Task 2: Project Vaani Sentinel X — Phase 2: "Pravaha" (Flow)

Project Context:

Now that the secure prototype backend is complete, your next mission is to expand Vaani Sentinel into a multilingual, production-grade autonomous system with adaptive publishing, voice-first enhancement, and smarter moderation.

This will be a critical building block of the Sanatan AI Engine, scaling beyond just English to handle Hindi, Sanskrit, and later vernacular Indian languages, plus smart, sentiment-tuned output for different platforms.

Your Mission: Build Phase 2 with These Key Upgrades

1. Agent F: Multilingual Content Pipeline

- Expand Agent A (Knowledge Miner) to handle Hindi and Sanskrit text ingestion.
- Add automatic language detection (language to fasttext).
- Auto-route the content for multilingual processing (different pipelines per language).

2. Agent G: Adaptive AI Writer and Voice Generator

• Enhance Agent B (AI Writer) to adapt style:

- Formal tone for LinkedIn
- Casual tone for Instagram
- Neutral, devotional tone for Sanatan voice assistants
- Allow dynamic TTS voice selection based on language (Hindi/Sanskrit/English voices).
- Use ElevenLabs Multilingual (if available) or fallback to Google Cloud Text-to-Speech.

3. Agent H: Sentiment Tuner

- New micro-agent that can adjust sentiment:
 - Optionally adjust the emotional tone (uplifting, neutral, devotional) of content before final generation.
 - Sentiment tuning options must be selectable at runtime (CLI or basic API parameter).

4. Agent I: Context-Aware Platform Targeter

- When simulating posts:
 - Tailor hashtags, post formats, and audio lengths according to the platform.
 - Examples:
 - Instagram: Emojis + 3–4 hashtags

- Twitter: 1–2 hashtags
- Spotify: 30-sec TTS audio intro + outro

5. Security + Compliance Layer Upgrade

- Expand Agent E (Security Guard):
 - Add basic detection of harmful religious bias triggers (simulate with predefined dummy cases).
 - Encrypt multilingual archives separately by language.
 - Include checksum generation for each archive for verification.

6. Dashboard and CLI Upgrades

- Enhance the web UI (optional) to display language and sentiment metadata.
- Create a simple Command Center CLI where you can:
 - Run any agent manually
 - See process logs
 - Kill or restart a pipeline

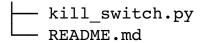
Stack Expectations

• Frontend: (optional upgrades) React.js / Next.js

- Backend: Python (FastAPI or Flask lightweight upgrade or Supabase if possible)
- Voice: ElevenLabs Multilingual / Google TTS/Spectral/ any other
- AI Models: OpenAI GPT-4 / Local LLMs via Ollama/ eventually our own.
- Security: Regex + basic ML flagging + Encryption
- DevOps: Local or Supabase/MongoDB + GitHub version control

Folder Structure Expansion

```
vaani-sentinel-x/
- agents/
    - miner sanitizer.py
    ai writer voicegen.py
    - scheduler.py
    - publisher sim.py
   — security guard.py
    - multilingual pipeline.py <-- NEW</pre>
    sentiment_tuner.pyadaptive_targeter.py
                                    <-- NEW
                                      <-- NEW
 web-ui/
  ___ nextjs-voice-panel/
 cli/
  __ command_center.py
                                      <-- NEW
 · content/
   — raw/
    — structured/
    - content ready/
    - multilingual ready/
                                    <-- NEW
  logs/
  scheduler db/
- archives/
   — encrypted_eng/
    - encrypted_hin/
    - encrypted san/
```



Deliverables

- Updated GitHub repo (with clear commit history and folder structure)
- A short video demo (showing multilingual + sentiment-adjusted output)
- Updated README detailing:
 - New agents added
 - Libraries used
 - Blockers, challenges faced, improvements made

Note on Time Expectation:

Although there's no hard timeline, ideally this Phase 2 is designed to be completed within 48 working hours.

Let's go carefully, Best of luck!