A New Generative Approach to Optimize the Network and Server Load of Websites (University Chatbot)

Project submitted to Dr. Yanqing Zhang for Artificial Intelligence (CS4810)

Authors: Manish Kolla & Ritesh Dumpala

Application Manual

This manual provides step-by-step instructions to set up and run the University Chatbot application locally.

Prerequisites

Before running the application, ensure you have the following:

1. **Gemini API Key**
   The Gemini API key is required for interacting with Generative AI. Follow these steps to obtain one:

   - Login to the Gemini AI Studio.
   - Sign in with your account or create a new one.
   - Click on **Get API Key** section and follow the prompts.
   - Generate a new API key and copy it.
     Note: Save the key securely as it will be used in the application.

2. **Bing Search API Key**
   The Bing Search API key is required for integrating web search functionality. Follow these steps to obtain one:

   - Visit the Microsoft Azure Portal.
   - Sign in with your Microsoft account or create a new one.
   - Go to **Create a Resource** and search for "Bing Search v7".
   - Set up a new resource for Bing Search and navigate to the **Keys and Endpoint** section.
   - Copy one of the provided API keys.
     Note: Save the key securely as it will be used in the application.

3. **Create the `config.py` File** To securely store your API keys, create a `config.py` file in the project directory and add the following code:

```
# config.py
GEMINI_API = "YOUR_API_KEY_HERE"
BING_API = "YOUR_API_KEY_HERE"
```

- Replace `"YOUR_API_KEY_HERE"` with your actual **Gemini API Key** and **Bing API Key**.

- Save this file in the same directory as your main application script (`app.py`).

Steps to Run the Application

1. Clone the Repository

Use the following command to clone the repository from your Git hosting platform (e.g., GitHub):

```
git clone https://github.com/manishkolla/GenAI_University_Chatbot
```

2. Navigate to the Project Directory

Change your working directory to the cloned repository folder:

```
cd <project-directory>
```

3. Check Python and pip Installation

Make sure Python and pip are installed on your system.

To check if Python is installed:

```
python --version
```

or

```
python3 --version
```

To check if pip is installed:

```
pip --version
```

If either is missing, download and install the latest version of Python from the official Python website.

4. Create and Activate a Virtual Environment

It is recommended to use a virtual environment to isolate dependencies.

For Windows:

1. Create a virtual environment:

   ```
   python -m venv venv
   ```

2. Activate the virtual environment:

```
venv\Scripts\activate
```

For Mac/Linux:

1. Create a virtual environment:

   ```
   python3 -m venv venv
   ```

2. Activate the virtual environment:

   ```
   source venv/bin/activate
   ```

---

5. Install Required Dependencies

Install all necessary Python libraries listed in the `requirements.txt` file:

```
pip install -r requirements.txt
```

---

6. Run the Flask Application

Run the Flask application using the following command:

```
python app.py
```

---

7. Access the Application Locally

Once the application starts, it will display a local URL (usually `http://127.0.0.1:5000/`). Open this URL in your web browser to access the application.

---

8. Test the Application

You can test the application by interacting with it through the web interface or API (depending on the implementation).

- Open the URL shown in the terminal (e.g., `http://127.0.0.1:5000/`) in your browser.
- Ask questions related to **Computer Science (CS)**, **Data Science departments**, or the **directory**. For example:
  - "What is the role of the CS department?"
  - "Can you provide information about the Data Science program?"
  - "Who is the head of the CS department?"
  - "Can you find a contact in the directory?"
- The application should respond with the relevant information or provide helpful answers based on its functionality.

Optional: Deactivate the Virtual Environment

Once done, deactivate the virtual environment using:

```
deactivate
```