AI-Powered Communication Assistant

Overview

This project is an AI-powered solution for automating the management of support-related emails in modern organizations. It leverages advanced language models to analyze, prioritize, and respond to customer queries, requests, and help tickets, all through a clean, interactive dashboard. The goal is to improve efficiency, response quality, and customer satisfaction while reducing manual effort for support teams.

Features

• **Email Retrieval & Filtering:**
• Loads emails from a CSV file (can be extended to IMAP/Gmail/Outlook APIs).
• Filters emails by subject keywords ("Support", "Query", "Request", "Help").
• Displays sender, subject, body, and date/time received.

• **Categorization & Prioritization:**
• Sentiment analysis (Positive, Negative, Neutral) using simple heuristics.
• Priority detection (Urgent/Not urgent) based on keywords ("immediately", "critical", etc.).
• Urgent emails are processed and displayed first.

• **Information Extraction:**
• Extracts contact details (phone, alternate email).
• Extracts customer requirements and sentiment indicators.
• Displays extracted info alongside raw email data.

• **Context-Aware Auto-Responses:**
• Uses LLM APIs (Gemini, Claude, Groq, etc.) to generate draft replies for urgent emails.
• Responses are professional, friendly, and context-aware.
• Only urgent emails receive AI-generated replies (to respect API quotas).

• **Dashboard / User Interface:**
• React frontend with a modern, dark-themed dashboard.
• Analytics section with stats (total emails, urgent, not urgent, sentiment breakdown).
• Toggle buttons to switch between urgent and not urgent emails.
• Centered, enlarged email cards for readability.
• Editable AI-generated responses before sending.

Technology Stack

• **Backend:** Python (Flask), pandas, requests, flask-cors, python-dotenv
• **Frontend:** React (JavaScript), CSS
• **LLM Integration:** Gemini API (default, easily switchable to Claude, Groq, OpenAI)
• **Storage:** CSV file (demo), can be extended to SQLite, MongoDB, PostgreSQL

Setup Instructions

1. Clone the Repository

```
git clone https://github.com/your-username/AI-Powered-Communication-Assistant.git
cd AI-Powered-Communication-Assistant
```

2. Backend Setup

• Create and activate a Python virtual environment (optional):
```sh
python -m venv venv
venv\Scripts\activate # On Windows
source venv/bin/activate # On Mac/Linux
```
• Install dependencies:
```sh
pip install flask pandas requests flask-cors python-dotenv
```
• Add your Gemini API key to a `.env` file:
```env
GEMINI_API_KEY=your_actual_key_here
```
• Start the backend server:
```sh
python app.py
```
The backend will run at `http://localhost:5000`.

3. Frontend Setup

• Go to the frontend directory:
```sh
cd frontend
```
• Install dependencies:
```sh
npm install
```
• Start the frontend server:
```sh
npm start
```
The frontend will run at `http://localhost:3000`.

Usage

• Open the frontend in your browser.

• View analytics and toggle between urgent and not urgent emails.
• Review and edit AI-generated responses before sending.
• Only urgent emails receive AI-generated replies (to respect API quotas).

Customization

• To use a different LLM API (Claude, Groq, OpenAI), update the `generate_response` function in `app.py` and set your API key in `.env`.
• To connect to a real email provider, replace the CSV loading logic with IMAP/Gmail/Outlook integration.
• Extend information extraction and analytics as needed for your organization.

Security

• API keys and secrets are stored in `.env` and excluded from version control via `.gitignore`.
• Never commit your API keys to GitHub.

Folder Structure

```
AI-Powered-Communication-Assistant/
■■■ app.py
■■■ requirements.txt
■■■ .env
■■■ .gitignore
■■■ README.md
■■■ 68b1acd44f393_Sample_Support_Emails_Dataset.csv
■■■ frontend/
■ ■■■ package.json
■ ■■■ public/
■ ■ ■■■ index.html
■ ■■■ src/
■ ■■■ App.js
■ ■■■ index.js
■ ■■■ components/
■ ■ ■■■ Dashboard.js
■ ■ ■■■ EmailCard.js
■ ■■■ styles/
■ ■■■ Dashboard.css
```

License

This project is for educational and hackathon use. Extend and customize as needed!

Contact

For questions or contributions, open an issue or pull request on GitHub.