📨 AI Personal Email Assistant (LangGraph + Ollama)

An intelligent, privacy-first email management agent built with **LangGraph** and **Local LLMs (Llama 3)**. This assistant fetches unread emails, categorises them by priority and type, and waits for human approval before storing the results.

🌟 Key Features

- **Privacy-First:** Uses **Ollama** to run Llama 3 locally. Your email data never leaves your machine.
- **Stateful Workflow:** Built with **LangGraph** to manage complex, multi-step logic and "memory."
- **Human-in-the-Loop:** A dedicated "Review" node ensures the AI doesn't make mistakes without your approval.
- **Gmail Integration:** Securely connects to the Gmail API using OAuth2.0.
- **Smart Categorization:** Automatically sorts mail into **Family, Friends, Shopping, Junk,** and **High-Priority**.

🏗 The Architecture

The assistant follows a linear **StateGraph** workflow:

1. **Fetch Node:** Connects to Gmail and retrieves the latest unread messages.
2. **Analysis Node:** Passes email snippets to **Llama 3** for classification and priority flagging.
3. **Review Node:** Pauses execution and displays a report in the terminal for user confirmation.
4. **Storage Node:** Upon approval (yes), saves the final analysis to a processed_emails.json file.

🛠 Tech Stack

- **Framework:** LangGraph
- **LLM:** Ollama (Llama 3)
- **Language:** Python 3.10+
- **APIs:** Google Gmail API

🚀 Getting Started

1. Prerequisites

- Python 3.10 or higher
- Ollama installed and running (ollama pull llama3)
- Google Cloud Project with Gmail API enabled

2. Installation

PowerShell

```
# Clone the repository
git clone https://github.com/yourusername/email-assistant.git
cd email-assistant

# Setup virtual environment
python -m venv venv
./venv/Scripts/activate

# Install dependencies
pip install langgraph langchain-ollama google-api-python-client google-auth-oauthlib
```

3. Configuration

1. Place your credentials.json from Google Cloud in the root folder.
2. Run the setup script to generate your access token:

PowerShell

```
python auth_setup.py
```

4. Running the Assistant

PowerShell

```
python assistant.py
```

📊 Example Output

Plaintext

```
--- 📥 FETCHING RECENT EMAILS ---
--- 🤖 OLLAMA CATEGORIZING EMAILS ---

--- 📋 ANALYSIS REPORT ---
* Email 1: [Junk] - Spam subscription
* Email 2: [High Priority] - Job Interview Invitation
* Email 3: [Family] - Dinner plans for Sunday
```

Do you approve these actions? (yes/no): yes

--- 💾 STORING ANALYSIS TO FILE ---

🗺️ Roadmap

- [ ] **Auto-Draft:** Generate reply drafts for high-priority emails.
- [ ] **Calendar Sync:** Automatically add detected events to Google Calendar.
- [ ] **Folder Sorting:** Move Junk emails to a specific Gmail folder automatically.