com).

~$12/year

SECURITY

Supabase Auth

Handles login, family codes, and making sure parents see parent things and teens see teen things.

Included with Supabase

CODE QUALITY

Testing & Linting Tools

Automated checks that catch bugs before they reach you. Like a building inspector.

Free (open source)

TOTAL MATERIALS COST

$12/year at launch (domain name only). Everything else is free until you outgrow the free tiers — which doesn't happen until 200+ families are using the app.

02 Build Order

Here's the order we'll build in — like constructing a house: foundation first, then walls, then the rooms, then finishing touches.

Phase	What We're Building	Tasks	Client Demo?	What You Can See After
0	Foundation & Setup	4	No	Empty app loads in browser, dev tools configured
1	Database & Data	3	No	Data structure ready (not visible to users yet)
2	Login & Roles	3	No	Users can sign in and the app knows if they're a parent or teen
3	Parent Experience	5	**Yes — Milestone 1**	Parent can create chores, see dashboard, verify work
4	Teen Experience	4	**Yes — Milestone 2**	**Full loop working:** assign → complete → verify → reward
5	Notifications & Activity	3	No	Users get alerted about new chores, completions, approvals
6	Polish & Edge Cases	4	**Yes — Final Demo**	**Launch-ready app**

MILESTONES = YOUR CHECKPOINTS

At Milestones 1, 2, and Final, we pause and show you what's been built. You click through it, give feedback, and we adjust before moving forward. You're never surprised at the end.

03 What We Need From You

Some things can only come from you. Here's everything we'll need, when we need it, and what happens if it's delayed.

1 **Register domain name (e.g., challey.app)**
You — purchase from Namecheap, Google Domains, or similar
Blocks: Phase 0
Needed before build starts

2 **Create Supabase account & project**
You — sign up at supabase.com (free), create a project, share the API keys
Blocks: Phase 1
Needed before database setup

3 **Create Vercel account & connect GitHub repo**
You — sign up at vercel.com (free), we walk you through connecting it
Blocks: Phase 0 (deployment)
Needed before first deployment

4 **Confirm branding: colors, logo, app name**
You — finalized during prototype phase
Blocks: Phase 3 (UI)
Needed before building screens

5 **Confirm default chore categories and reward amounts**
You — provide a list of common chores and typical reward amounts for seed data
Blocks: Phase 1 (seed data)
Needed before demo data is created

6 **Test with your family during Milestone 2**
You + Maggie — use the app for real chores for at least 3 days
Blocks: Phase 6 (final polish)
Needed before final phase

> CRITICAL PATH WARNING
>
> Items 1 and 2 (domain + Supabase account) block the entire build. **We cannot start without them.** Items 3-5 can be provided during the first few days of building. Item 6 is needed later but is critical for a successful launch.

Priority Legend

Priority	Meaning	Impact If Delayed
P1 — Critical	Build cannot start or continue without this	Entire project pauses

P2 — Important	Specific phase is blocked, but other work can continue	Phase delayed, may push final delivery
P3 — Nice to Have Early	Improves quality but doesn't block progress	Can use defaults, adjust later

04 Detailed Task Breakdown (Sample)

Below is a sample of what each task looks like. The full build plan has 26 tasks across 7 phases. Each task tells the developer exactly what to build, how to test it, and what "done" looks like.

Phase 0: Foundation & Setup

0.1 Scaffold Next.js Project

What we're building

Create the base project with Next.js 15, TypeScript, Tailwind CSS, and ESLint. This is the empty shell everything else goes into.

What "done" looks like

App runs locally in browser showing a welcome page. All code quality tools are configured and passing.

Files created

`package.json`, `tsconfig.json`, `tailwind.config.ts`, `.eslintrc.json`, `app/layout.tsx`, `app/page.tsx`

0.2 Configure Environment & Deployment

NEEDS: VERCEL ACCOUNT

What we're building

Connect the project to Vercel for automatic deployments. Set up environment variables for Supabase connection. Every code change automatically deploys a preview.

What "done" looks like

Push code to GitHub → Vercel automatically builds and deploys → app is live at a temporary URL.

What we need from you

Vercel account created and connected to the GitHub repository (we'll walk you through this — takes 5 minutes).

Phase 3: Parent Experience (Sample Tasks)

3.1 Parent Dashboard

What we're building

The first screen Dad sees after logging in. Shows a summary of active chores, how many need verification, and quick stats (chores completed this week, total rewards paid).

What "done" looks like

- Dashboard loads showing active chore count, pending verification count, and weekly stats
- "New Chore" button is prominent and tappable
- Chore cards show title, assignee, reward amount, and status
- Empty state shows helpful message when no chores exist yet
- Works on phone screens (responsive layout)

Screens from prototype

Parent Dashboard (Screen 1 of 7) — matches approved prototype layout

Tests

- Dashboard renders with correct chore counts
- Empty state shows when no chores exist
- "New Chore" button navigates to create form
- Dashboard is responsive on mobile viewport

3.2 Create Chore Form

What we're building

The form where Dad creates a new chore. Fields: chore name, description, assign to (select kid), reward amount, due date (optional). Validates input and saves to database.

What "done" looks like

- Form shows all fields with clear labels
- Name is required — shows "Please enter a chore name" if empty
- Reward must be a positive number — shows "Reward must be at least $0.01" if invalid
- Assign to shows list of kids in the family
- Submit creates the chore and returns to dashboard
- Chore appears on the assigned kid's list in real-time

Business rules

- Only parents can create chores
- Chore name max 100 characters
- Description max 500 characters
- Due date must be in the future

Tests

- Form renders all fields
- Validation shows correct error messages for each field
- Successful submission creates chore in database
- Non-parent users cannot access this form (redirect to teen view)

05 Milestone Schedule

These are your checkpoints — the moments you see working software and give feedback before we continue.

MILESTONE 1 **Parent Experience Demo** After Phase 3

What you'll see

A working app where Dad can log in, see the dashboard, create a chore with a reward, and verify a completed chore. Real data, real interactions.

Demo script

1. Log in as Dad → see empty dashboard with "Create your first chore" message
2. Tap "New Chore" → fill in "Clean garage", $15 reward, assign to Maggie
3. Return to dashboard → see the chore card appear
4. (We simulate Maggie completing it)
5. See "1 chore awaiting verification" → tap to review
6. Approve → see reward granted, chore moves to completed

Your feedback

Does this feel right? Is anything missing? Is it fast enough? Would Dad actually use this?

MILESTONE 2 Full Loop Demo

After Phase 4

What you'll see

The complete core loop working end-to-end. Dad assigns, Maggie sees it, Maggie marks done, Dad verifies, reward is tracked. Both roles fully functional.

Demo script

1. Log in as Dad → create chore "Wash dishes", $5 reward
2. Switch to Maggie's account → see new chore on "My Chores" list
3. Tap chore → see details, reward, due date
4. Tap "Mark Complete" → add note "Also wiped counters!"
5. Switch to Dad → see pending verification
6. Approve → Maggie's reward tracker updates
7. Check Maggie's rewards tab → see $5 earned, history shows the chore

After this milestone

Real family testing begins. You and Maggie use this for real chores for 3+ days.

FINAL DEMO Launch-Ready Review

After Phase 6

What you'll see

The finished MVP: notifications working, all edge cases handled, mobile-optimized, smooth transitions, helpful empty states and error messages.

What's changed since Milestone 2

- In-app notifications for new chores, completions, and approvals
- Activity feed showing family chore history
- Polish: loading states, error handling, offline behavior
- Bug fixes from family testing

Your decision

Launch or iterate? If you're happy, we go live. If not, we address remaining feedback.

06 Approval

WHAT YOU'RE APPROVING

- The building materials and services listed in Section 01
- The build order and phasing in Section 02
- The action items assigned to you in Section 03
- The milestone checkpoints in Section 05

Before we start building, please:

1. **Review** the build order — does the phasing make sense?
2. **Check** your action items — can you provide items 1-3 within the next few days?
3. **Confirm** the milestones — are these the right checkpoints?
4. **Approve** — "Go ahead and build" or "I have questions first"

WHAT HAPPENS NEXT

Once you approve, the Developer starts Phase 0 (foundation setup). You'll be contacted when we need your action items and again at each milestone for a demo. Between milestones, the build runs automatically — no input needed from you unless we hit a blocker.

This document is confidential and prepared for the intended recipient only.