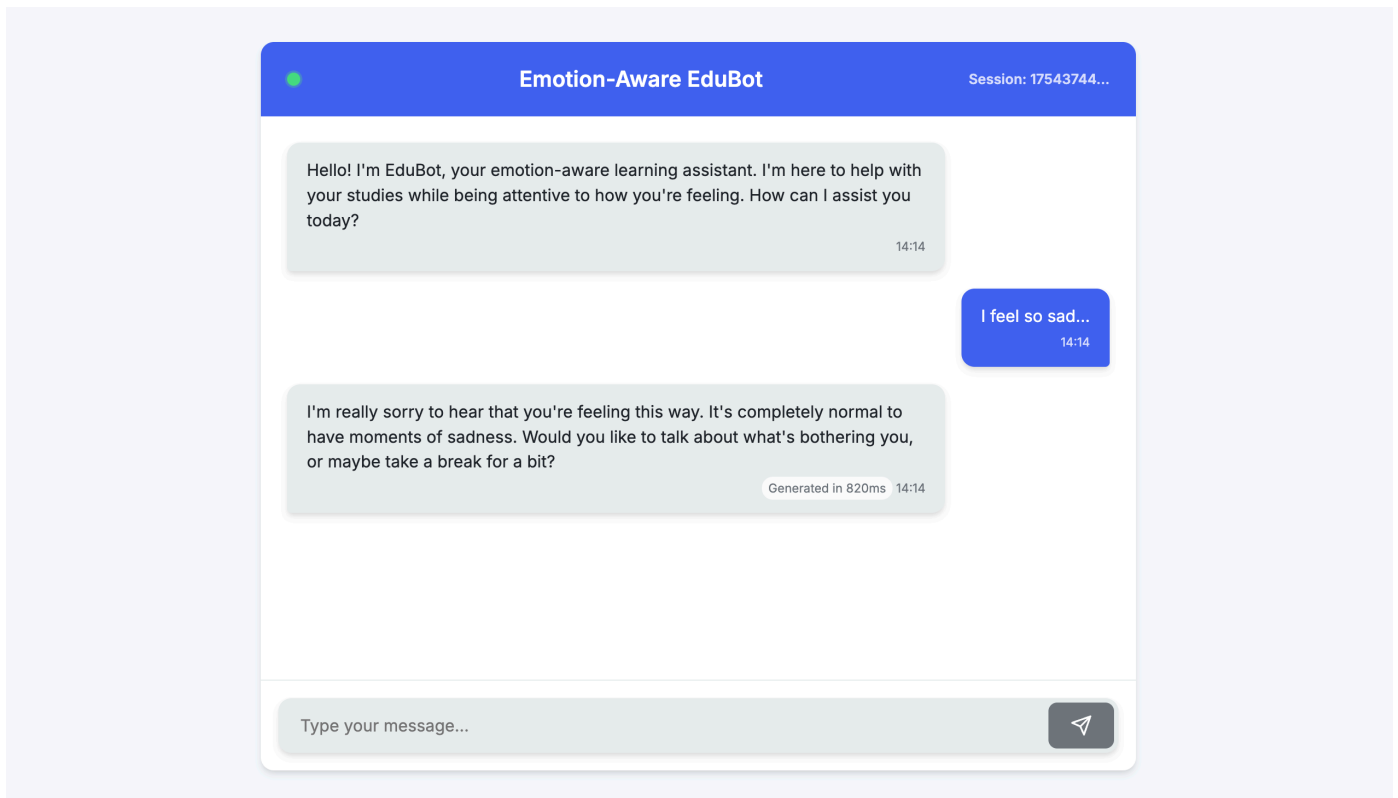


# Emotion-Aware EduBot

## Introduction

**Emotion-Aware EduBot is a proof-of-concept web-based chatbot designed to function as an empathetic AI tutor.** Its primary goal is to assist students with their academic questions while also being attentive to their emotional state. The application intelligently detects signs of negative emotions, such as frustration or stress, and adjusts its conversational approach to provide support and empathy before returning to the educational content.

This project showcases a more human-centered approach to AI interaction in education. The core technology combines a robust Python backend using the `Flask` framework with a dynamic JS frontend. It leverages the `OpenAI API` for natural language understanding and generation. A key feature is its `dual-method emotion detection`, which analyzes both explicit keywords in the user's text and implicit behavioral patterns captured on the client-side, such as typing speed, hesitation, and correction frequency. Responses are streamed in real-time to the user via `Server-Sent Events (SSE)` for a fluid and responsive chat experience.



## Environment Setup

Before running the application, you must configure your local environment.

### Prerequisites:

- Python 3.7+

### Dependencies:

This project relies on two main Python libraries: `Flask` for the web server and `openai` to interact with the language model. Create a file named `requirements.txt` in the root of your project directory and add the following lines:

```
1 Flask>=2.0
2 openai>=1.0
```

## API Key Configuration:

You need a valid API key from OpenAI. Open the `app.py` file and locate the `OpenAI` client initialization. Replace the placeholder API key with your own secret key.

It is recommended to use [CloseAI](#) to obtain more LLMs.

```
1 client = OpenAI(
2     base_url='https://api.openai-proxy.org/v1', # Don't change if you use CloseAI
3     # IMPORTANT: Replace with your actual key, preferably loaded from an environment
4     # variable
5     api_key='YOUR_OPENAI_API_KEY',
6 )
```

## Negative Words File:

Do not move or remove `negative_words.txt`.

## Deployment

Follow these steps to get the application running on your local machine.

## Virtual Environment:

```
1 conda create -n ChatBot python=3.8
2 conda activate ChatBot
```

## Install Required Packages:

Use `pip` to install the packages listed in your `requirements.txt` file.

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

## Run the Flask Application:

With your virtual environment activated and dependencies installed, you can now start the Flask development server.

```
1 python app.py
```

The server will start, and you should see output indicating it is running in debug mode and listening for connections, typically on `http://127.0.0.1:5000`.

```
Loading negative words from negative_words.txt...
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
Loading negative words from negative_words.txt...
* Debugger is active!
* Debugger PIN: 117-893-195
```

