## **Al Email Assistant Chatbot**

With NGRok

Github: https://github.com/Montegan/Ai-Email-Assistant

Developer: Simon Fasil Date: 10-16-2024

### **Introduction to AI Email Assistant**

- An Al-powered chatbot designed to assist businesses in automating email responses to customer comments on purchased products.
- To illustrate the capabilities of this Assistant the user comments are generated automatically by the Assistant.
- After generating a random comment then the Assistant first summarizes the comment, then extract the sentiment of the comment.
- Based on the above analysis it composes an email in the desired language and sends it automatically to the user.

### **Development environment - FrontEnd**

Set up a Node.js environment to develop the React app.

```
npm init -y
```

Create a React Project Using Vite

```
npm create vite@latest my-react-app --template react
```

Create a React Project Using Vite

```
cd my-react-app
npm install axios tailwindcss react-router-dom
```

Launch App on Localhost

```
npm run dev
```

### **Development environment - BackEnd**

Install Virtual Environment Package

pip install virtualenv

Create a Virtual Environment

virtualenv Email\_ai

Activate the Virtual Environment

.\Email\_ai\Scripts\activate

Install Required Packages

pip install flask flask-cors openai python-dotenv

Start Flask Server

flask run

### **Development-Backend**

First i loaded my environment variables to enable me access my keys securely

```
load_dotenv()
```

 Then i created my flask App instance. To help me setup my API Endpoint and Flask webserver.

```
app = Flask(__name__)
CORS(app)
```

 Then i created my opena ai instance to enable interact with the GPT-4o-mini model. As follows

```
client = OpenAI()
```

### **Development-Backend**

 First i created the "/originalComment" endpoint that is used to generate random user comment on a product. And sends the output comment to the frontend. It listens to Get requests

```
@app.route('/originalComment', methods=['GET'])
def create_Comment():
```

Then i implemented my "/composeEmail" Post RestAPI endpoint that is used to
extract the sentiment of the comment summarize the original comment and
then compose an email based on the provided above analysis. All the major
functionalities of the app are inside this endpoint.

```
@app.route('/composeEmail', methods=['POST'])
def compose_email():
```

### **Development- Frontend**

I started by designing the User Interface (UI) with a clear project name displayed prominently at the top for better identification.

#### **Text Areas for Generated Output:**

- Two text areas were added:
  - One to display the randomly generated customer comment.
  - Another to show the generated email response based on that comment.

#### **Action Buttons:**

- **Generate Comment Button**: I implemented a button to generate a random comment when clicked. This comment simulates feedback on a product.
- Generate Email Button: Another button was added to generate an email response based on the comment, providing a structured reply to the customer's feedback.

#### **Language Selection:**

 I included a dropdown button that allows the user to select the language for the email response, making the app versatile for different language preferences.

### NGrok-Setup

- Pip Install NGrok Locally:
  - And then run the following command to initialize an environment:

```
./ngrok config add-authtoken <YOUR_AUTH_TOKEN>
```

• Expose Your Localhost:

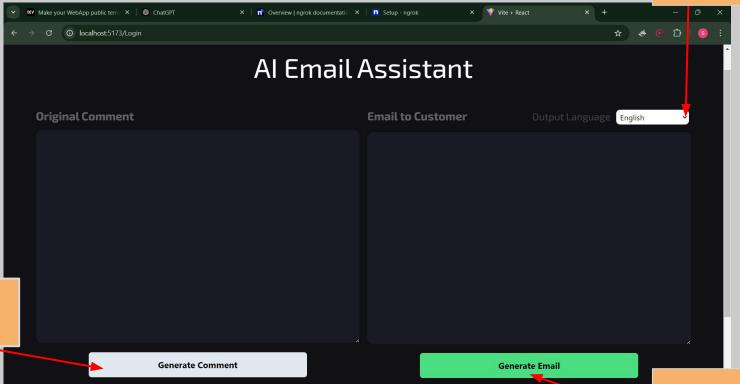
```
ngrok http 5000
```

 After starting NGrok, you'll see something like this in your terminal with the public url available in the forwarding section of the below output:

```
ngrok
Session Status
                              stesfats595@student.sfbu.edu (Plan: Free)
Account
Version
                              3.17.0
                              United States (California) (us-cal-1)
Reaion
deb Interface
                              http://127.0.0.1:4040
                              https://07b6-209-36-108-210.ngrok-free.app -> http://localhost:5000
Forwarding
Connections
                              tt1
                                                       rt5
                                                               p50
                                                                        p90
                                               rt1
                                                                        0.00
                                               0.00
                                                       0.00
                                                               0.00
```

### **Demo-App UI**

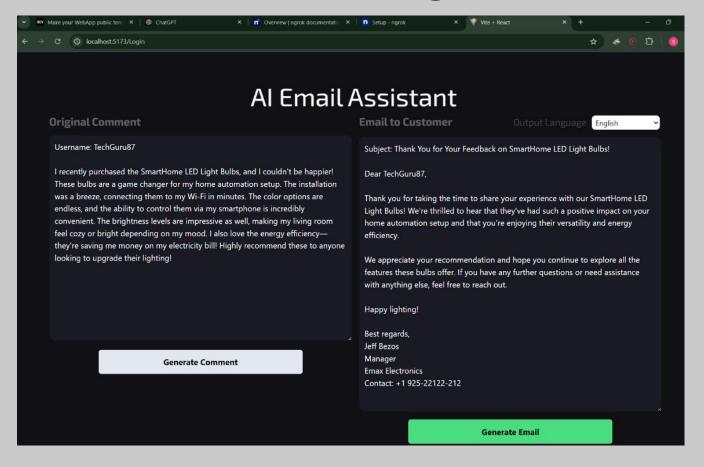
Step2: Click Choose Language



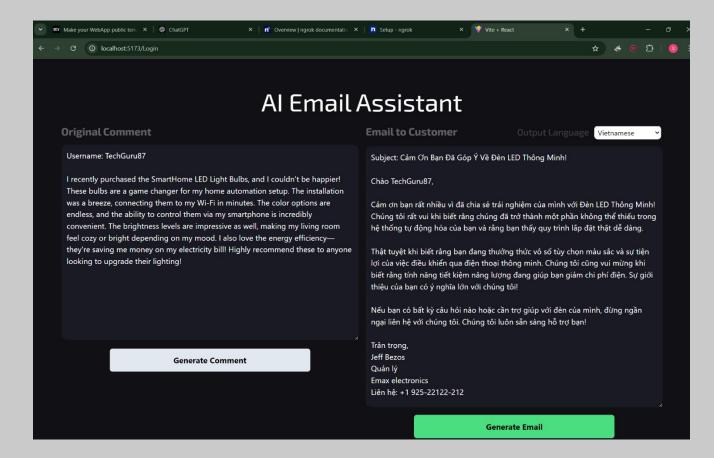
Step 1: Click
Generate comment

Step3: Click Choose Language

### **Demo- In English**



### **Demo-In Vietnamese**



# Thank you!

Github: https://github.com/Montegan/Ai-Email-Assistant

Developer: Simon Fasil Date: 10-16-2024