Masjid Notify - Project Roadmap

Project Overview

A WhatsApp-based notification system for mosques. Users scan a QR code, subscribe with their phone number, and receive prayer time reminders, daily hadith, and mosque announcements.

Target Market: South African mosques, starting with one pilot mosque

Core Value: Low-friction subscription (QR scan), reliable delivery (WhatsApp), and simple admin management for mosque caretakers.

Phase 1: Foundation (Week 1-2)

Goal: Get the basic system working for one mosque with manual announcements.

Deliverables:

- Landing page with QR code scan destination
- Phone number collection form with basic preferences (Fajr only, all prayers, announcements only)
- WhatsApp Business API integration via WATI or Respond.io
- Simple admin dashboard for sending manual announcements
- Subscriber list management (view subscribers, remove if needed)

Tech Stack:

- Frontend: Next.js or simple React app
- Backend: Node.js or Python
- Database: Supabase or Firebase
- WhatsApp: WATI ($40/month) or Respond.io

Success Criteria: 10 people subscribed, receiving manual test announcements

---

Phase 2: Automated Prayer Times (Week 3-4)

Goal: Automatic daily prayer time notifications without manual input.

Deliverables:

- Aladhan API integration for prayer time calculations
- Madhab selection (Hanafi/Shafi'i) per mosque
- Time offset settings (e.g., +10 minutes for Isha)
- Scheduled notifications X minutes before each prayer
- Subscriber preference respect (only send prayers they opted into)

Admin Features:

- Set mosque location (latitude/longitude or city selection)
- Choose calculation method and madhab
- Set offset per prayer time
- Toggle which prayers send notifications

Success Criteria: Automated Fajr and Maghrib reminders running for 1 week without manual intervention

---

Phase 3: Content & Engagement (Week 5-6)

Goal: Add value beyond just prayer times.

Deliverables:

- Curated hadith database (Sahih Bukhari, Sahih Muslim - verified sources only)
- Morning/evening adhkar reminders
- Daily hadith delivery after Fajr
- Countdown feature ("Fajr in 1 hour", "Maghrib in 30 minutes")

Content Rules:

- Only authenticated hadith from verified collections

- No AI-generated religious content or rulings
- Sources cited with each hadith

Success Criteria: Daily hadith sending automatically, positive feedback from subscribers

---

Phase 4: Ramadan Mode (Week 7-8)

Goal: Special features for Ramadan.

Deliverables:

- Suhoor reminder (configurable time before Fajr)
- Iftar countdown and notification
- Taraweeh time announcements
- Ramadan-specific content (virtues of fasting, Laylatul Qadr reminders)
- Special program announcements for Ramadan events

Admin Features:

- Toggle Ramadan mode on/off
- Set Taraweeh start time
- Schedule Ramadan-specific programs

Success Criteria: Full Ramadan support ready before Ramadan 2025 (starts ~Feb 28)

Phase 5: Multi-Mosque & Scale (Week 9-12)

Goal: Allow any mosque to join the platform.

Deliverables:

- Mosque onboarding flow (register mosque, set location, configure settings)
- Unique QR code per mosque
- Isolated subscriber lists per mosque
- Role-based admin access (Head Admin, Muazin, Announcer)

- Mosque-specific branding on landing pages

Business Model Options:

- Free tier: Basic announcements, limited subscribers
- Paid tier: Unlimited subscribers, automated prayers, hadith, Ramadan features
- Or: Mosque pays monthly subscription (~R500-1000/month?)

Success Criteria: 3 mosques onboarded and actively using the system

---

Technical Architecture

```
[User Phone]
   ↓ scans QR
[Landing Page] → collects number, preferences
   ↓
[Database] → stores subscriber, mosque link, preferences
   ↓
[Scheduler] → checks prayer times, triggers notifications
   ↓
[WhatsApp API (WATI)] → delivers message
   ↓
[User receives on WhatsApp]

[Admin Dashboard]
   ↓
[Send Announcement] → [WhatsApp API] → [All subscribers of that mosque]
```

Cost Breakdown (Pilot Phase)

Item	Monthly Cost
WATI WhatsApp API	~$40 (R750)
Hosting (Vercel/Railway)	~$0-20
Database (Supabase free tier)	$0

Item	Monthly Cost
Domain	~R150/year
Total	**~R800-1000/month**

Per-message costs depend on volume. WhatsApp Business API charges per 24-hour conversation window, not per message. So multiple messages in a day to one person = one conversation charge (~$0.05-0.08).

Unsubscribe Flow

User sends "STOP" → System removes from list → Confirms removal

User sends "PAUSE" → Pauses for X days (optional feature)

User sends "SETTINGS" → Receives link to update preferences

Content Sources (Verified Only)

- **Hadith:** Sunnah.com API (Sahih Bukhari, Sahih Muslim, authenticated collections)
- **Prayer Times:** Aladhan API (supports multiple calculation methods)
- **Adhkar:** Fortress of the Muslim (Hisnul Muslim) - authenticated duas

No content is AI-generated for religious rulings. All hadith include source reference (e.g., "Sahih Bukhari 1234").

QR Code Message Ideas

- "Scan for daily Salah reminders"
- "Never miss a Jamaat - Subscribe here"
- "Stay connected to [Mosque Name]"
- "Get prayer times & programs on WhatsApp"

Next Steps

1. Choose WhatsApp provider (WATI vs Respond.io)

2. Build landing page and subscription form
3. Create admin dashboard MVP
    a. Set up WhatsApp Business account for pilot mosque
4. Test with 5-10 people
5. Iterate based on feedback
6. Roll out to full mosque congregation

User Journey (Step by Step)

The Setup

User is at the mosque. They see a poster or sticker near the entrance that says: **"Stay connected to Masjid Al-Noor - Get prayer times & programs on WhatsApp"** with a QR code.

Step 1: Scan the QR Code

User scans with phone camera → Opens a clean webpage (no app download needed)

Step 2: Welcome Page

User sees:

- Mosque name and logo
- Short message: "Get daily prayer reminders, program updates, and Islamic reminders directly on WhatsApp"
- WhatsApp number input field
- Preference checkboxes: Fajr reminder, All 5 daily prayers, Jumu'ah reminder, Program announcements, Daily hadith, Ramadan reminders
- Subscribe button

Step 3: Submit

User enters number, selects preferences, hits Subscribe. Page shows: "Check your WhatsApp for a confirmation message"

Step 4: WhatsApp Confirmation

User receives: "Assalamu Alaikum! You're now subscribed to Masjid Al-Noor updates. Reply STOP to unsubscribe. Reply SETTINGS to change preferences."

Step 5: Ongoing Messages

Fajr Reminder: "⏰ Fajr begins at 5:02 AM. Sunrise at 6:15 AM. Don't miss your Salah."

Daily Hadith: "📖 The Prophet ﷺ said: 'The two rak'ahs of Fajr are better than the world and everything in it.' — Sahih Muslim 725"

Program Announcement: "📣 Tonight after Maghrib: Tafseer class with Sheikh Ahmad. All welcome."

Jumu'ah Reminder: "🕌 Khutbah begins at 1:00 PM. First Adhaan at 12:45 PM."

Ramadan Suhoor: "🌙 Suhoor ends at 4:45 AM. Eat something, even if just a date and water."

Ramadan Iftar: "🌇 Iftar in 30 minutes. Maghrib at 6:32 PM."

User Commands

- STOP → Unsubscribed
- PAUSE 7 → Paused for 7 days
- SETTINGS → Link to update preferences
- HELP → Support information

Database Schema

Subscribers Table

Fields: id (UUID), phone_number (String with country code), mosque_id (UUID), subscribed_at (Timestamp), status (active/paused/unsubscribed), pause_until (Timestamp), pref_fajr (Boolean), pref_all_prayers (Boolean), pref_jumuah (Boolean), pref_programs (Boolean), pref_hadith (Boolean), pref_ramadan (Boolean), last_message_at (Timestamp)

Mosques Table

Fields: id (UUID), name (String), city (String), country (String), latitude (Float), longitude (Float), madhab (hanafi/shafii), calculation_method (String), fajr_offset (Integer), dhuhr_offset (Integer), asr_offset (Integer), maghrib_offset (Integer), isha_offset (Integer), jumuah_khutbah_time (Time), jumuah_adhaan_time (Time), timezone (String), whatsapp_number (String)

Admins Table

Fields: id (UUID), mosque_id (UUID), name (String), email (String), password_hash (String), role (owner/admin/announcer)

Messages Table (Log)

Fields: id (UUID), mosque_id (UUID), type (prayer/hadith/announcement/ramadan), content (Text), sent_to_count (Integer), sent_at (Timestamp), sent_by (UUID)

Hadith Table

Fields: id (UUID), text_arabic (Text), text_english (Text), source (String), reference (String), category (String), verified (Boolean)

Technical Requirements

Frontend

- Framework: Next.js 14 (React)
- Styling: Tailwind CSS
- Hosting: Vercel (free tier)

Backend

- Framework: Node.js with Next.js API routes
- Database: Supabase (PostgreSQL)
- Authentication: Supabase Auth
- Scheduler: Vercel Cron Jobs

WhatsApp Integration

- Provider: WATI (wati.io) - ~$40/month
- Needs: WhatsApp Business Account, Facebook Business verification, Message templates approved

External APIs

- Prayer Times: Aladhan API (FREE)
- Hadith: Sunnah.com API (FREE)

Environment Variables

```
# Supabase
SUPABASE_URL=https://jlqtuynaxuooymbwrwth.supabase.co
SUPABASE_ANON_KEY=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6ImpscXR1eW5heHVvb3ltYndyd3RoIiwicm9sZSI6ImFub24iLCJpYXQiOjE3Njk3MzA3NDQsImV4cCI6MjA4NTMwNjc0NH0.LHyGdmdSvkkaD-IvBouhWBM7496dI1lIggGsaiq5ZLg
SUPABASE_SERVICE_ROLE_KEY=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6ImpscXR1eW5heHVvb3ltYndyd3RoIiwicm9sZSI6InNlcnZpY2Vfcm9sZSIsImlhdCI6MTc2OTczMDc0NCwiZXhwIjoyMDg1MzA2NzQ0fQ.dT2BlLDTn9XFrt2bzpzrQ0yALE1Fx7EIX_Mhw1ErWHM

# WhatsApp Cloud API (Meta Direct - NO platform fees!)
WHATSAPP_ACCESS_TOKEN=EAFu94PkhATkBQlDnuXqvbqqgZAlZCSt5NqhReYrpBt5Vv4ZBVmYNP8KzJXWKOTmPQiVHRDbSAjnZC8ZBkrlBZCPqXrvLVztpQlh7ZCG32ej7vn8VKHnrbE8SahZCBUUrnVqZBvtZCPptoLkQZCcZB3k2qc16NGpbw2RWrZBmrUEpTuOiCsQaez2SbrRQk2CnBTSMS4A7JMQZDZD
WHATSAPP_PHONE_NUMBER_ID=895363247004714
WHATSAPP_BUSINESS_ACCOUNT_ID=1443752210724410
META_APP_ID=2582299683068334 49
WHATSAPP_PHONE_NUMBER=+27694135669

# Auth
NEXTAUTH_SECRET=<generate random string>
```

```
# External APIs (free)
ALADHAN_API_URL=https://api.aladhan.com/v1
```

Folder Structure

```
masjid-notify/
├── app/
│   ├── page.tsx (Home/marketing)
│   ├── [mosque]/page.tsx (Subscription landing)
│   ├── admin/
│   │   ├── page.tsx (Dashboard)
│   │   ├── subscribers/
│   │   ├── announcements/
│   │   ├── settings/
│   │   └── login/
│   └── api/
│       ├── subscribe/
│       ├── webhook/
│       ├── send/
│       └── cron/
├── components/
├── lib/
│   ├── db.ts
│   ├── wati.ts
│   ├── prayer-times.ts
│   └── hadith.ts
├── prisma/schema.prisma
├── public/qr-codes/
├── .env.local
└── package.json
```

API Endpoints

Public

- POST /api/subscribe - New subscription
- POST /api/webhook/wati - WhatsApp webhook for replies
- GET /api/mosque/[id]/times - Get prayer times

Admin (Authenticated)

- GET /api/admin/subscribers - List subscribers
- DELETE /api/admin/subscribers/[id] - Remove subscriber
- POST /api/admin/announcements - Send announcement
- PUT /api/admin/settings - Update mosque settings

Cron Jobs

- GET /api/cron/prayer-reminders - Every 5 minutes
- GET /api/cron/daily-hadith - Once daily after Fajr
- GET /api/cron/jumuah-reminder - Friday mornings

Admin Walkthrough (Step by Step)

First Time Setup (Done once with you)

You meet the admin (e.g., Azinde) at the mosque. Go to: `/admin/register`

Fill in together:

- Mosque name
- City
- Admin email
- Create password

Admin gets confirmation email, clicks link, account active.

Then set up mosque settings:

- Location (type address, system converts to coordinates)
- Madhab: Hanafi or Shafi'i
- Time adjustments per prayer (e.g., Isha +10 minutes)

- Jumu'ah times: Khutbah start, first adhaan

Print QR codes, stick around mosque. Live.

Daily Admin Experience

Step 1: Open Browser

Admin goes to `/admin` on phone browser. No app needed.

Step 2: Login

Email and password. Simple.

Step 3: Dashboard

Clean screen showing:

- Subscribers count (e.g., 47 people)
- Messages sent this week
- Last announcement sent

Big clear buttons:

- 📣 Send Announcement
- 👥 View Subscribers
- ⚙️ Settings
- 🕌 Prayer Times

Sending an Announcement

Admin taps **Send Announcement**

Simple form:

- Message: (text box)
- Send to: Everyone / Only program subscribers / Only Ramadan subscribers
- When: Send now / Schedule for later

Admin types message → taps Preview → sees WhatsApp preview → taps Send → confirms → done.

Success screen: "✓ Sent to 47 subscribers"

Viewing Subscribers

Admin taps **View Subscribers**

Sees list with: Phone (masked), Subscribed date, Preferences, Status

Can:

- Search by phone
- Filter by preference
- See who unsubscribed
- Remove someone manually if needed

Adjusting Prayer Time Settings

Admin taps **Settings → Prayer Times**

Sees:

- Calculation Method (dropdown)
- Madhab toggle (Hanafi/Shafi'i)
- Adjustments per prayer: Fajr, Dhuhr, Asr, Maghrib, Isha (in minutes)

Change any value → Save → future notifications use new times.

Updating Jumu'ah Times

Admin taps **Settings → Jumu'ah**

Sees:

- First Adhaan time
- Khutbah start time
- Reminder timing (1 hour before, 2 hours before, morning)

Change → Save → next Friday uses new times.

Ramadan Mode

In Settings: 🌙 **Ramadan Mode: OFF** (toggle)

Switch ON → new options appear:

- Suhoor reminder: X minutes before Fajr
- Iftar reminder: X minutes before Maghrib
- Taraweeh time: (set manually)
- Taraweeh reminder: X minutes/hours before

Automatic Ramadan messages until toggled off after Eid.

Adding Another Admin

Admin goes to **Settings → Team**

Sees themselves as Owner.

Taps **Add Admin**:

- Enter new admin email
- Select role: Admin (full access) or Announcer (send only)

New admin gets email invite, creates password, has access.

Key Points for Explaining to Mosque Admins

1. "Login on phone browser - no app download"
2. "Send announcement = tap Send, type, send. Done."
3. "Prayer times are automatic - only touch settings if jamaat times change"
4. "Subscribers manage themselves - you just watch the numbers grow"
5. "Ramadan comes? Flip the switch on."

Design & Animation Guidelines

Brand

- Footer on all pages: "Powered by {alqode}" with subtle link
- Clean, minimal, professional

- If 5 people see it, it should still look premium

Animation Standards

- Toggle switches: smooth sliding animation with subtle bounce
- Buttons: gentle scale on hover/tap (1.02x), smooth color transition
- Page transitions: fade or subtle slide
- Headings: subtle fade-in on page load
- Loading states: clean spinner or skeleton, never jarring

Visual Elements

- Consider moon phase icon for Ramadan features
- Mosque silhouette or crescent for branding moments
- Keep Islamic aesthetic subtle and tasteful, not over-designed

Mobile First

- Admin dashboard must work perfectly on phone
- Large tap targets for buttons
- Readable without zooming

Color Palette (Suggested)

- Primary: Deep teal or green (Islamic association)
- Secondary: Warm gold/amber (moon, warmth)
- Background: Clean white or very light gray
- Text: Near-black for readability

Claude Code Instructions

When building this project, follow these rules:

Get Shit Done Mode

- Focus on completing one task fully before moving to next
- No over-planning mid-build
- If stuck for more than 10 minutes, simplify and ship
- Working code > perfect code

Loop Until Done

- If a feature doesn't work, keep iterating until it does
- Don't abandon half-finished features
- Test each feature before marking complete

Quality Bar

- Every UI element should feel polished
- Animations are not optional - they're part of the standard
- Mobile experience is primary, desktop is bonus
- Alqode brand reputation is on the line

Priority Order

1. Subscription flow working (QR → landing → WhatsApp confirmation)
2. Admin can send manual announcement
3. Prayer times auto-calculate and send
4. Polish and animations
5. Additional features (Ramadan mode, multi-admin, etc.)

Development Tools & Skills

A curated collection of tools, skills, and MCPs to supercharge the Masjid Notify build.

Core Development Tools

1. Ralph - Autonomous Agent Loop

Repo: https://github.com/snarktank/ralph

What it does: Runs Claude Code in an autonomous loop until all PRD items are complete. Each iteration gets fresh context but maintains memory via git history + progress.txt + prd.json.

Install:

```
git clone https://github.com/snarktank/ralph
cp -r ralph/skills/* ~/.claude/skills/
```

Use for: Letting Claude Code work autonomously on defined tasks while you step away.

---

2. GSD (Get Shit Done) - Meta-Prompting System

Repo: https://github.com/glittercowboy/get-shit-done

What it does: Spec-driven development system that solves context rot (quality degradation as context fills). Uses multi-agent orchestration with researchers, planners, executors, and verifiers working in parallel.

Install:

```
npx get-shit-done-cc
# Select: Claude Code + Global
```

Workflow:

1. `/gsd:new-project` → Initialize
2. `/gsd:discuss-phase` → Clarify requirements
3. `/gsd:plan-phase` → Create execution plan
4. `/gsd:execute-phase` → Build with atomic git commits
5. `/gsd:verify-work` → Validate output

Use for: Structured development with clear phases and verification.

---

3. Playwright - End-to-End Testing

Repo: https://github.com/microsoft/playwright

What it does: Microsoft's testing framework for Chromium, Firefox, and WebKit. Auto-waits for elements (no flaky tests), captures screenshots/videos for debugging.

Install:

```
npm init playwright@latest
```

Use for: Testing subscription flow, admin dashboard, mobile responsiveness, WhatsApp webhook handling.

UI & Design Tools

4. UI UX Pro Max - AI Design Skill

Repo: https://github.com/nextlevelbuilder/ui-ux-pro-max-skill

What it does: Professional UI/UX design skill with 67 UI styles, 96 color palettes, 56 font pairings, and 100 industry-specific reasoning rules. Generates design systems based on project type.

Install:

```
npm install -g uipro-cli
cd masjid-notify
uipro init --ai claude
```

For Masjid Notify: Soft UI Evolution style, wellness color palettes, elegant typography, smooth animations.

Use for: Ensuring premium, consistent design across all pages.

---

5. ReactBits + shadcn MCP - Premium Components

Registry Config (add to components.json):

```
{
  "registries": {
    "@react-bits": "https://reactbits.dev/r/{name}.json"
  }
}
```

shadcn MCP Install:

```
npx shadcn@latest mcp init --client claude
```

What it does:

- **ReactBits:** 135+ animated React components (glow cards, typing animations, interactive backgrounds)
- **shadcn MCP:** Connects Claude Code directly to shadcn registry for accurate TypeScript props and real component code

Use for: Premium animated UI without the manual work. Claude Code gets real-time component data instead of outdated training info.

Note: Some ReactBits components (buttons, forms, loaders) have incomplete implementations. Use for animations/effects, stick to standard shadcn for core form elements.

Video & Content Creation

6. Remotion Skills - AI Video Creation

Repo: https://github.com/remotion-dev/skills

What it does: Teaches Claude Code how to create videos using Remotion (React-based video framework). Describe videos in natural language, Claude writes the code, render to MP4.

Install:

```
# Create Remotion project
npx create-video@latest my-video && cd my-video

# Install skill
npx skills add remotion-dev/skills

# Start Claude Code
claude

# Render
npx remotion render <composition-name> out/video.mp4
```

Use for: Promo videos for Masjid Notify, tutorial videos showing subscription flow, Ramadan announcement videos, Alqode portfolio content.

Reference & Official Skills

7. anthropics/skills - Official Anthropic Skills

Repo: https://github.com/anthropics/skills (⭐ 40.2k)

What it contains:

- Official document skills (docx, pdf, pptx, xlsx) - powers Claude's file creation
- Example skills for creative, technical, enterprise tasks
- Agent Skills specification
- Skill template for creating custom skills

Install in Claude Code:

```
/plugin marketplace add anthropics/skills
/plugin install document-skills@anthropic-agent-skills
/plugin install example-skills@anthropic-agent-skills
```

Use for: Reference for skill structure, document generation capabilities.

---

8. vercel-labs/agent-skills - One-Command Deploy

Repo: https://github.com/vercel-labs/agent-skills

What it does: Deploy to Vercel with natural language. Just say "deploy my app" and it handles everything - no authentication needed.

Install:

```
cp -r skills/vercel-deploy ~/.claude/skills/
```

Features:

- Auto-detects 40+ frameworks from package.json
- Returns preview URL and claim URL

Use for: Quick deployments during development. Alternative to `vercel deploy` command.

- Excludes node_modules and .git automatically

Quick Reference Table

Tool	Type	Priority	Use Case
GSD	Meta-prompting	High	Structured dev workflow
Ralph	Autonomous loop	Medium	Hands-off task completion
UI UX Pro Max	Design skill	High	Premium UI consistency
Playwright	Testing	High	E2E testing before launch
ReactBits + shadcn MCP	Components	High	Animated UI components
Remotion Skills	Video	Low	Marketing content later
anthropics/skills	Reference	Medium	Learn skill patterns
vercel-labs/agent-skills	Deploy	Low	Quick deploys

Installation Checklist

- [ ] Install GSD: `npx get-shit-done-cc`
- [ ] Clone Ralph: `git clone` https://github.com/snarktank/ralph
- [ ] Install UI UX Pro Max: `npm install -g uipro-cli`
- [ ] Init Playwright: `npm init playwright@latest`
- [ ] Add shadcn MCP: `npx shadcn@latest mcp init --client claude`
- [ ] Add ReactBits registry to components.json
- [ ] (Optional) Set up Remotion for video content

Untitled