Challenging Task Brief: Project Vaani Sentinel X

Duration: ~16–18 Hours

Level: Advanced

Track: AI x Cybersecurity x Full Stack Automation x Voice Interfaces

Project Context for Candidate (Visible)

You're tasked with building an AI-integrated voice-first platform that auto-generates platform-specific content from structured knowledge, simulates social media scheduling, and ensures end-to-end modularity and security. This is meant to serve as a blueprint for deploying secure AI systems in real-time public environments.

This is not a UI-centric project. Your focus is backend intelligence, agent orchestration, and voice-driven interaction pipelines. You're encouraged to be modular, agent-driven, secure, and scalable in your design.

Your Mission – Build an Autonomous Agent-Based System with the Following Flow:

1. Agent A: Knowledge Miner & Sanitizer

- Accept raw CSV or JSON data input (facts, quotes, micro-articles, etc.)
- Structure it into modular "content blocks" for further processing
- Add a content verification layer:
    - Auto-check for profanity
    - Detect if content is neutral/biased
    - Validate against a dummy "truth-source.csv" file

2. Agent B: AI Writer & Voice Synth Generator

- Use OpenAI API (or a local model if offline) to:
    - Convert each block into:
        - A tweet (≤ 280 characters)
        - A 1-paragraph post for Instagram/LinkedIn

        - A 20–30 second voice script for use in an AI assistant
- Generate the TTS (Text-to-Speech) file using:
    - Vapi / IBM Watson / Google TTS / ElevenLabs (if access)
- Save outputs in a /content_ready folder, maintaining versioning

3. Agent C: Secure Web Interface (Minimalist UI)

- Next.js / React frontend with:
    - Login system (Firebase or JWT)
    - Panel to view and play AI-generated content (with audio)
    - Download / Copy-ready buttons for each content type
    - Show publishing time and versioning

4. Agent D: Scheduler & Publisher Simulator

- Schedule each post as if it would be published on:
    - Twitter (tweet)
    - Instagram (longer post)
    - Spotify/Voice Platforms (TTS audio)
- Simulate these by:
    - Storing to scheduled_posts/
    - Creating dummy POST calls to mock endpoints
    - Logging time, platform, and result to Firebase or a local DB

5. Agent E: Security & Ethics Guard (Advanced Subsystem)

- Add a mini-AI/regex tool to:
    - Flag controversial content (religion, politics, bias)
    - Simulate an alert dashboard (even if backend-only)
    - Encrypt content archives with basic Python encryption
    - Include a “kill switch” to wipe data if misuse is detected
    -

Stack Expectations

- Frontend: React.js / Next.js (minimal UI)
- Backend: Python (FastAPI), Firebase OR local storage
- Voice: IBM Watson TTS / Web Speech API / Vapi
- AI: OpenAI, Ollama, or local LLMs (text generation)
- Security: Regex + encryption + flagging heuristics
- DevOps: GitHub repo + README + folder structure

Bonus Challenges (Optional, Worth Extra Credit)

- Add logging system with user ID, timestamps, and agent actions
- Create your own “Command Center CLI” to run agents individually
- Create a dashboard showing how each post scored on “ethics”, “virality”, and “neutrality”

Folder Structure Expected

```
vaani-sentinel-x/
├── agents/
│   ├── miner_sanitizer.py
│   ├── ai_writer_voicegen.py
│   ├── scheduler.py
│   ├── publisher_sim.py
│   └── security_guard.py
├── web-ui/
│   └── nextjs-voice-panel/
├── content/
│   ├── raw/
│   ├── structured/
│   └── content_ready/
├── logs/
├── scheduler_db/
├── kill_switch.py
└── README.md
```

Deliverables

- GitHub repo with full code + setup steps
- A video demo (screen recording, 3–5 min)
- Report/README outlining how agents work, what libraries were used, and any blockers you faced

Evaluation Rubric

Category	Points
Modular Agent Design	20
AI + TTS Integration	20
Secure Content Processing	20
Voice-Ready UI	15
Automation & Scheduling	15
Bonus Implementations	10
Total	100