

Frontend Guide

Frontend at a Glance

- Next.js App Router + TypeScript + Tailwind UI with theme palettes via next-themes.
- Global providers prevent duplicate Firestore reads and keep UX responsive.
- PWA service worker registers silently and stores push subscriptions in Firestore.

Table of Contents

Right-click and update field to generate TOC.

1. Tech Stack & Major Libraries

- Framework: Next.js (App Router; server + client components)
- Language: TypeScript
- Styling/UI: Tailwind CSS; next-themes (dark/light + palettes)
- Icons: Lucide React; Animations: Framer Motion; Toasts: Sonner
- PWA: Service Worker (public/sw.js) + web-push

2. Structure & Layout

Layout & theming tokens

- app/layout.tsx controls global metadata and SW registration; CSS variables live in globals.css to avoid hardcoded colors.
- ThemeProvider toggles light/dark and theme palettes by applying classes onto the body.

3. Providers & State Management

Provider	Where used	Responsibilities
AuthProvider.tsx	components/providers	Listen to onAuthStateChanged; gate main routes
ThemeProvider.tsx	components/providers	Initialize next-themes and palette switching
SidebarProvider.tsx	components/providers	Global sidebar expand/collapse state (desktop/mobile)
SoundProvider.tsx	components/providers	Audio context for subtle UI sounds
ReminderProvider.tsx	components/providers	Single Firestore onSnapshot for reminders; hydrate global state

4. Key UI Flows

Reminder creation & editing

- components/reminders/ReminderForm.tsx drives addReminder/updateReminder (lib/reminders.ts).
- Wrapper UX via ReminderSheet (mobile) or modal patterns.

Calendar view data

- ReminderProvider provides the canonical reminder list; calendar transforms are memoized for fast render.

Routines UI

- components/routines/* manages ordered steps (HH:MM) and toggles enable/disable via api/routines/[id]/run.

Notes expand/collapse

- Framer Motion height animation avoids jarring reflow while expanding notes.

PWA + service worker registration

- ServiceWorkerRegister registers /sw.js; Settings flow creates VAPID subscription and sends to api/push/subscribe.

5. Component Map

Component	Where used	Responsibilities
ReminderForm	Reminders UI	Create/edit reminder; parse dates/timezones; map notifications
ReminderSheet	Mobile UX	Slide-up wrapper for ReminderForm
ReminderList	Dashboard	Display reminders; quick actions
Calendar grid	/calendar	Month/week/day grouping and rendering
RoutineForm	/routines	Edit steps and schedule; local HH:MM abstraction
RoutineCard	/routines	Enable/disable toggle; triggers catch-up run
ServiceWorkerRegister	Layout	Register sw.js and validate PushManager support

6. Performance Checklist

✓ Best Practice: Keep Firestore reads centralized

- Use a single global `onSnapshot` (`ReminderProvider`) to avoid duplicate reads across pages.

✓ **Best Practice: Memoize heavy calendar transforms**

- Group reminders by month/week/day with `useMemo` to prevent rerender thrash.