JanSathi AI - Project Progress Summary

Date: February 2026 **Current Phase:** Phase 2: Production Backend (Sprint 5, 6, 7 Completed)

What is JanSathi AI?

JanSathi AI is a bilingual, voice-first AI assistant designed for rural India. It offers an intuitive chat interface bridging the digital divide by providing accessible information and services. The project handles multiple domains through three core modules:

- **JanSeva (Citizen Services):** Government schemes, document checklists, and grievance forms.
- **JanKrishi (Agriculture):** Crop diagnosis, weather forecasts, and Mandi (market) prices.
- **JanVyapar (Business):** Product showcasing and business support for rural entrepreneurs.

Current Architecture

The project successfully utilizes a modern V2 decoupled architecture:

1. **Frontend (UI Shell):** Built with Next.js 15, React 19, Tailwind CSS v4, and Zustand. It features dynamic module-specific theming, Framer Motion animations, and PWA capabilities.
2. **Backend (Logic & API):** Express server leveraging TypeScript, routing all heavy operations and database queries.
3. **Database:** PostgreSQL on Supabase, managed via Prisma ORM for relational data (Users, Conversations, Messages, Analytics).
4. **AI Capabilities:** Groq Llama 3.3 70B runs the AI inference with an Intent Router to dynamically inject module-specific JSON knowledge bases.

Accomplishments To Date

Phase 1: MVP Core (✅ 100% Completed)

- Designed and built the Next.js 15 frontend shell with a 5-pillar design system.
- Successfully implemented Groq Llama 3.3 integration and an AI Intent Router.
- Built UI components for all three modules (`SchemeCard` , `CropDiagnosis` , `MandiPrices` , etc.).
- Created the "CommPulse" Analytics Dashboard and PWA service workers for offline caching.

Phase 2: Production Backend (✅ Sprints 5, 6, and 7 Completed)

- **Database Migration:** Successfully migrated from an in-memory database to a real PostgreSQL database on Supabase using Prisma ORM.
- **Backend Hardening:** Implemented session-based authentication with httpOnly cookies, structured JSON logging, and a hardened middleware pipeline (rate limiting, CORS).
- **Live Data APIs Integrated (Free Tier):**
  - **Open-Meteo:** real-time 7-day weather forecasting and farming advisories.
  - **Mandi Prices Dataset:** integration for Hindi/English search for crop prices across mandis.
  - Dynamic chat context injection where JanKrishi auto-detects queries requiring weather or price data.

Next Steps

Currently, the focus is finalizing the **V2 Architecture Split** (Sprint 11 & 12).

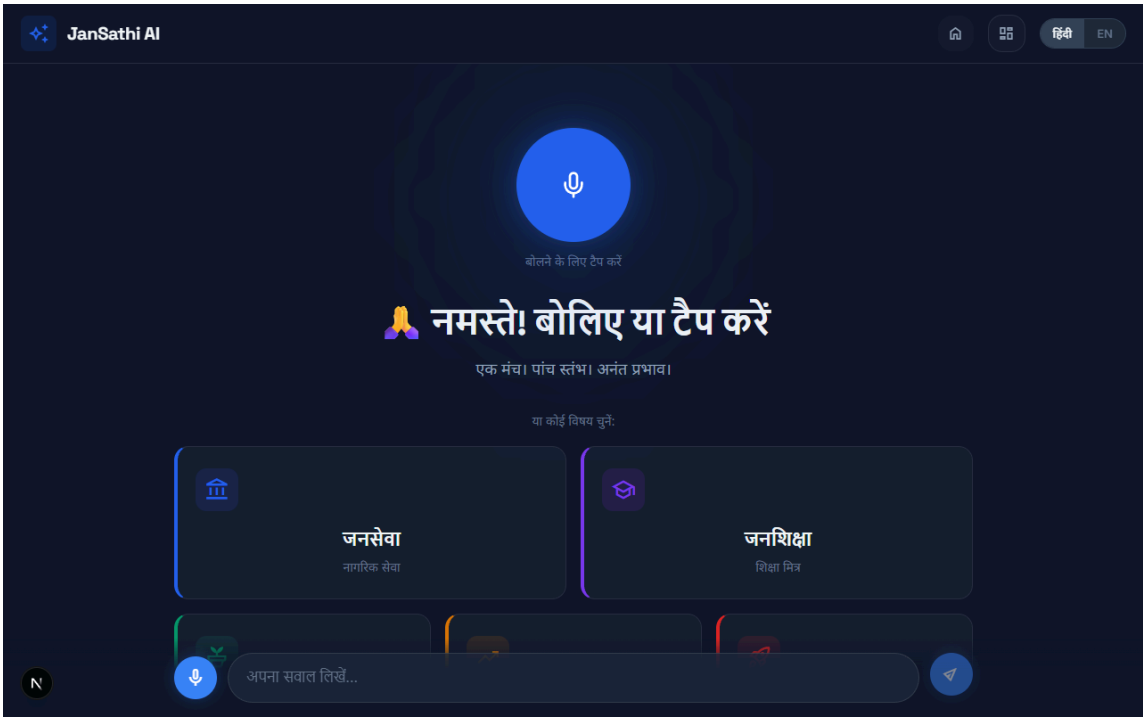
- Completing the extraction of all API routes from Next.js to the discrete Express backend.

- Deploying the backend to a production environment (Render).
- Upcoming UX refinement for the UI shell.

Visual Progress (Screenshots)

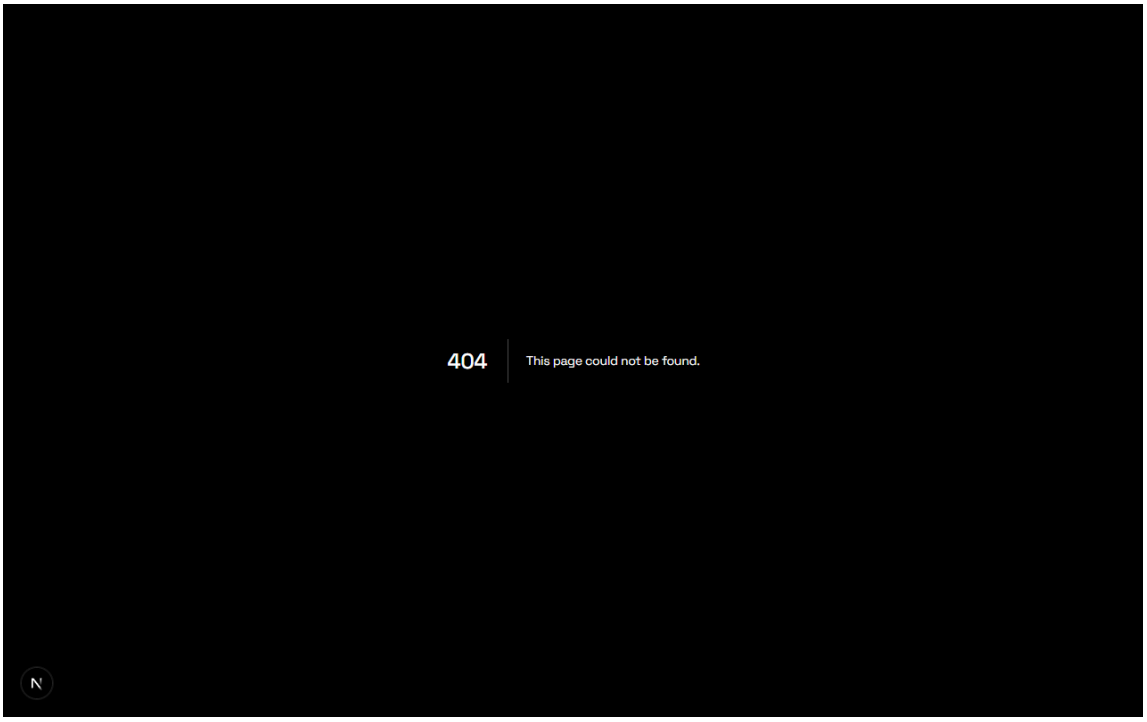
Home Dashboard

(The main conversational interface and entry point)



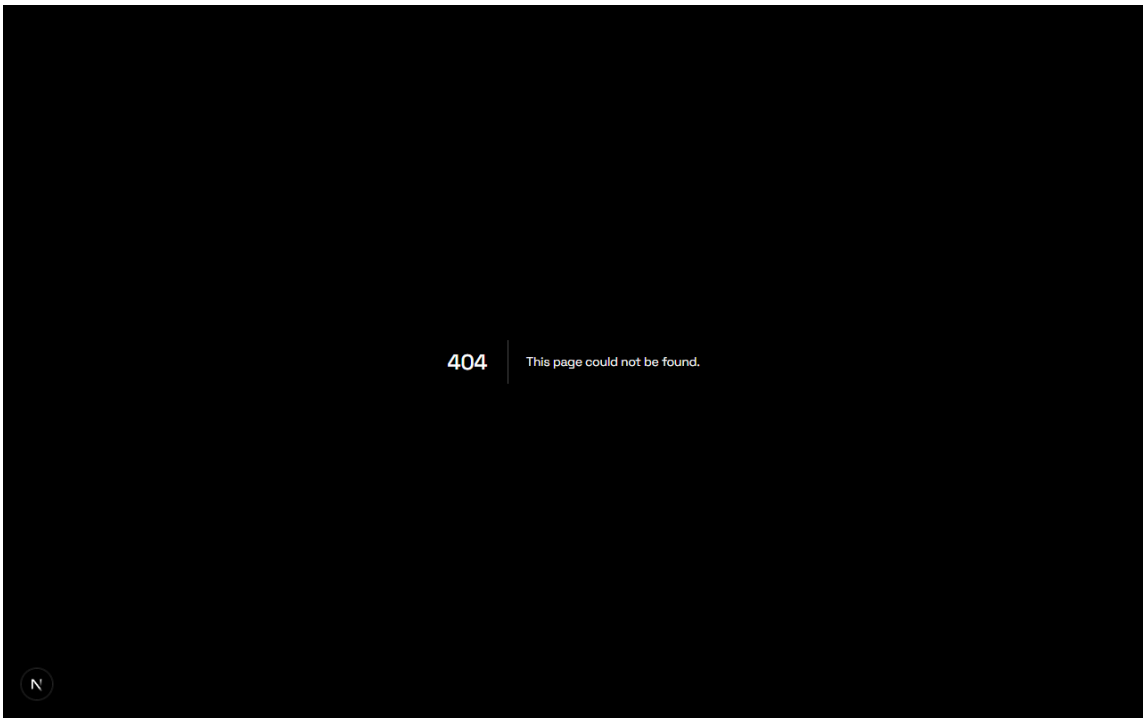
JanSeva Module

(Citizen Services and Schemes)



JanKrishi Module

(Agriculture, Weather, and Crops)



JanVyapar Module

(Business & Commerce)