Omdena-MindfulChat – Mental Health Chatbot

(Team 1 Contribution Report)

Overview

This report summarizes the final contributions by Team 1 toward the development of the MindfulChat Agentic Mental Health Chatbot under the Omdena AI Core Challenge. The chatbot offers intelligent, personalized, and accessible mental health support using Retrieval-Augmented Generation (RAG), CrewAI agents, and voice-enabled interaction. A modern architecture using FastAPI and React enables real-time assessment, emotional guidance, and adaptive conversation.

Key Features

- Modern Frontend: Migrated from Streamlit to a React-based UI with assessment flow and chat interface.
- FastAPI Backend: Modular API endpoints for chat, speech, memory logging, and assessments.
- CrewAI Integration: Multi-agent architecture with EmotionDetector, SuggestionAgent, SafetyAgent, RAGRetriever, and RAGReader.
- LLM Access: Gemini 2.0 Flash model using API key-based access, avoiding Google Cloud dependency.
- RAG Module: FAISS vector search over tips embedded with HuggingFace all-MiniLM-L6-v2.
- Voice Features:
- • STT: Whisper for speech-to-text conversion.
- • TTS: pyttsx3 for offline text-to-speech response.
- Assessment Pipeline: Dynamic intake and scoring for Stress, Anxiety, Depression, and FOMO.
- Memory Logging: CSV-based logs for user history, emotion, and scores.

Pipeline & Architecture

1. User fills profile and selects mental health concerns via React forms.
2. Assessment Routing: Conditional navigation to selected issues' questionnaires.
3. CrewAI Chat Activation:
   • Emotion Detection → RAG Retrieval → Suggestion Agent → Safety Agent
4. FastAPI Routes:
   • /chat: CrewAI pipeline
   • /transcribe: Whisper STT
   • /speak: pyttsx3 TTS
   • /retrieve: Vector-based tip lookup

Agent Design and Justification

The system follows a 5-agent setup using CrewAI, designed for modularity and domain separation:

• EmotionDetector – Analyzes tone and emotion from user input.
• SuggestionAgent – Delivers personalized self-care and coping advice.
• SafetyAgent – Identifies crisis or red flags, optionally alerting caregivers.
• RAGRetriever – Uses FAISS to fetch tips from stored documents.
• RAGReader – Summarizes tips into coherent and user-friendly language.

Technology Stack

- Frontend: JavaScript-(React, TailwindCSS)
- Backend: FastAPI (Python)
- LLM: Gemini 2.0 Flash via API key
- Embeddings: HuggingFace (MiniLM-L6-v2)
- Vector Store: FAISS
- Voice Modules: Whisper (STT), pyttsx3 (TTS)
- Agent Framework: CrewAI with YAML configuration
- Memory: Local CSV logging

Real-World Impact

• Enables low-barrier mental health access without clinical diagnosis.
• Supports users through voice, visual, and text interaction.
• Modular design allows extensibility into business use cases.
• Fully deployable.