DAYANANDA SAGAR COLLEGE OF ENGINEERING

(An Autonomous Institute affiliated to VTU, Belagavi, Approved by AICTE & ISO 9001:2008 Certified) Accredited by National Assessment & Accreditation Council (NAAC) with 'A' grade, Shavige Malleshwara Hills,Kumaraswamy Layout, Bengaluru-560078.

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

1. Introduction

Mental health is an increasingly critical issue in today's fast-paced, high-stress world. Unfortunately, access to mental health support is often limited due to financial, societal, or logistical barriers. To bridge this gap, we developed an AI-Powered Mental Health Assistant MVP that provides users with an initial assessment of their emotional well-being, personalized self-care suggestions, and access to professional mental health resources—all through an anonymous, chat-based interface.

2. Problem Statement

Millions of people experience mental health challenges but hesitate to seek help due to stigma, limited resources, or lack of access to professionals. There is a pressing need for a non-judgmental, always-available digital solution that can offer preliminary support, detect early signs of emotional distress, and guide individuals toward improved mental well-being or professional care when needed.

3. Objective

The goal of this MVP is to:

- Conduct basic mental health assessments through natural language conversations.
- Offer sentiment analysis to understand the user's emotional state.
- Provide personalized self-care recommendations.
- Maintain strict privacy and anonymity for users.
- Refer users to verified professional mental health resources when necessary.

Team Assignment Table

Module	Assigned Team Member(s)	Associated Files / Responsibilities
Client Development	Chaitra	App.jsx (Client-side application)
	Jayshree	Auth.jsx (Authentication module)
Server Development	Namratha	Server.js (Backend server management)
	Chandu	Server.js (Backend server management)
Encryption	Nouman	encryption.py (Data encryption mechanisms)
	Rahul	Support in encryption tasks
Database and History Management	Rahul	Database integration and user interaction history
Chatbot Development	Sneha	General chatbot functionalities
	Prajwal	chatbot.py (Core chatbot logic)
	Samarth	chatbot.py (Supporting chatbot functionalities)
User Interface (UI)	Shrihari	UI/UX development
	Saketh	Dataset Collection
	Uday	Streamlit UI (read main.py)
Sentiment Analysis	Sneha	sentiment.py (Sentiment evaluation)
	Siddarth	sentiment.py (Sentiment evaluation)
	Shreyansh	sentiment.py (Sentiment evaluation)
Recommendation System	Abhinand	recommendation_system.py (Recommendation engine)
	Abhishek	recommendation_system.py (Recommendation engine)
	Adithya	vid.py, Magic Loop, Faker
Documentation	Deekshita	Project documentation and reporting

5.Tech Stack

Chatbot

- Streamlit: For building the chatbot's user interface.
- LangChain Community Models: For leveraging pre-trained models.
- OpenAI/Groq: For fine-tuning the chatbot.
- NLTK: For tokenising vector embeddings.
- PyMongo: For interacting with MongoDB.
- Pycryptodome: For encryption and secure data handling.
- Pandas & NumPy: For data manipulation and analysis.
- PyJWT: For generating and verifying JSON Web Tokens.
- Scikit-learn: For machine learning model metrics.
- Transformers: For sentiment analysis

Web Application

-Frontend

- React 18: For building the user interface.
- React Router: For client-side routing.
- Framer Motion: For animations.
- Lucide Icons: For UI icons.
- Tailwind CSS: For styling.
- Vite: For fast development and build tooling

-Backend

- Express.js: For building the REST API.
- MongoDB: For database storage.
- Mongoose: For MongoDB object modeling.
- JWT: For authentication.
- bcryptjs: For password hashing.
- Cors: For handling cross-origin requests.
- dotenv: For managing environment variables.

DevOps

- Docker: For containerizing the application.
- Docker Compose: For orchestrating multi-container applications.
- Nodemon: For automatic server restarts during development.

6. Results

Successfully deployed a chat-based assistant that:

- Engages users empathetically.
- Detects sentiment using NLP and emotion analysis.
- Provides self-care tips and professional resource links.

7. Challenges Faced

- Emotional Nuance Detection: Some user messages were difficult to classify due to sarcasm or ambiguous language.
- API Rate Limits: IBM Watson and WHO APIs had limits that required efficient batching and fallback responses.
- Privacy Concerns: Implementing true anonymity while storing useful interaction data was tricky.
- Tone Misclassification: Occasionally, strong negative words skewed sentiment detection inaccurately.

8.Learnings

- Learned how to integrate multiple APIs into a cohesive user experience.
- Gained deeper insights into ethical design considerations in mental health applications.
- Discovered the value of empathetic conversational design in AI interactions.

9. Future Scope

- Add voice and video input for richer emotional context using tone and facial analysis.
- Develop a learning loop to personalize advice over time based on user history (with opt-in).
- Implement daily mood tracking and weekly mental health check-ins.
- Build a mobile app version with push notifications for self-care nudges.
- Offer local mental health professional directories based on user location (optional).

10. Conclusion

This MVP demonstrates the potential of AI to play a supportive role in mental health care. While it does not replace professional help, it offers a scalable, accessible first line of support for those who need someone to talk to or don't know where to start. The project proves that empathetic AI, combined with real-world resources, can make a meaningful impact on mental wellness.

11. References

- Google Dialogflow Documentation
- IBM Watson Tone Analyzer API Docs
- WHO Mental Health API Portal
- Twilio Developer Documentation
- Journal of Medical Internet Research – Chatbots in Mental Health
- Stanford ML Course (Sentiment Analysis Module)