

# **Emotion-Based AI Chatbot**

A project Report

Submitted by:

Prabin Shrestha

# 1. Introduction

This project introduces an AI chatbot that can recognize emotions from what users say and respond in a way that's aware of those emotions. The chatbot is built using a modular pipeline structure, which is managed by LangGraph , a tool that uses graphs to connect different parts of the system together. The interface for interacting with the chatbot is created with Streamlit, making it simple to test and use.

## 2. Objective

The goal of this project is to show how natural language processing and emotion analysis can be used in a chatbot system. By understanding the user's emotional state, the chatbot can offer more caring and realistic responses.

## 3. Architecture Overview

The chatbot is made up of several modular nodes that work together in a LangGraph pipeline. Each node handles a particular task, which makes it simpler to check for problems, run tests, and grow the system as needed.

### Project Structure:

```
Ai_proj/
├── app.py          # Streamlit UI
├── chat_history.json  # Memory stored here
├── requirements.txt  # Dependencies
└── langgraph_pipeline/
    ├── emotion_node.py  # Detects emotion from input
    ├── memory_node.py   # Adds memory context
    ├── pci_node.py      # Injects personality/context
    ├── action_node.py   # Handles final response
    └── graph_builder.py  # Constructs LangGraph pipeline
```

## 4. Pipeline Node Details

- Emotion Node: Uses the `j-hartmann/emotion-english-distilroberta-base` model to determine the emotional tone.
- Memory Node: Keeps track of the conversation using a permanent JSON storage system.
- PCI Node: Adds extra personality or context-based rules.
- Action Node: Takes previous information and uses it to create the final reply.

## 5. Technologies Used

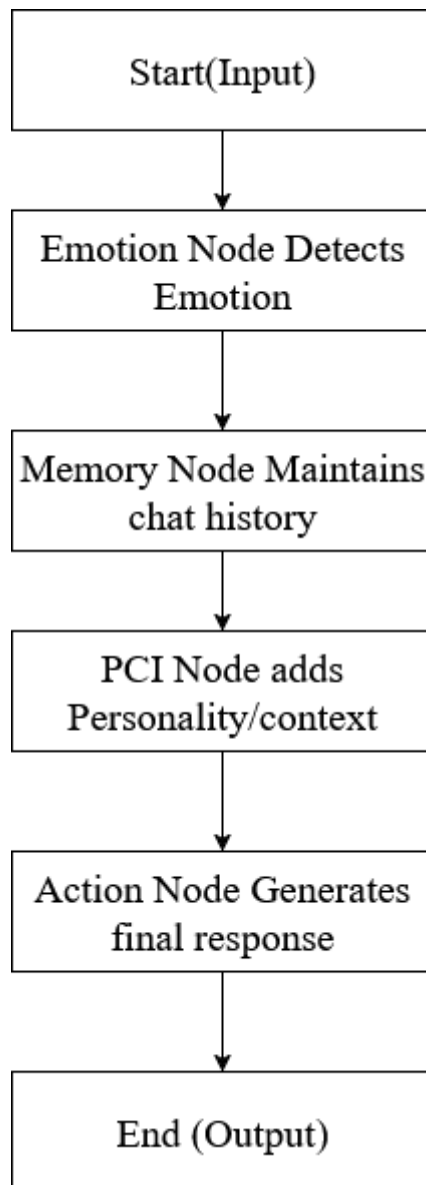
- Python
- Streamlit
- Hugging Face Transformers
- LangGraph
- JSON

## 6. How to Run the Application

1. Install the required packages: ``pip install -r requirements.txt``
2. Run the app: ``streamlit run app.py``
3. Chat with the bot via the browser interface.

## 7. Architecture Diagram:

The diagram below visualizes the LangGraph-based pipeline:



## 8. Conclusion

This project gave me real-world experience in merging NLP with modular design to create chatbots that behave more like humans. It highlighted how important it is for AI to understand emotions during conversations and showed how easy tools like memory and personality settings can greatly enhance the user experience.