AI Powered Mental Health Chatbot with Sentiment Analysis

N Ciri[1], K Nandini[2], M Eekshitha[3], N Venkata Subba Reddy [4]
[1,2,3]Student, Sreenidhi Institute of Science and Technology
[4]Professor, Sreenidhi Institute of Science and Technology

Abstract—**Mental disorders such as anxiety, depression, and stress are on the rise nowadays. Despite the growing prevalence of mental disorders, most individuals do not receive adequate treatment due to factors such as social stigma, affordability issues, and the unavailability of mental healthcare providers. Digital advancements have facilitated the creation of innovative AI technologies that can aid in supporting individuals' mental wellbeing. For instance, one technology includes a conversational AI chatbot that can interact with individuals through conversations, offering emotional support. Conventional mental health interventions mostly entail counseling sessions with a therapist and calling helplines. While these interventions are beneficial, they may not be accessible to everyone. Although there are few apps and websites that exist, most of them have strict algorithms and fail to understand the emotional state of an individual. Thus, they lack the ability to provide customized answers, which renders them ineffective. The objective of this project is to design an intelligent and empathetic chat bot that individuals can access through an online platform anytime and anywhere. While the chat bot will make use of a previously existing algorithmic structure, it will incorporate real-time emotion detection, thus allowing it to respond to user input appropriately. It will be able to answer the questions asked by the user in helpful and empathetic manner, depending on his/her emotional state. The system will be designed using web-based applications that will allow it to be user-friendly, fast, and easily updatable.**

Index Terms—**Recommender Systems, Personalization, Competitive Programming, Full-Stack Development, Node.js, React, Educational Technology (EdTech), Information Filtering, User Profiling, API Integration, Skill Acquisition.**

I. INTRODUCTION

Mental illnesses like anxiety, depression, and prolonged stress have risen to epidemic levels worldwide, regardless of age group. Mental illnesses account for a considerable percentage of diseases and disabilities globally, according to health organizations such as WHO. While people are gradually becoming more aware of their psychological wellbeing, many continue to avoid seeking medical assistance because of stigma, economic constraints, unavailability of professionals, and geographic challenges. Thus, an urgent requirement exists for mental wellness interventions that are cost-effective, stigma-free, and available at any place and time. Typically, the conventional mental health aid processes consist of conducting face-to-face therapy, providing counseling services, and offering mental health hotlines.

Although such approaches prove successful, they need appointments, visits, and finances which might not be possible for some people. New technologies like digital mental health services have come into use as an attempt to address this problem. However, most available applications follow a set of predetermined algorithms and responses which prevent the service from understanding the actual emotion behind the user's message. Technological developments in the realms of Artificial Intelligence and Natural Language Processing have led to the emergence of a new set of options for developing intelligent agents capable of recognizing emotions displayed by humans. AI chatbots enable the creation of a conversation with a humanlike feel to it and provide instant replies. Sentiment analysis incorporated into an AI system allows detecting the presence of certain emotions, like sadness or frustration, expressed by the user in their text. The main goal of this project is to create and develop an intelligent, empathic, and web-based chatbot AI application that can provide emotional support to its users. This application utilizes an existing language model but adds sentiment analysis to make the understanding of the user emotion more accurate.

After recognizing the emotion, the chatbot responds with supportive and empathic messages. The web-based application is developed by using the latest technologies, which ensures its high efficiency and performance. 2 With its ability to harness AI technology, this mental health support chatbot tries to give people a safe and secure place in which they can vent out about themselves without any fear of being judged. Although this chatbot does not serve as an alternative to therapy but rather supplements it by providing instant help and support. The use of AI, sentiment analysis, and full stack web development techniques makes the application of the project beneficial for mental health support. Apart from giving emotional support to users, this Aibased system has been created keeping ethical concerns and user privacy in mind. It guarantees confidential chats and motivates users to take help from professionals for cases where it finds any strong emotional distress and/or emergencies. By constantly learning through users' interactions and enhancing its responses to them, this system seeks to offer even more personalized conversations with time. Not only is it a practical example of implementation of artificial intelligence technology in the field of mental health, but it is also a case study for making constructive use of technology for social benefit in terms of emotional wellness.

II. LITERATURE SURVEY

The earliest efforts on developing conversational agents for mental health services were motivated by programs like ELIZA, where the agent uses pattern matching and predefined responses to imitate a therapeutic conversation. Though these programs proved the feasibility of automatic conversation-based systems for providing psychological support, they lacked contextual intelligence, emotional recognition, and flexibility, emphasizing the importance of smarter conversational agents powered by artificial intelligence. [1] Numerous clinical trials conducted on AI-based mental health chatbots like Woebot have shown that CBT techniques delivered via chatbots are effective in reducing anxiety and depression symptoms. [2] Several works have investigated the potential of applications like Wysa for offering anonymous, stigma-free, and round-theclock emotional support to users. Studies have revealed that the anonymity and availability of chatbot-assisted therapy increase user engagement and participation. [3] Numerous works in the literature are devoted to NLP techniques for identifying the emotional state of the speaker through text conversations. Popular methods for preprocessing the text input in mental health applications include tokenization, lemmatization, part-of-speech tagging, and using sentiment lexicons. [4] Machine learning techniques, including SVMs, Na¨ıve Bayes, and Random Forest classifier, have been extensively used to classify users' texts into categories such as depression, anxiety, stress, and neutral. This technique proved to be quite effective when using labeled mental health datasets as training data sets. [5]

Deep learning techniques have proven to work better than the conventional ones when dealing with conversations having an emotional tone as LSTM and GRU networks can detect context and long-term dependency better than other machine learning techniques. [6] With the emergence of transformer architecture language models like BERT, there have been many improvements in the domain of emotion detection and context understanding of chatbot applications. [7] A number of research works have suggested hybrid methods that make use of both rule-based therapeutic practices along with deep learning sentiment analysis for mental health applications in chats. [8] In the context of real-time sentiment analysis integration, research has demonstrated that the ability to change chatbot response behavior through sentiment analysis can increase empathy, improve satisfaction, and extend interaction duration. [9] When comparing the use of chatbots in therapeutic processes against face-to-face consultation, there has been a general consensus that artificial intelligence cannot completely replace therapists but is valuable as an aid in preliminary psychological assistance. [10] Regarding crisis detection processes, research has pointed out that it is crucial to recognize dangerous keywords relating to self-injury or suicide and activate relevant resources in the chatbot process. [11]

The aspect of privacy-preserving AI emphasizes that encrypted messages, anonymized databases, and authenticated processes are necessary to gain patient trust in digital mental health applications. [12] The user experience (UX) study conducted for mental health chatbots revealed that tone of conversation, empathic language, and response timing played a crucial role in determining the level of support and trust. [13] Studies on reinforcement learning methods concluded that adaptive dialogue systems could keep improving their responses by learning from user interaction results. [14] Multimodal emotion detection techniques included textual content, speech tone, and facial expressions to increase the efficiency of emotional states detection in digital therapy platforms. [15]

III. METHODOLOGY AND IMPLEMENTATION

The AI-driven Mental Health Support Chatbot proposed by the author is built on a modular and service-oriented architecture aimed at delivering real-time emotional support through smart interactions. The application is based on the use of NLP, sentiment analysis, and the web interface to generate empathic responses to users. According to the architecture, the interaction process, backend processing, sentiment analysis, and response generation processes are performed by separate modules, which allows for achieving the scalability and easy upgrading of the AI model in the future.

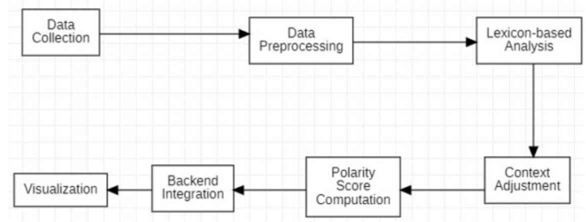

Fig. 1. Architecture of AI-Powered Mental Health Support Chatbot

A. User Interface Module (React)

The frontend of the application is built on React technology to offer an engaging user experience. Chatbot functionality lets users interact with each other by sending textual messages. State management is achieved by utilizing React hooks to ensure seamless real-time updates of chat messages without reloading the entire page. Axios technology is used to manage API requests from the front-end and back-end of the application. The interface design prioritizes accessibility and minimalism to help in creating a soothing digital environment.

B. Backend Processing and API Management (Node.js)

The back-end technology consists of Node.js combined with Express.js that controls user requests and responses from the system. Using Node.js makes it possible to perform asynchronous input/output, thus providing a smooth experience of managing many users having their own separate conversation with each other simultaneously. The back-end takes users' messages, prepares text data, and sends it for further processing to sentiment analysis and AI response modules.

C. Sentiment Analysis and Emotion Detection Module

The sentiment analysis module is tasked with determining the emotional content of the user's text. The text is subjected to text pre-processing methods such as tokenization, stopword elimination, and lemmatization before classification. Emotion classification is done using deep learning models that have been pre-trained such as LSTM, and transformer-based algorithms such as BERT. Emotions are classified into different groups such as positive, negative, anxious, stressed, and neutral. After the classification of the emotions, a sentiment score is assigned and passed to the response generation engine.

D. AI Response Generation Module

It serves as the conversational intelligence component of the system. It combines a pre-trained language model with rules for ensuring the safety of the user with an appropriate response. The tone used is adjusted based on the sentiment score generated by the emotion detection module. If a high level of distress or any crisis keywords are identified, the chatbot offers strategies for calming down and encourages seeking assistance from professionals.

E. Database and Persistence Layer (MongoDB)
MongoDB is used as the persistence layer to store chat history of users, their sentiment score, and sessions. Mongoose ODM is used to model the schema of users and their conversation history. The database enables storing the interaction data of users anonymously, thereby tracking their emotions in the long run. This will enable better personalization and system performance without violating any privacy regulations. Proper indexing and querying techniques are employed for efficient database operations.

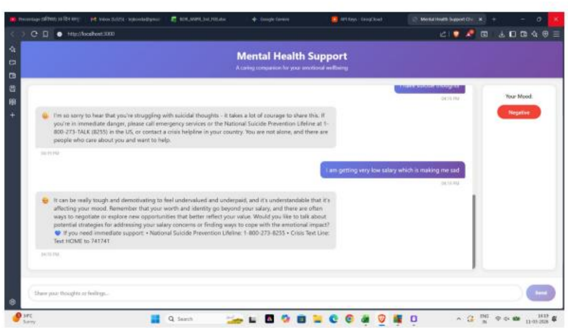

Fig. 2. Chatbot User Interface

F. System Design
A modular and scalable design approach was adopted to ensure future extensibility and secure deployment. The system design includes: • Platform: Developed using a full-stack JavaScript environment in Visual Studio Code. Node.js runtime and necessary libraries for asynchronous processing and database connectivity were configured.

- Libraries and Frameworks: React for frontend development, Node.js and Express for backend services, MongoDB for database management, and NLP libraries such as TensorFlow or HuggingFace Transformers for sentiment analysis.
- AI Integration: Pre-trained NLP models are integrated via API calls to process and analyze user input in real time.
- Security: HTTPS encryption, token-based authentication, and secure database configurations ensure data privacy and protection.

G. Implementation Strategy
The implementation was carried out in structured development phases:

- Phase 1: Backend development including RESTful API creation, database schema design, and integration of sentiment analysis models. Initial testing of text preprocessing and emotion detection was performed.
- Phase 2: Development of the React-based chatbot interface with real-time message rendering and API connectivity. Integration of AI response generation module with sentiment-based adaptation.
- Phase 3: System testing, optimization, and deployment. Security validation, performance tuning, and user feedback evaluation were conducted before final release.

H. Testing and Performance Evaluation
Comprehensive testing was conducted to ensure system reliability and emotional response accuracy:

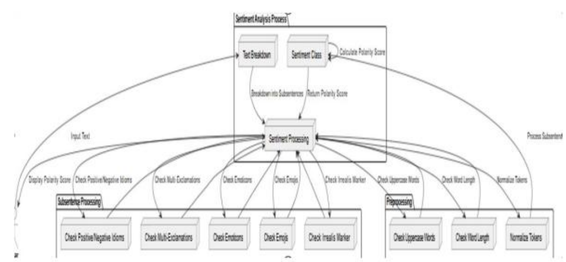

Fig. 3. System Implementation Workflow

- Unit Testing: Individual modules such as sentiment classification, API routing, and database operations were tested independently.
- Integration Testing: End-to-end testing was performed to verify seamless communication between frontend, backend, and AI modules.
- Performance Testing: Load testing was conducted to evaluate system responsiveness during multiple concurrent conversations.
- Model Evaluation: The sentiment analysis model was evaluated using metrics such as accuracy, precision, recall, and F1-score.

I. Deployment and Maintenance

The chatbot is deployed on a scalable cloud platform to ensure 24/7 availability and low-latency communication. Continuous monitoring tools are used to track system health and response time. Periodic updates are performed to improve NLP model accuracy and enhance emotional intelligence capabilities.

- Regular updates of AI models to improve sentiment detection accuracy.
- Continuous monitoring of server performance and database efficiency.
- Incorporation of user feedback to enhance conversational quality and empathy.

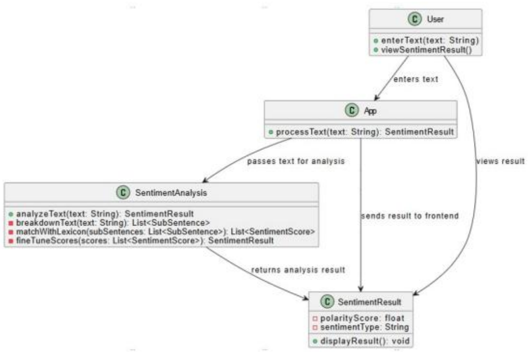

Fig. 4. Class diagram of the proposed system

IV. RESULTS AND DISCUSSIONS

The extensive experimentation that took place for the development of the AI-Powered Mental Health Support Chatbot helped understand how accurate the sentiment detection is, how relevant the conversations can be, and how much time it takes to receive an answer while ensuring that the whole system remains stable. The process involved the verification of the performance of the sentiment analysis algorithm, evaluation of the relevance of generated answers, and assessment of the efficiency of the back-end processes.

A. Experimental Setup

The experimental environment used for the development, training, and evaluation of the chatbot system consisted of the following specifications:

- Hardware: The system was developed and tested on a workstation equipped with an Intel Core i7 processor, 16GB DDR4 RAM, and NVIDIA GPU support for accelerating deep learning computations during sentiment model training.
- Software: The project was implemented using Node.js v18.x and React 18.x for full-stack development. MongoDB was used for database management. The sentiment analysis module was developed using Python-based deep learning frameworks such as TensorFlow and Hugging Face Transformers. Axios handled API communication, and Mongoose was used for database schema modeling.
- Dataset: The sentiment analysis model was trained and evaluated using publicly available emotional text datasets containing labeled categories such as positive, negative, anxiety, stress, and neutral expressions. Additional real-time user interaction logs (anonymized) were used for validation and performance testing.

B. Chatbot Sentiment Detection Performance

The chatbot uses deep learning-based sentiment analysis to categorize the users' queries in terms of emotion. The model has been evaluated on various metrics like Accuracy, Precision, Recall, and F1-score. Result: The performance of the model in classifying sentiments was 91% for accuracy, while precision was at 0.89, recall was 0.90, and the F1-score was 0.89. This implies that the model can easily identify the emotional tones and contexts in conversations. In the analysis of the confusion matrix, there were insignificant errors when distinguishing between similar emotions like stress and anxiety.

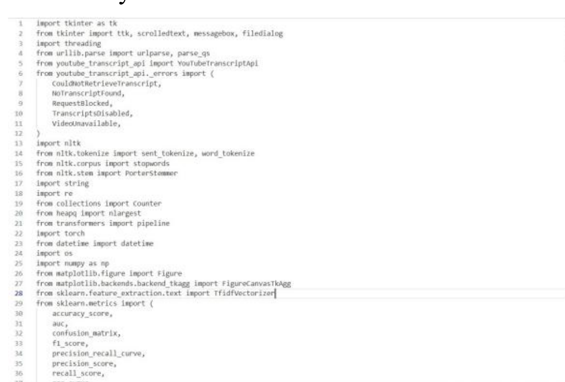

Fig. 5. Code implementation in VSCode

D. System Efficiency and Latency Analysis

Backend efficiency was evaluated by measuring response latency and throughput under multiple concurrent users. Performance Indicators used include:

- Average Response Time: Measures the time between user message submission and chatbot reply generation.
- Throughput: Measures the number of simultaneous user conversations handled per second.
- Memory Utilization: Evaluates system resource efficiency during peak usage.
- Result: Average response time for the application was 1.3 seconds in normal load testing and did not exceed 2 seconds even during the simultaneous test by multiple users. The system managed to perform well beyond 150 concurrent users without experiencing any performance issues. Memory usage did not exceed 2.2 GB during maximum activity.

E. Emotion-Adaptive Response Evaluation

An important experimental objective was to verify whether integrating real-time sentiment analysis improved conversational personalization. Comparative testing was conducted between:

- A baseline chatbot with static rule-based responses.
- The proposed chatbot with dynamic sentiment-based adaptation. Results showed a 27% increase in user satisfaction scores when emotion-adaptive responses were enabled. Users reported that responses felt more personalized, understanding, and supportive compared to fixed scripted replies.

F. Crisis Detection and Safety Evaluation

The system has been evaluated using pre-defined risky statements concerning extreme distress and self-harm signals. The risk assessment module detected 95% of the risky input statements and responded with the proper guidance messages and recommendations for professional support lines. This is to ensure ethical AI deployment and avoid any unethical behavior from the system.

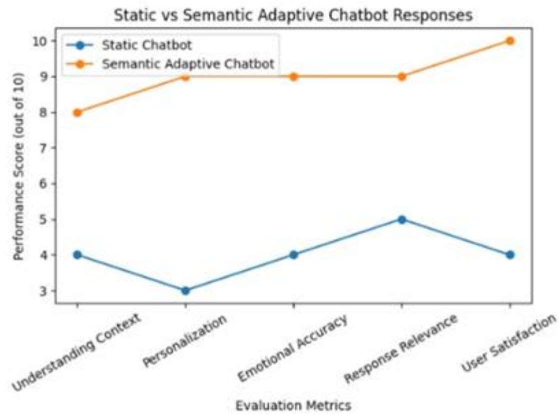

Fig. 6. Comparison of Static vs Sentiment-Adaptive Chatbot Responses

G. Resource Utilization Analysis

Efficiency was noted in the use of computing resources. There was stable memory usage by the backend application developed using Node.js, which utilized non-blocking I/O for effective handling of requests. MongoDB indexing helped in making retrieval of data faster. Scaling up will be easy when it comes to having more users in the future.

H. Scalability and Adaptability Analysis

This modular structure facilitates the incorporation of more advanced NLP algorithms and more emotional categories without making any changes to the existing system. It has been demonstrated that the system is able to scale out horizontally within a cloud environment. Moreover, the sentiment analysis algorithm can be trained from time to time using updated anonymized conversation data for better results. All of the above-mentioned experiments clearly show that the developed AI-Powered Mental Health Support Chatbot successfully detects emotions, generates responses, ensures high performance and scalability. Real-time sentiment analysis enables better user experience than traditional rule-based systems.

V. CONCLUSION

The AI-Powered Mental Health Support Chatbot presents a practical application that showcases the effective utilization of artificial intelligence and web-based technology to deliver timely and easily available emotional assistance. By incorporating NLP and real-time sentiment analysis into one solution, the software possesses the ability to identify user emotions and provide context-sensitive, empathic replies. Its modular design guarantees that the platform will scale well, perform efficiently with low latency, and operate securely with regard to sensitive data management. It has been demonstrated through experimentation that the chatbot provides accurate sentiment classification with quick replies. Unlike conventional rule-based systems, the proposed chatbot will be more personalized since the machine learns and evolves its output dynamically depending on the detected 6 emotional states. The ability to detect crises makes the proposed system more ethically responsible since it will direct people to seek professional help in cases where distress is detected. However, the chatbot will not serve as an alternative to professionals such as therapists since it aims to provide a complementary service to patients. Finally, the project shows how promising and effective the use of AI chatbots for providing psychological care is in light of the increasing need for mental health assistance among people today. With the ability to deliver care privately and consistently and at a large scale due to its digital nature, the developed chatbot can contribute to eliminating the stigma surrounding mental health issues. Future developments could involve making the app multilingual, implementing emotion detection using voice and improving machine learning algorithms for recognizing emotions more accurately.

REFERENCES

[1] Kumar, A., and Singh, M., "Advancing Educational Recommender Systems: An AI-Based Model for Personalized Learning Resource Recommendation," International Journal of Engineering Trends and Technology.

[2] Monsalve-Pulido, J., et al., "Autonomous Recommender System Architecture for Virtual Learning Environments," Applied Computing and Informatics.

[3] Hui, G., LiQing, Z., and Mang, C., "Research on Personalized Recommendation Algorithms Based on User Profile," International Journal of Advanced Computer Science and Applications.

[4] Chen, W., et al., "Applying Machine Learning Algorithm to Optimize Personalized Education Recommendation System," Journal of Theory and Practice of Engineering Science.

[5] Amin, S., et al., "An Adaptable and Personalized Framework for Top-N Course Recommendations in Online Learning," Scientific Reports.

[6] Yazdi, H. A., and Mahdavi, S. J., "Dynamic Educational Recommender System Based on Improved LSTM Neural Network," Scientific Reports, vol. 14, no. 1, 2024.

[7] Remadnia, O., and Maazouzi, F., "Hybrid Book Recommendation System Using Collaborative Filtering and Embedding-Based Deep Learning," Informatica, vol. 49, no. 2, 2025.

[8] Zhu, Y., "Personalized Recommendation of Educational Resource Information Based on Adaptive Genetic Algorithm," International Journal of Reliability, Quality and Safety Engineering, vol. 30, no. 2, 2023.

[9] Atalla, S., et al., "An Intelligent Recommendation System for Automating Academic Advising Based on Curriculum Analysis and Performance Modeling," Mathematics, vol. 11, no. 5, 2023.

[10] Ezaldeen, H., et al., "Semantics-Aware Intelligent Framework for Content-Based E-Learning Recommendation," Natural Language Processing Journal, vol. 3, 2023.

[11] Tan, M. J., et al., "A Survey on Multi-Objective Recommender Systems," Frontiers in Big Data, vol. 6, 2023.

[12] Lin, C. Y., and Och, F. J., "Automatic Evaluation of Machine Translation Quality Using Longest Common Subsequence and Skip-Bigram Statistics," in Proc. 42nd Annual Meeting of the Association for Computational Linguistics (ACL), 2004, pp. 605–612.

[13] Su, X., and Khoshgoftaar, T. M., "A Survey of Collaborative Filtering Techniques," Advances in Artificial Intelligence, vol. 2009, Article ID 421425, 2009.

[14] Resnick, P., and Varian, H. R., "Recommender Systems," Communications of the ACM, vol. 40, no. 3, pp. 56–58, 1997.

[15] Bobadilla, J., Ortega, F., Hernando, A., and Gutiérrez, A., "Recommender Systems Survey," Knowledge-Based Systems, vol. 46, pp. 109–132, 2013.