

---

**EARLY DETECTION AND GUIDANCE THROUGH MENTAL  
HEALTH CHATBOT**

---

**\*<sup>1</sup>Praveen V.D., <sup>2</sup>Dr. S. Saranya**

<sup>1</sup>Department of Artificial intelligence and machine learning, Dr. N.G.P Arts and Science  
College Coimbatore-641048, Tamil Nadu, India.

<sup>2</sup>Associate Professor and Head Department of Artificial intelligence and machine learning  
Dr. N.G.P Arts and Science College Coimbatore-641048, Tamil Nadu, India.

Article Received: 03 March 2026, Article Revised: 21 March 2026, Published on: 11 April 2026

**\*Corresponding Author: Praveen V.D.**

Department of Artificial intelligence and machine learning, Dr. N.G.P Arts and Science College Coimbatore-641048,  
Tamil Nadu, India.

DOI: <https://doi-doi.org/101555/ijarp.1500>

**ABSTRACT**

Mental health issues such as stress, anxiety, depression, and emotional distress have become increasingly common in today's fast-paced world. Factors such as academic pressure, work stress, and social isolation often contribute to these challenges. However, many people hesitate to seek professional help due to social stigma, lack of awareness, or limited access to mental health services. To address this gap, this paper introduces AURA AI, an intelligent mental health chatbot designed to offer early emotional support through conversational interaction.

The system utilizes Natural Language Processing (NLP) and machine learning techniques to understand user messages and detect emotional intent. A custom intent classification model based on TF-IDF vectorization and Logistic Regression is used to categorize user inputs into mental health-related intents such as stress, sadness, anxiety, or general conversation. The chatbot is developed using Python and Flask, while the user interface is built with HTML, CSS, and JavaScript to allow smooth, real-time communication with users.

In addition, the platform can integrate external AI services to generate empathetic responses and provide helpful mental wellness suggestions. By making emotional support more accessible and encouraging users to express their feelings, AURA AI aims to serve as an initial support system that can guide individuals toward better mental well-being and, when necessary, motivate them to seek professional help.

**KEYWORDS:** Artificial Intelligence, Mental Health Chatbot, Natural Language Processing, Machine Learning, Emotional Support, Conversational AI, Intent Classification, Web-based Chatbot System.

## 1. INTRODUCTION

Mental health problems such as stress, anxiety, and depression have become more common due to modern lifestyle pressures and social challenges. Many people avoid seeking professional help because of stigma, lack of awareness, or limited access to mental health services. In addition, traditional counseling often requires appointments and direct interaction with professionals, which may not always be easily available.

With the advancement of Artificial Intelligence (AI) and Natural Language Processing (NLP), intelligent chatbots can now interact with users and provide basic emotional support. This research introduces AURA AI, an AI-powered mental health chatbot that analyzes user messages to understand emotional intent. The system offers real-time conversational support through a web-based platform, helping improve access to mental health assistance and encouraging users to express their feelings openly.

## 2. Literature Review

Several studies have explored the use of AI to support mental health and emotional well-being. Earlier approaches mainly relied on traditional counseling and helpline services, which often faced challenges such as limited availability of professionals, high costs, and the social stigma associated with seeking help. Some early digital solutions introduced automated agents for therapy, but many of these systems lacked strong predictive abilities and struggled to handle longer, meaningful conversations.

More recent research has applied Natural Language Processing (NLP) and machine learning models, including Logistic Regression and Random Forest, with TF-IDF vectorization to identify emotional states from user text. However, many existing systems still do not effectively combine real-time interaction with context-aware conversations. AURA AI aims to address this gap by integrating a local machine learning model with a Google Gemini-based AI agent, enabling real-time emotional support and personalized guidance through a secure and interactive dashboard.

## 3. Proposed System Architecture

The proposed system is designed with three main components that work together to provide a smooth and reliable user experience.

**Data Acquisition and Validation Layer:**

This layer acts as the starting point of the system, where user messages are collected through the web interface. It includes a content moderation filter to prevent inappropriate inputs and performs basic validation to ensure that the data is safe and reliable before further processing.

**Data Processing and AI Engine Layer:**

This is the core part of the system. It uses a Natural Language Processing (NLP) pipeline to understand user messages. A machine learning model built with TF-IDF vectorization and Logistic Regression helps identify mental health-related intents. For more complex conversations, a Google Gemini-powered AI agent supports context-aware and multi-turn dialogue to provide better responses.

**Persistent Storage and Interface Layer:**

This layer manages user data and conversation records. The system uses a Flask backend with a MySQL database to store session details and past interactions. The user interface includes a modern dashboard that allows users to chat in real time, record their daily moods, and review their previous conversations.

**AURA AI - System Architecture & Data Flow Diagram**

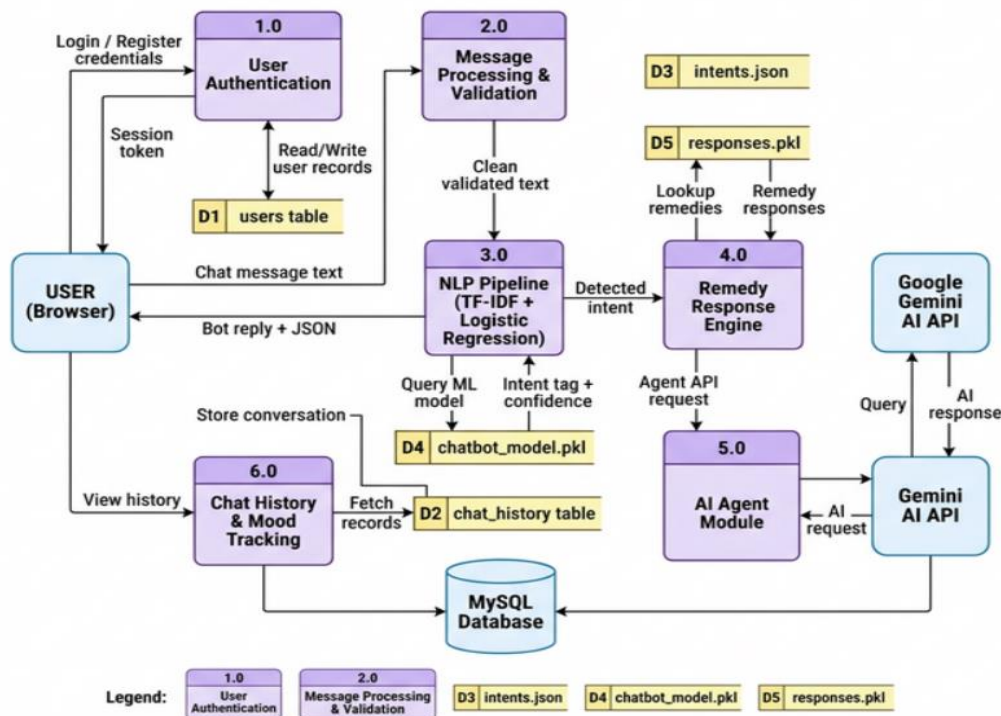


Figure 1: System Architecture.

## 4. METHODOLOGY

### Data Collection and Preparation

The system uses a structured dataset containing more than 250 conversational patterns across 17 mental health categories, including anxiety, depression, and stress. The dataset is stored in JSON format, making it easy to update and expand when new conversation patterns are added.

### Preprocessing and Feature Engineering:

User input text is first cleaned and converted to lowercase to remove inconsistencies. Then, a TF-IDF vectorizer converts the text into numerical features by analyzing both single words and two-word phrases, helping the system better understand emotional context.

### Model Training and Classification:

A Logistic Regression model built with a scikit-learn pipeline is trained using the processed data to identify the user's intent. The model is evaluated using an 80–20 train–test split to ensure reliable performance across all categories.

### Inference and Response Generation:

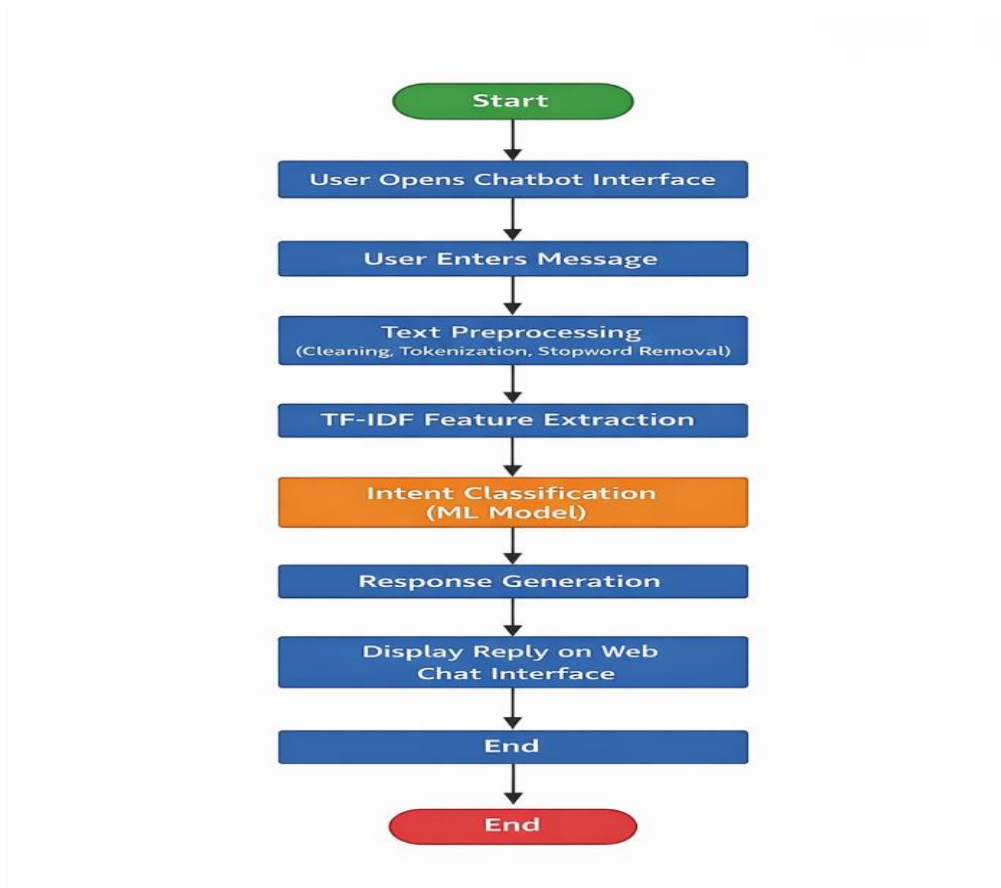
When a user sends a message, the Flask backend processes it through the trained model in real time. If the prediction confidence is above 25%, the chatbot provides an appropriate response related to the detected issue; otherwise, it returns a safe fallback reply.

### Persistent Interaction:

All user messages, detected intents, and chatbot responses are stored in a MySQL database. This helps maintain conversation history and allows the system to support features such as mood tracking over time.

### Methodology Flowchart:

The overall methodology is represented through a flowchart that illustrates the sequence from data collection and preprocessing to intent prediction, response generation, and data storage.



**Figure 2: Methodology Flowchart.**

#### 4.1 Software Design

The machine learning pipeline is built using Python and the scikit-learn library. It uses a TF-IDF vectorizer to convert user text into numerical features and a Logistic Regression classifier to identify different mental health-related intents from user messages.

The backend server is developed using the Flask framework in Python. It manages user authentication with bcrypt-hashed passwords, processes API requests, and securely stores user data in a MySQL relational database.

The frontend interface is designed using HTML, CSS, and JavaScript. It features a modern glassmorphism-style layout with a responsive chat interface and interactive features such as mood tracking, ensuring a smooth and user-friendly experience.

#### 4.2 Input Design

- **User Credentials:** Basic registration and login information such as username, email, and phone number.
- **Conversational Text:** Messages written by users in natural language to express their feelings, thoughts, or concerns.

- **Current Mood:** User-selected emotional states that indicate how they are feeling at a particular moment.
- **Stress Levels:** Scale-based inputs that help measure the user’s level of stress or anxiety. These inputs are sent from the web interface to the Flask backend, where they are processed using an NLP pipeline to analyze the text and identify possible mental health patterns.

### 4.3 Output Design

The dashboard offers several helpful features to support users during their interactions. It provides real-time responses to user messages, helping users feel heard and supported during conversations. The system also displays emotional insights by identifying the user’s possible mental state based on their input.

In addition, the platform offers personalized mental wellness suggestions and coping strategies to help users manage their emotions. Users can also access past conversation history and mood tracking data, allowing them to monitor their emotional patterns over time.

Based on the emotional analysis, the system categorizes results into three levels:

- **Low Risk:** Indicates stable emotional conditions.
- **Moderate Risk:** Suggests noticeable emotional stress that may require attention.
- **High Risk:** Signals serious emotional distress and recommends seeking professional help.

### 4.4 Database Design

The system uses a MySQL database to store user authentication and interaction records.

**Table 1: users**

This table stores registered user credentials securely.

Field	Type	Description
<b>Name</b>	VARCHAR(100)	Not Null
<b>Email</b>	VARCHAR(150)	Unique
<b>Phone</b>	VARCHAR(20)	Unique
<b>Password</b>	VARCHAR(255)	Bcrypt hashed
<b>Created_at</b>	DATETIME	Default Current Timestamp

**Table 2: chat\_history**

This table maintains the persistent log of all interactions.

Field	Type	Description
<b>ID</b>	<b>INT</b>	<b>Primary Key, Auto Increment</b>
<b>User_ID</b>	<b>INT</b>	<b>Foreign Key</b>
<b>User_msg</b>	<b>TEXT</b>	<b>User message text</b>
<b>Bot_reply</b>	<b>TEXT</b>	<b>Chatbot response text</b>
<b>Intent</b>	<b>VARCHAR(100)</b>	<b>Predicted intent category</b>

#### 4.5 Machine Learning Model

The system uses machine learning to identify mental health–related intents and provide supportive responses. Logistic Regression is used as the main classification model because it performs well with conversational text. The model is trained on a dataset containing over 250 conversational patterns across 17 mental health categories such as anxiety, depression, and stress.

Based on the prediction confidence, the system provides different responses:

- **Detected Intent:** Identifies emotional states like stress or anxiety.
- **Fallback Response:** Neutral reply if confidence is below **25%**.
- **Crisis Referral:** Suggests professional helpline support for high-risk inputs.

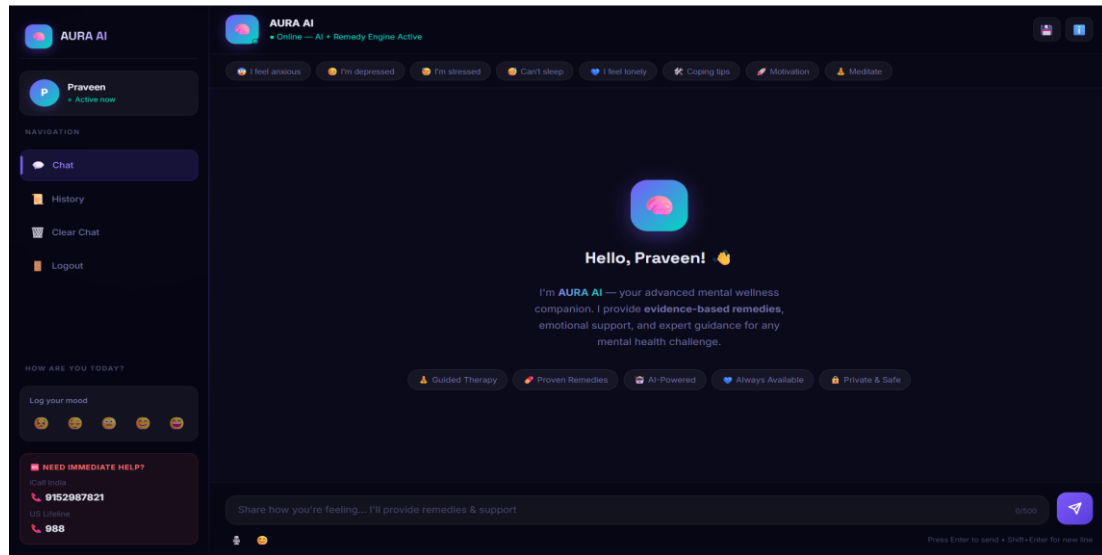
## 5. RESULTS AND DISCUSSION

The implemented system successfully detects mental health–related intents and provides real-time emotional support. User messages are processed through the NLP pipeline and delivered from the Flask backend to the web interface with minimal delay. The dashboard displays real-time chats and mood-tracking insights for better user understanding.

Key results include:

- **Accurate Intent Detection:** The model identifies emotions such as anxiety, stress, and depression from user messages.
- **Safe Responses:** Low-confidence inputs are handled using neutral fallback replies to avoid incorrect guidance.
- **Better Conversations:** Integration with a Google Gemini–powered agent enables more natural multi-turn interactions.

- **Improved Accessibility:** The system offers 24/7 support without the need for appointments or professional fees.
- **Early Awareness:** Mood tracking and wellness suggestions help users understand and manage their emotional state.



**Figure 3: AURA AI Chatbot User Interaction Interface.**

## 6. CONCLUSION

This project introduced AURA AI, an intelligent mental health chatbot that uses Natural Language Processing and machine learning to provide accessible emotional support and early mental health guidance.

The system allows users to interact in real time through a web-based platform and analyzes their messages to identify emotional states such as anxiety, stress, and depression. With a responsive dashboard and a dual AI engine, users can receive helpful suggestions while also tracking their emotional patterns over time.

Overall, the system helps improve access to mental health support, encourages people to seek help early, and provides immediate 24/7 assistance in a modern digital environment.

## 7. Future Work

Future improvements to the system could further enhance its capabilities and user experience. These may include integrating advanced transformer-based models such as BERT to better understand context and more complex user inputs. Adding multilingual support through language detection and translation can help make the platform accessible to a wider global audience.

The system could also include voice interaction features, such as voice input and text-to-speech, to improve accessibility for different types of users. A personalization and mood analytics module could be introduced to track emotional trends and provide customized wellness suggestions based on a user's history. In addition, implementing cloud-based infrastructure and mobile integration would allow the platform to scale easily and support remote monitoring of mental well-being.

## REFERENCES

1. Pedregosa, F., et al., Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research (JMLR), 2011. <https://scikit-learn.org/> (Used for TF-IDF and Logistic Regression-based intent classification)
2. Python Software Foundation, Python Programming Language Documentation, <https://www.python.org/>
3. Flask Documentation, Flask – A Lightweight WSGI Web Application Framework, <https://flask.palletsprojects.com/>
4. MySQL Documentation, MySQL Database Management System, <https://dev.mysql.com/doc/>
5. Google AI, Google Gemini API Documentation, <https://ai.google.dev/> (Used for advanced multi-turn context-aware conversations)
6. Fitzpatrick, K. K., et al., Delivering CBT using a fully automated conversational agent, JMIR Mental Health, 2017. (Reference for evidence-based digital therapy)
7. Jurafsky, D., & Martin, J. H., Speech and Language Processing, Stanford University, 2023. <https://web.stanford.edu/~jurafsky/slp3/>
8. World Health Organization (WHO), Mental Health and Well-Being Guidelines, <https://www.who.int/>
9. American Psychiatric Association, Diagnostic and Statistical Manual of Mental Disorders (DSM-5-TR), <https://psychiatry.org/>
10. Géron, A., Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media, 2019.