# Compassion AI: A Guide for Your Mental Health

### 1. System Design and Implementation

**AgenticAI** is a lightweight web-based chatbot system aimed at providing mental health support through empathetic and kind conversations. It uses:

- **Flask** as the backend web server
- **Flask-CORS** to enable communication between frontend and backend
- **Anthropic's Claude LLM** as the AI engine
- **HTML** frontend (index.html, not yet analyzed)

**Backend Flow:**

- A Flask server is set up to handle POST requests to the /chat endpoint.
- The server reads the user's message from the incoming request.
- It sends this message to the **Claude-3 Haiku** model using the anthropic API.
- A predefined **system prompt** guides the model to act as "Compassion AI" — a supportive, non-judgmental AI assistant focused on mental health.
- The model's response is extracted and returned to the frontend as JSON.

### 2. Use of LLMs and Other Tools

**LLM: Claude-3 Haiku by Anthropic**

- Selected for its lightweight nature and responsive capability.

- Guided by a **system prompt** focused on emotional intelligence and non-judgmental tone.
- API interaction is handled using the official anthropic Python client.

**Other Tools:**

- **Flask** – Micro web framework for building the backend service.
- **dotenv** – For managing API keys securely via .env files.
- **Flask-CORS** – To avoid cross-origin issues between frontend and backend.

### 3. Challenges Faced

While the project is functionally sound, several technical and structural challenges are observed:

#### a. Hardcoded API Key

- The API key is hardcoded: client = anthropic.Anthropic(api_key="API KEY").
  - This can lead to accidental exposure.
  - Use only the environment variable os.getenv("CLAUDE_API_KEY").

#### b. Lack of Error Handling Detail

- The error message returned to the user is generic: "Sorry, something went wrong."
  - Adding more descriptive logs or error codes could aid debugging.

#### c. No Rate Limiting or Security Measures

- No protection against spamming or malicious use of the /chat endpoint.

#### d. Limited Frontend-Backend Feedback

- The backend doesn't validate if a message is empty or too long.

## 4. Future Work & Recommendations

### Documentation

- Update the README.md to include setup steps, API usage, and deployment instructions.

### Feature Expansion

- Add chat history, user sessions, or optional identity masking.
- Include feedback/rating system for AI responses.

### Security Improvements

- Implement proper authentication or API throttling.