Project Name: MAILER

System Architecture: Cloud-Native AI Email Assistant

Model Architecture: Llama-3.3-70b-versatile (Fine-tuned via Groq LPU™ Inference Engine)

Date: January 12, 2026

"The secret of change is to focus all of your energy, not on fighting the old, but on building the new."

~Socrates

Short Description

MAILER is a cloud-based, AI-driven automation platform designed to revolutionize inbox management through autonomous triaging and contextual drafting. Utilizing a fine-tuned **Llama-3.3-70b** model, the system executes user-defined "Manifestos" to respond to correspondence in real-time, providing users with high-level summaries and an interactive AI interface for seamless oversight.

Comprehensive Purpose

The primary objective of Project **MAILER** is to serve as a high-intensity technical laboratory for mastering the intersection of full-stack web engineering, cloud-native automation, and Large Language Model (LLM) orchestration. This initiative moves beyond theoretical study, challenging the team to architect a secure, production-grade web application that leverages a fine-tuned **Llama-3.3-70b-versatile** model via the **Groq API**.

By constructing **MAILER**, the team will navigate the complexities of real-time data streaming, asynchronous background processing, and secure OAuth2 integration with the Google ecosystem. The project is designed to cultivate expertise in building "Privacy-First" AI systems—utilizing **FastAPI** and **Firestore** to manage user metadata and "Manifesto" logic without the inherent risks of long-term local email storage. **MAILER** is, fundamentally, a journey in technical craftsmanship, collaborative problem-solving, and the practical deployment of autonomous AI agents.

System Operations & Workflow Statement

The operational lifecycle of **MAILER** is engineered to ensure a frictionless transition from user onboarding to fully autonomous email management. The following describes the sequence of operations:

1. **Authentication & Identity:** Users interface with a professional landing page and proceed to a secure authentication gateway managed by **Firebase**. Upon verification, the system establishes a unique user profile within the **Firestore** database.
2. **Configuration & The "Manifesto":** New users enter a mandatory Setup Wizard. Here, they authorize **MAILER** to access their Google account via **OAuth2**, granting the system a **Refresh Token** for background operations. Users then define their "Manifesto"—a set of specialized instructions governing the AI's persona, tone, and response logic—and select a lookback duration (ranging from 1 to 7 days).
3. **Autonomous Background Execution:** Once configured, the **FastAPI** backend initiates a background worker. This service periodically polls the Google API to identify unread or repliable threads within the defined duration. Using **LangChain**, the system fetches thread context, pairs it with the user's "Manifesto," and requests a draft from the **Groq** inference engine.
4. **Action & Logging:** The system dispatches the AI-generated response via **SMTP**. Simultaneously, a lightweight summary of the incoming message and the outgoing reply is stored in **Firestore** Activity Logs. No full email bodies are retained, ensuring maximum data privacy.
5. **Interactive Monitoring (The AI Bubble):** Users can access an interactive "AI Bubble" chat interface. This component provides real-time status updates and allows users to query **MAILER** regarding recent activities.
6. **Persistence & Reporting: MAILER** remains operational regardless of the user's session status. Upon return, the user receives a "Brief Report"—a synthesized summary of all correspondence handled during their absence, derived from the stored Activity Logs.

Organizational Roles & Expectations

Project Manager [GitHub Projects, WhatsApp]

- **Description:** The central coordinator responsible for the project's technical roadmap, task delegation, and milestone delivery.
- **Expectations:** Facilitate real-time communication via WhatsApp; manage the 30-day "Lean" sprint on GitHub Projects; ensure all 15 members document progress in GitHub Issues to maintain transparency and prevent code overlap.

Fullstack Developer [HTML5, Tailwind CSS, JavaScript]

- **Description:** Architect of the user-facing web dashboard and the interactive onboarding experience.
- **Expectations:** Design and implement a responsive Landing Page, a secure Setup Wizard, and the "AI Bubble" chat interface; ensure the frontend communicates efficiently with the FastAPI backend.

Backend Developer [Python, FastAPI, Google OAuth2]

- **Description:** Engineer in charge of the server-side logic, API development, and external service integration.
- **Expectations:** Implement the "Read-on-Demand" logic using the Google API; develop secure endpoints for the dashboard; manage the background scheduler that monitors emails while the user is offline.

AI Quality Specialist [Groq API, LangChain, Llama-3.3-70b]

- **Description:** Strategist focused on the behavior, accuracy, and contextual intelligence of the AI model.
- **Expectations:** Design the "Summary Engine" for absence reporting; optimize prompt chains to ensure the AI strictly adheres to the user's "Manifesto"; manage the flow of thread history into the model.

Data Engineer [Firestore, Google Cloud IAM]

- **Description:** Specialist responsible for data structure, user metadata storage, and security permissions.
- **Expectations:** Securely manage the storage of Google Refresh Tokens; design the Firestore schema for "Activity Logs" and user settings; ensure data privacy by preventing the storage of full email bodies.

Automations Engineer [Vercel, Render, GitHub Actions]

- **Description:** Engineer focused on the deployment pipeline, background workers, and system uptime.

- **Expectations:** Set up the CI/CD pipeline via GitHub Actions; deploy the frontend to Vercel and the backend to Render; ensure the background bot remains operational for live testing and demonstration.

Implementation Timeline

(30-Days)

To ensure the delivery of a functional web-based Proof of Concept within 30 days, the team will adhere to the following highly parallelized roadmap. This schedule prioritizes the "Read-on-Demand" architecture and the establishment of the FastAPI-to-Groq bridge.

Week 1: Infrastructure & Authentication

- **Project Management:** Initialization of the **GitHub Projects** board and establishment of the **WhatsApp** communication protocols.
- **Environment Setup:** Configuration of the **FastAPI** development environment and **Firebase** project for user authentication.
- **Security:** Implementation of the **Google OAuth2** flow to acquire and securely store **Refresh Tokens** in **Firestore**.
- **Frontend:** Development of the high-fidelity Landing Page and the initial User Login interface.

Week 2: Core Logic & AI Integration

- **Data Pipeline:** Development of the "Read-on-Demand" logic to fetch email thread headers via the **Google API** without persistent storage.
- **AI Engine:** Integration of **LangChain** with the **Groq API** to communicate with the **Llama-3.3-70b-versatile** model.
- **The Manifesto:** Implementation of the **Firestore** schema to store user-defined response instructions and duration settings.
- **Backend:** Construction of API endpoints to bridge the frontend "Setup Wizard" with the backend configuration logic.

Week 3: Automation & The Interactive Dashboard

- **Background Workers:** Development of the asynchronous listener service to monitor inboxes and draft replies while users are offline.
- **Activity Logging:** Implementation of the "Summary Engine" to generate and store lightweight interaction logs in **Firestore**.
- **UI/UX:** Finalization of the **"AI Bubble"** chat interface and the primary user dashboard for monitoring recent activities.
- **Drafting Logic:** Rigorous testing of the "Manifesto" adherence to ensure AI responses match user expectations.

Week 4: Deployment & Validation

- **Refinement:** Implementation of the "Absence Report" feature, synthesized from **Week 3** activity logs.
- **Deployment:** Orchestration of the CI/CD pipeline via **GitHub Actions** for deployment to **Vercel** (Frontend) and **Render** (Backend).
- **Quality Assurance:** Stress testing of the background worker and final security audit of the **Firestore** permission rules.
- **Documentation:** Completion of a technical post-mortem and a brief user onboarding guide to finalize the learning journey.

Closing Statement

MAILER represents a sophisticated synthesis of modern web architecture and autonomous artificial intelligence. This project is not merely a build; it is a collaborative masterclass in engineering a privacy-first, AI-driven future. Over the next thirty days, your individual contributions will converge to transform a complex vision into reality.

Let us proceed with excellence.