📧 AI Executive Email Assistant

An intelligent, agentic email manager that uses **Groq LLM** to triage your inbox. This assistant automatically fetches unread emails, categorizes them, assigns priority levels, drafts suggested replies, and syncs meeting requests to **Google Calendar**.

✨ Key Features

- ⚡ **Groq-Powered Speed:** Uses LPU Inference Engine for near-instant email analysis.
- 📥 **Unread-Only Triage:** Specifically targets unread messages to keep your workflow focused.
- 📊 **Smart Classification:** Automatically identifies **Category** (Work, Finance, Social) and **Priority** (High, Medium, Low).
- 🤖 **Suggested Replies:** Generates professional, 2-sentence drafts that you can edit and send instantly.
- 🗓️ **Calendar Integration:** One-click scheduling for detected meeting requests.
- 🎨 **Streamlit Dashboard:** A clean, modern UI with color-coded priority indicators and orange export buttons.

🧠 System Architecture & Workflow Nodes

The assistant follows a "Node-Based" logic flow to process your data securely and efficiently:

Node 1: The Fetcher (Gmail API)

- **Filter:** is:unread
- **Action:** Connects to Gmail v1 API using OAuth 2.0 to pull raw subject lines, snippets, and sender info.

Node 2: The Intelligence Node (Groq Cloud)

- **Model:** Groq
- **Logic:** Converts unstructured email text into a structured JSON schema containing category, priority, and intent.

Node 3: The Priority & Intent Node

- **Logic:** Evaluates the AI's output to determine UI styling:
    - 🔴 **High:** Critical/Action-oriented items.
    - 🟡 **Medium:** Standard correspondence.
    - 🟢 **Low:** Updates/Newsletters.

Node 4: Execution Node (Action)

- **Gmail Dispatch:** Encodes edited replies into base64 for secure transmission.
- **Calendar Injector:** Formats meeting snippets into ISO 8601 timestamps for Google Calendar insertion.

🚀 Getting Started

1. Prerequisites

- Python 3.10+
- Google Cloud Project credentials.json with Gmail and Calendar APIs enabled.
- Groq API Key.

2. Installation

PowerShell

Create virtual environment

python -m venv venv

.\venv\Scripts\activate

Install strictly required libraries

pip install streamlit langchain-groq google-auth-oauthlib google-api-python-client python-dotenv pandas

3. Setup

1. Create a .env file:

Code snippet

GROQ_API_KEY=your_groq_key_here

2. Ensure credentials.json is in the root directory.
3. **Delete token.json** if you are resetting permissions for the first time.

4. Running the App

PowerShell

streamlit run app.py

📁 Project Structure

- app.py — The core application and Groq integration.
- Assistant – Agent langGraph logic

- auth_setup -- console.cloud.google setup

  https://console.cloud.google.com/

- gmail_check: Gmail setup
- .env — Protected API keys.(Groq API)
- requirements.txt — List of dependencies .

📜 License