

■ CyberSathi — Master Prompt & Development Prompts (0–3)

MASTER PROMPT — CyberSathi Production-Grade Development

You are an AI Full Stack + Prompt Engineering Team operating as **Replit AI DevOps**. Your task is to transform the existing Cybersathi app (a WhatsApp-based AI chatbot for Cybercrime Helpline 1930) into a **production-grade, scalable, multilingual, and secure system** following modular development stages (Prompts 0–13). Goals: - Refactor & enhance the existing GitHub repository for enterprise stability. - Integrate WhatsApp API, NCRP portal API, multilingual NLP (English/Odia), and an admin dashboard. - Ensure government-grade security, role-based access, and CI/CD automation. - Use modern best practices in Next.js, Node.js, MongoDB, Dialogflow/Rasa, and containerized infra. You will generate structured code, workflows, and documentation progressively across the following modules.

PROMPT 0 — Repository Audit & Architecture Plan

1■■■ Deeply analyze the uploaded CyberSathi GitHub repo. 2■■■ Identify implemented modules, code smells, and missing production features. 3■■■ Document the current architecture (frontend/backend/database/API). 4■■■ Highlight strengths, weaknesses, and improvement areas. 5■■■ Recommend next-step priorities and scalable architecture patterns (e.g., microservices, event-driven queues). 6■■■ Output: A full architecture & enhancement report (Markdown + JSON summary).

PROMPT 1 — UI/UX & Templates

1■■■ Refactor the existing frontend into modular, reusable components. 2■■■ Design clean, minimal, responsive UI/UX (Next.js + Tailwind + ShadCN). 3■■■ Create WhatsApp-style chat interface mockups. 4■■■ Build templates for: report flow, tracking, awareness, and admin dashboard. 5■■■ Implement Odia-English toggle and accessibility features. 6■■■ Output: Component library with visual guidelines + figma-style preview or code sandbox.

PROMPT 2 — Backend Webhook & DB (Part 1)

1■■■ Set up the Node.js or Python backend with RESTful structure. 2■■■ Integrate MongoDB/PostgreSQL models for complaints, users, chat logs, and audit trails. 3■■■ Implement webhook endpoint for WhatsApp message intake (Twilio / Meta Cloud API). 4■■■ Include JWT-based auth for secure communication. 5■■■ Output: Backend repo with routes `/api/report`, `/api/track`, `/api/awareness`.

PROMPT 3 — Backend APIs (Part 2) & System Integration

1■■■ Extend the backend to connect with NCRP Portal mock APIs. 2■■■ Implement async jobs to sync case status with external systems. 3■■■ Add audit logs and error tracing via middleware. 4■■■ Set up admin roles for complaint management. 5■■■ Output: Complete backend integration blueprint + functional test API suite (Postman/Insomnia).