

AI Mind Reset Bot Documentation

Student Name: Askal Thapa

Student RollNo: 2315434

Date: January 31, 2026

Link to git project repo: <https://github.com/askal-thapa/chat-bot>

1 Introduction

The **AI Mind Reset Bot** is an intelligent and empathetic chatbot designed to support users in managing stress, anxiety, and emotional well-being. Unlike standard chatbots, it uses intent classification to distinguish between casual small talk, therapeutic needs, and off-topic queries.

A key feature of the project is a dynamic *Mood Engine*, which updates the application's user interface theme in real-time based on the user's detected emotional state. This creates a more immersive and emotionally responsive experience.

2 Background

Mental health support tools have increasingly integrated conversational AI to provide accessible and low-friction assistance. Modern large language models (LLMs) enable natural and supportive conversations, but safe and relevant interaction requires structured control mechanisms such as intent classification and response routing.

This project was inspired by the need for an emotionally-aware assistant that can provide grounding techniques, calming guidance, and supportive dialogue while remaining user-friendly and visually engaging.

3 Implementation Details for Interactive Application

3.1 System Architecture

The system follows a decoupled architecture:

- **Frontend:** Next.js application responsible for UI and interaction
- **Backend:** FastAPI service responsible for API and orchestration
- **Agent Logic:** LangGraph stateful agent graph for intent routing
- **LLM:** Google Gemini used for response generation and mood detection

3.2 Conversation Flow

At a high level, the system works as follows:

1. The user submits a message through the browser interface.
2. The Next.js frontend sends the message to the backend via a REST API.
3. The FastAPI backend invokes the LangGraph agent.
4. The agent classifies the input into one of the following intents:
 - `small_talk`
 - `therapy`
 - `off_topic`
5. Based on the intent, the agent routes the request to the correct handler node.
6. Google Gemini generates a structured response including a detected mood.
7. The backend returns a JSON response to the frontend.
8. The frontend updates the UI theme based on the mood.

3.3 Backend Stack

The backend is implemented using:

- FastAPI (API framework)
- LangGraph (stateful orchestration)
- Google Gemini API via `langchain-google-genai`
- Pydantic (schema validation)

3.4 Structured Output Format

Instead of returning only text, the bot returns a structured object to keep the UI consistent and enable mood-based theming.

Listing 1: Structured Bot Response Model

```
1 class BotResponse(BaseModel):
2     response: str
3     mood: str
4     suggested_activity: str
5     activity_type: str
```

3.5 Mood Engine: Theme Changes Based on Mood and Intent

A unique part of this project is the **Mood Engine**, which automatically updates the user interface theme based on the detected emotional state of the user. Instead of keeping a static theme, the application changes colors and background gradients dynamically to match the user's mood.

The mood is detected by the backend model during response generation. Along with the reply text, the backend returns a **mood** label such as **Neutral**, **Sad**, or **Happy**. The frontend then maps this mood label to a predefined theme palette using CSS variables.

In addition, intent classification improves how the theme is used:

- For **therapy** intent, calming themes are preferred to reduce stress and support relaxation.
- For **small_talk** intent, brighter themes are used to make the experience more friendly and engaging.
- For **off_topic** intent, the theme remains neutral to keep the conversation stable.

This feature makes the application feel more interactive and emotionally supportive compared to a standard chatbot.

4 Frontend Mood Theming

The frontend includes a **Mood Provider** that maps the backend mood output into a theme palette. CSS variables are updated dynamically, allowing Tailwind classes to automatically reflect new colors with smooth transitions.

Example mood-to-theme mapping:

- **Neutral:** clean and balanced sky-blue theme
- **Sad:** soft cool tones for calmness and comfort
- **Happy:** warm bright colors for positivity

5 Details of Additional Experimentation

Additional development and experimentation performed during the project includes:

- Testing different prompt styles for intent classification accuracy.
- Improving the consistency of structured response generation.
- Experimenting with mood detection labels for better UI mapping.
- Reviewing how off-topic redirection can be done gently without breaking user trust.

6 Critical Reflection

This project successfully demonstrates how a conversational AI system can be made more supportive through structured intent routing and mood-based UI adaptation. The strongest part of the system is its ability to adapt both the conversation and the interface to the emotional tone of the user.

However, limitations include the dependency on LLM classification accuracy and the need for careful safety boundaries in mental-health related responses. If more time were available, improvements would include streaming responses, voice support, and persistent sessions stored in a database.

Ethical considerations were important throughout development. The system should be used as a supportive tool and not as a replacement for professional mental health services, particularly in crisis scenarios.

7 LLM Disclaimer

Large language models were used to support this project by generating responses, assisting with structured output formatting, and enabling intent classification. Any generated content was reviewed and adapted to fit the intended empathetic and safe tone of the chatbot.

8 Bibliography

- Association for Computing Machinery (2018). *ACM Code of Ethics and Professional Conduct*. Available at: <https://www.acm.org/code-of-ethics>

A Appendix

A.1 Theme Mode Screenshots

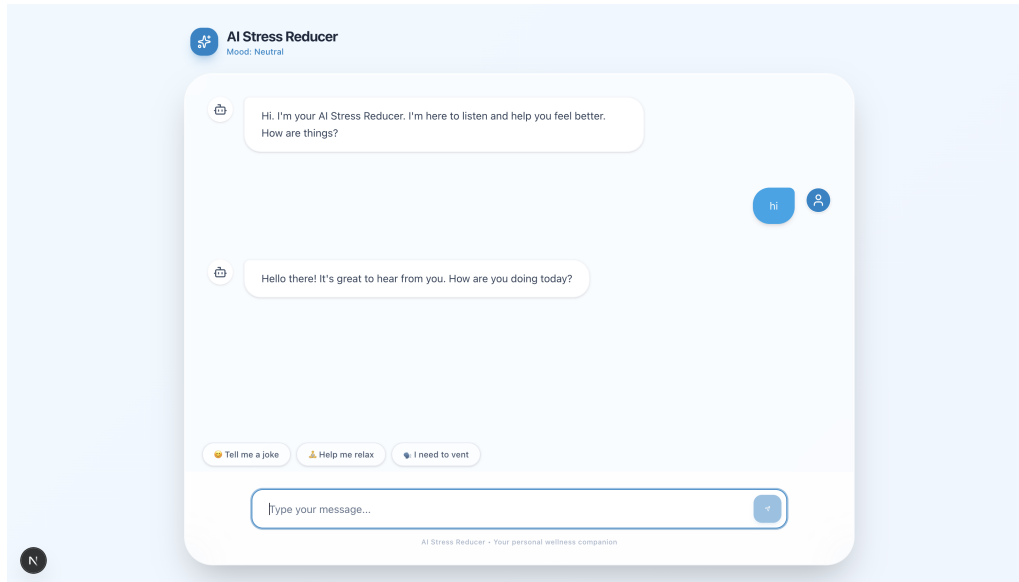


Figure 1: Neutral Mode: Default theme when the user is calm or normal.

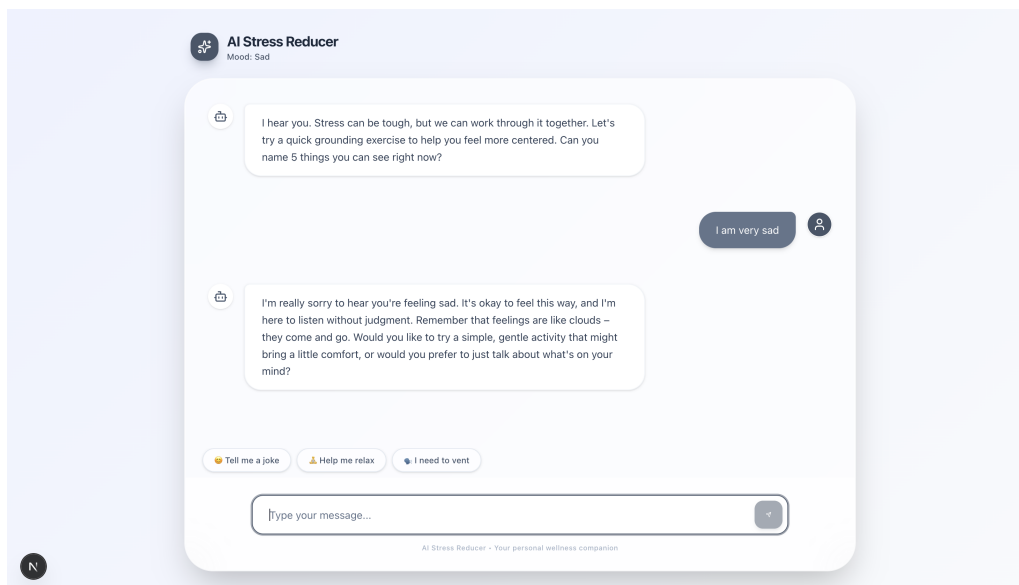


Figure 2: Sad Mode: Softer theme applied when the user mood is detected as sad.

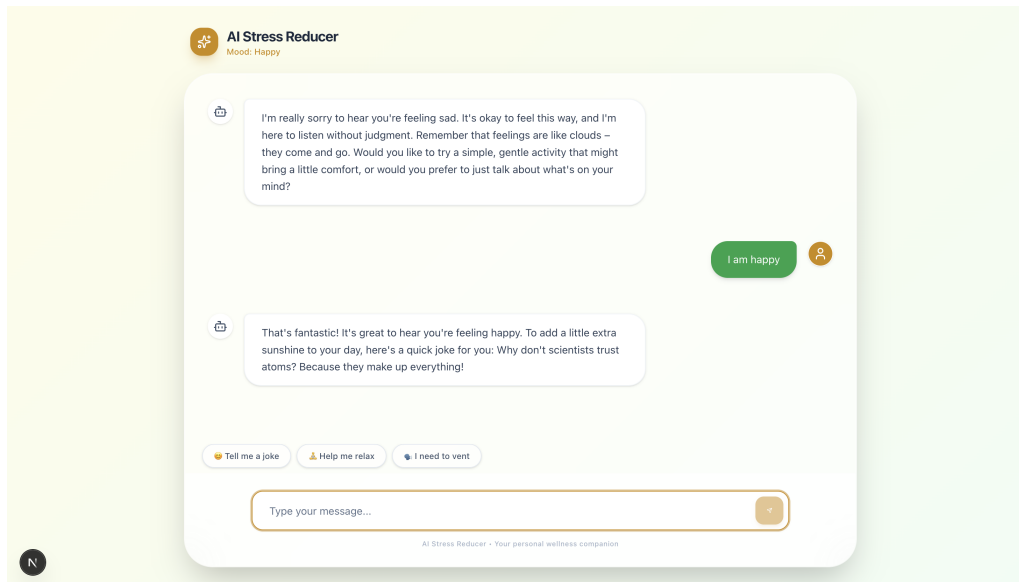


Figure 3: Happy Mode: Bright theme applied when the user mood is detected as happy.