Governance.ai

Compliance Bot (GST Filing Verification System)

1. Problem Statement

Indian businesses incur significant financial losses every year due to **avoidable GST non-compliance penalties**, including late fees and penal interest. These penalties are rarely caused by an inability to pay tax, but rather due to **verification failures** such as:

- Missed statutory deadlines
- Assumptions that returns are filed when they are only saved as drafts
- Lack of real-time confirmation from government systems
- Delayed receipt of GST notices, by which time penalties have already compounded

2. Proposal

The proposal is to build **Governance.ai – Free Compliance Bot**, an automated compliance verification system that:

- Accepts a GSTIN from a user via WhatsApp
- Periodically verifies GST filing status using a legal, API-based GST data provider
- Determines filing status (Filed / Pending) using deterministic rules
- Sends proactive alerts before statutory deadlines
- Operates on free or near-zero-cost infrastructure

3. Workflow Overview

- ➔ User sends GSTIN to Governance.ai WhatsApp number
- ➔ System validates GSTIN format
- ➔ GST filing data is fetched using a government-linked wrapper API
- ➔ Latest filed return period is identified
- ➔ Expected statutory filing period is calculated
- ➔ Filing status is derived by comparison
- ➔ User receives immediate confirmation (Filed / Pending)
- ➔ System continues automated monitoring near due dates

4. Technical Stack

Backend - Python, FastAPI (API orchestration, business logic, webhook handling)

Database - PostgreSQL Supabase (Store GSTIN, phone number, check status, timestamps)

Messaging - WhatsApp Cloud API Meta (User interaction, alerts, confirmations)

Data Source - GST Wrapper API via RapidAPI (Fetch GST return metadata without scraping or CAPTCHA bypass)

Scheduler - GitHub Actions/Cron-job.org (Trigger automated daily or deadline-based checks)

Hosting- Render (Host FastAPI backend with HTTPS support)

5. Detailed Implementation Process

Phase 1: GST Data Source Integration (Obtain reliable GST return metadata legally)

- Register on RapidAPI
- Subscribe to GST Return Status API
- Test endpoint with real GSTINs
- Validate availability of:
  - Latest GSTR-3B filing period
  - Sync timestamps
- Document API response structure and rate limits
- Confirm working API and Stable JSON response schema

Phase 2: Backend Development (Implement core logic and APIs)

- Create FastAPI project structure
- Implement GSTIN format validation
- Create API client for GST wrapper API
- Implement filing status derivation logic:
  - Calculate expected filing period
  - Compare with latest filed period
- Create webhook endpoint to receive WhatsApp messages
- Implement response handling and error cases

Phase 3: Database Setup (Persist user and compliance data)

- ➔ Create Supabase project
- ➔ Define tables:
  - ◆ users (phone, gstin, created_at)
  - ◆ compliance_status (gstin, latest_period, last_checked)
- ➔ Store and update filing status per check

Phase 4: WhatsApp Cloud API Integration

- ➔ Create Meta Developer account
- ➔ Create WhatsApp Cloud API app
- ➔ Obtain Phone Number ID and access token
- ➔ Configure webhook callback URL
- ➔ Implement message templates:
  - ◆ Filing confirmed
  - ◆ Pending alert
  - ◆ Deadline warning
- ➔ Test end-to-end message delivery

Phase 5: Scheduler & Automation

- ➔ Create scheduled job (daily or deadline-based)
- ➔ Query all registered GSTINs
- ➔ Re-check filing status
- ➔ Trigger alerts if:
  - ◆ Filing is pending
  - ◆ Due date is approaching
- ➔ Log execution results

Phase 6: Hosting & Deployment

- ➔ Push code to GitHub repository
- ➔ Deploy FastAPI app to Render / Railway
- ➔ Configure environment variables securely
- ➔ Verify HTTPS endpoint
- ➔ Re-validate WhatsApp webhook connectivity

Phase 7: Testing & Validation

1. Test with multiple GSTINs
2. Validate Filed vs Pending logic across dates
3. Simulate deadline scenarios
4. Validate message clarity and timing