

Reminders App

A modern, responsive PWA for managing reminders, routines, and habit workflows with reliable multi-channel notifications and a precomputed queue architecture.

At a Glance

- User features: reminders, routines, calendar, repeat rules, snooze/completion.
- System features: denormalized notification_queue, delta sync, cron runner dispatch.
- Platforms: Desktop web + iPhone PWA (Home Screen) with iOS 16.4+ push support.

Table of Contents

Right-click and update field to generate TOC.

1. App Purpose & Target Users

The Reminders App is a Progressive Web App (PWA) designed to help users manage tasks, routines, and habits. It combines a calendar interface with dynamic scheduling logic and supports push, SMS, and email notifications.

2. Core Features

2.1 User Features

- Reminders with due dates/times/timezones; notes attached and gracefully truncated in push payloads.
- Completion and Snooze: marking done halts future queued notifications; snooze reschedules and updates the queue.
- Calendar views (grid/list) using robust date boundary handling.
- Repeat rules: presets (daily/weekly/monthly) plus custom repeatRule cloning behavior.
- Routines: sequential steps generated into standalone reminders for a rolling 24-hour window.

2.2 System Features

- Multi-channel notifications: Web Push (VAPID + Service Worker), SMS (Twilio), Email (SMTP/Nodemailer).
- Denormalized notification_queue per user: avoids scanning reminders every minute.
- Delta sync: edits to a reminder/routine rebuild only its affected queue items.
- Theme system: next-themes with multiple palettes; UI micro-animations via framer-motion.

- Multi-device push subscriptions stored in `push_subscriptions`.

3. Platform Support

| Capability | Desktop Web | iPhone PWA (Home Screen) |
|--------------------|---|---|
| Install | No install required | Must use Add to Home Screen |
| Push notifications | Standard Web Push API | iOS 16.4+ Web Push via Service Worker |
| Offline behavior | Firestore offline support (client) | Firestore offline support (client) |
| Multi-device push | Supported via <code>push_subscriptions</code> | Supported via <code>push_subscriptions</code> |
| Best UX path | Settings → Enable Push | Install PWA → Settings → Enable Push |

4. Operational Behavior

How Notifications Work (High Level)

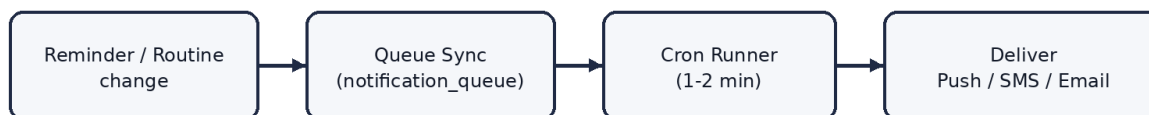


Figure: boxed flow diagram

Cron schedule calls

- Minute-runner pinged externally every 1–2 minutes to `app/api/cron/run/route.ts` (Bearer secret protected).
- Queries `notification_queue` for unsent items bounded strictly in a `[now-2m, now]` window to account for drift.

Daily 12am generation & rebuild

- On the first minute-runner ping of a new day, the system triggers routine generation and rebuilds the queue horizon (24h).

5. Known Limitations & Assumptions



Warning: Missing client clock sync

- Due times are treated as absolute instances based on provided timezone or user fallback timezone.



Warning: SMS rate limits

- If many SMS reminders are scheduled for the exact same minute, Twilio may throttle unless configured for high throughput.



Warning: Offline completion gap

- If a reminder is marked done offline, the cloud queue may still fire before the client reconnects and syncs.