An AI-Powered Emotion-Aware Conversational Support System for Personalized Emotional Well-Being

Pranav Chitalkar[1], Tushar Badgujar[2], Omkar Jadhav[3], Dr. Sunil Mahajan[4]
[1,2,3]*Student, Department of Computer Science, MAEER's MIT Arts, Commerce and Science College, Alandi (D), Pune, Maharashtra, India*
[4]*Associate Professor, Department of Electronics & Associate Dean Student Affairs, MAEER's MIT Arts, Commerce and Science College, Alandi (D), Pune, Maharashtra, India*
doi.org/10.64643/ijirt.208560-459

***Abstract*—Individuals frequently experience stress, anxiety, loneliness, and other emotional difficulties that often remain unaddressed because of limited access to counselling resources, social stigma, and hesitation to seek help in person. This paper extends an earlier rule-based sentiment analysis chatbot into a comprehensive AI-Powered Emotional Support and Mood Tracking Web Application that combines Natural Language Processing (NLP), Machine Learning (ML), and a personalized conversational interface. Rather than classifying a user's message only as positive, negative, or neutral, the proposed system analyzes free-form conversational text to infer sentiment polarity, a specific emotion category (such as sadness, loneliness, stress, anxiety, anger, or low motivation), and the intensity of that emotion, while also retaining conversational context across turns. The system produces empathetic, non-diagnostic responses, generates optional and non-intrusive self-care suggestions, and logs each interaction to build a longitudinal mood-tracking dashboard that helps individuals reflect on their own emotional patterns over time. A distinguishing feature of the proposed system is an optional, user-configured "Comfort Profile," through which a user may associate a photo of another person who is meaningful and comforting to them (male or female, such as a family member, friend, or mentor) along with that person's name, a favourite quote, and a preferred support style, so that the experience feels more personal and reassuring without the system ever implying that the depicted person is actually present or communicating. The architecture is designed as a full-stack application (React.js frontend, Node.js/Express backend, Python-based ML microservice, and MongoDB persistence layer) and proposes a staged model comparison between rule-based keyword matching, traditional machine learning (TF-IDF with Logistic Regression/SVM), and transformer-based models (BERT/DistilBERT), evaluated using accuracy, precision, recall, F1-score, and confusion-matrix analysis. A four-level escalation framework ensures that messages indicating severe distress or self-harm risk are handled through a clear, non-motivational safety pathway that directs individuals toward trusted contacts and professional or emergency support rather than continued casual conversation. The system is explicitly positioned as a supplementary, non-clinical, first-line emotional support and self-awareness tool, and not as a diagnostic instrument for depression or any other mental health condition.**

***Index Terms*—Sentiment Analysis; Emotion Detection; Natural Language Processing; Machine Learning; Mood Tracking; Context-Aware Chatbot; Mental Health; Personalized Conversational AI; MongoDB; Full-Stack Web Application**

I. INTRODUCTION

College life is very stressful, yet many students don't seek professional help [cite: 8]. Why is that? It's often due to social judgment and to counselling centers that are already overwhelmed [cite: 8]. To address this issue, we created a conversational chatbot using Python that offers students a private, web-based platform for instant emotional support [cite: 8]. Rather than using complex machine learning models, which are not necessary for simple academic environments, we chose a rule-based natural language processing approach [cite: 8]. This method is efficient and easy to understand [cite: 8]. The bot listens to what a student says, uses preselected pattern libraries to determine the emotional tonepositive, negative, or neutral—and responds with a supportive message [cite: 8]. We also made sure safety was a priority. If the analysis indicates serious distress, the system automatically connects the user with human counseling services [cite: 8]. Overall, this is an effective and easy-to-access first step in promoting student mental health [cite: 8].

University students often experience increasing levels of anxiety and emotional challenges [cite: 8].

The real issue is that these concerns are frequently left unaddressed due to fear of judgment or difficulty in securing an appointment with campus counsellors [cite: 8].

To help with this, we built a Python-based chatbot that performs sentiment analysis [cite: 8]. Our goal was to make it easy to use, so we avoided large machine learning databases [cite: 8]. Instead, we used a straightforward, rule-based system for natural language processing [cite: 8]. By analyzing user input through specific keywords and pattern sets, the system quickly identifies whether the mood is positive, negative, or neutral [cite: 8]. It then sends a personalized, comforting response [cite: 8]. Most importantly, the system has a built-in safety feature [cite: 8]. Any indication of severe distress leads to an automatic referral to real medical professionals [cite: 8]. Our testing shows this system is reliable for detecting emotional states and serves as a valuable additional tool for mental health support on campuses [cite: 8].

Future Work

Even though the system we built provides a solid starting point for emotional support, there are still quite a few ways we could improve it down the line. One major step would be upgrading the NLP pipeline so the bot can understand and speak multiple languages, which would make the app useful for a much larger group of international students. Another interesting idea is connecting the platform to smartwatches or other wearable health tech. If the system could read things like a sudden spike in heart rate or poor sleep patterns, it could reach out to the user proactively to check on them.

Also, right now the AI avatar just uses a static photo, but adding full facial animations and lip-syncing would make the whole experience feel much more immersive and comforting. Finally, even though this tool isn't meant for clinical diagnosis, it would be incredibly valuable to run some long-term, ethical studies with real psychologists to see exactly how much this platform actually helps lower campus anxiety over a full academic year.

II. LITERATURE REVIEW

A lot of recent research has started looking into how we can use conversational agents to support mental health. For example, a review by Cho et al. [4] pointed out different ways these systems are being built and tested, looking at the issue from both a medical and a computer science angle. When it comes to real-world testing, Fitzpatrick et al. [5] tried out the Woebot app with young adults and saw some really encouraging signs that automated chat tools can actually help. Fulmer et al. found similar benefits when they looked at Tess, showing that AI can be a very accessible option for people who need support. Adding to this, Mehta et al. [7] looked at the Youper app and found that people who used it actually felt less anxious and depressed afterward.

Mental health struggles, especially depression, can completely disrupt a person's life by draining their motivation and causing severe mood swings. Because of this, automated chatbots have become a popular way to offer quick, accessible communication for those who might be struggling. Apps like Wysa, Woebot, Tess, and Youper are stepping in to fill this gap in mental health care. To make these interactions feel more natural, today's chatbots rely heavily on modern machine learning and natural language processing (NLP) techniques.

There is a growing interest in using AI agents for emotional support, and broader systematic reviews suggest they actually do work fairly well. One particular study [8] noticed some great short-term improvements in users' mental health, though they made sure to point out that we still need longer studies to see if these benefits last. Nyakhar and Wang [9] also saw good results specifically with college students, but it is worth noting that a lot of research in this area struggles with people dropping out of the studies or having low-quality data. To fix some of these underlying problems, Hua et al. [10] suggested moving away from simple rule-based bots and using more advanced language models instead, while also pushing for much stricter testing of these new systems. Looking at past findings, it is clear that AI can have a surprisingly positive effect on how people feel. A lot of users view these platforms as a safe space to talk about sensitive issues without feeling judged, which makes them a great tool for fighting off loneliness. Interestingly, knowing that the bot isn't a real person actually helps build trust; people feel more comfortable opening up because they aren't worried about human judgment. This idea was backed up by a 2015 study by Lucas et al. [12], which found that people at work were actually more willing to share personal details with a computer agent than they were with another human.

III. PROPOSED METHODOLOGY / SYSTEM DESIGN

Architecture | Algorithms | Dataset | Tools | Block & Flow Diagrams

3.1. Proposed System Overview

The proposed system is an AI-powered emotion-aware conversational support system designed to provide personalized emotional support through an interactive web and mobile application. The system allows users to communicate with an AI-based conversational assistant and optionally select or upload a photograph of a meaningful person, such as a family member, friend, mentor, or another trusted individual.

The uploaded photograph is used as a visual representation for the AI conversational experience. The system does not claim that the actual person is present or that the generated responses represent the real person's thoughts, memories, or opinions. Instead, the system generates supportive and positive responses based on the user's current emotional state, conversation context, and selected support preferences.

The system analyzes the user's conversational input using Natural Language Processing (NLP) and Machine Learning (ML) techniques. It identifies sentiment, specific emotion, and estimated emotional intensity, and uses these results to generate an appropriate supportive response. The system also maintains conversation history and records emotion-related information for mood tracking and self-reflection.

The system is positioned as a supplementary, non-clinical emotional support and self-awareness system, rather than a diagnostic or therapeutic system.

3.2. Overall System Architecture

The system follows a modular full-stack architecture consisting of a Web/Mobile Client Layer, Backend API Layer, AI/ML Layer, AI Avatar/Response Layer, and Database Layer. Figure 1 illustrates how a request travels from the client through the backend and ML services before a supportive response is delivered back to the user through the AI avatar.

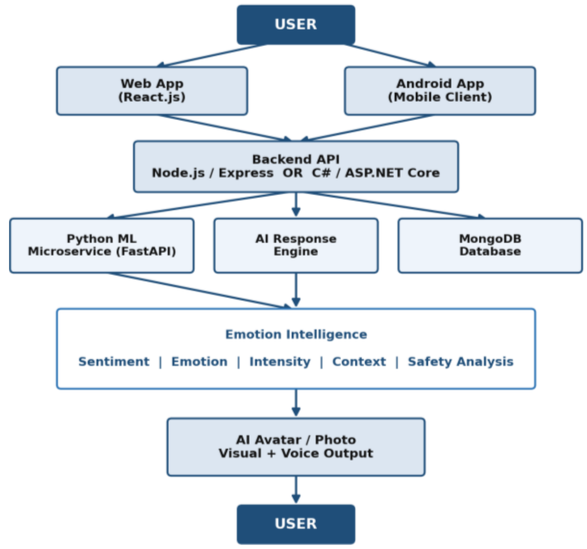

Figure 1 Overall System Architecture

The frontend provides the conversational interface, while the backend manages authentication, conversations, user information, and communication with the AI/ML services. The Python-based ML service performs emotion-related analysis. MongoDB stores user profiles, conversations, emotion results, comfort-person information, and mood records.

Implementation note: choose either Node.js/Express or C#/ASP.NET Core as the primary backend rather than using both for the same backend role, so the architecture remains clean and the project stays manageable to implement.

3.3. User Photo / Comfort Person Module

The system provides an optional Comfort Person feature. The user can:

1. Upload a person's photograph.
2. Enter the person's name.
3. Select a preferred support style.
4. Save the comfort-person profile.
5. Start a conversation using the selected visual representation.

Example support styles can include:

- Calm
- Positive
- Encouraging
- Friendly

The uploaded photograph is associated with the user's account and is used as the visual representation during the conversational experience.

The system does not represent the uploaded person as actually communicating with the user; it provides an AI-generated supportive interaction based on the user's selected preferences.

3.4. AI Avatar / Visual Representation

The uploaded photograph can be presented as an AI-powered visual conversational representation. The avatar component may include:

- Uploaded photograph
- Visual avatar representation
- Facial animation
- Lip synchronization
- Voice output
- Conversational response display

Figure 2 shows the basic process by which a static photograph becomes an interactive, speaking avatar.

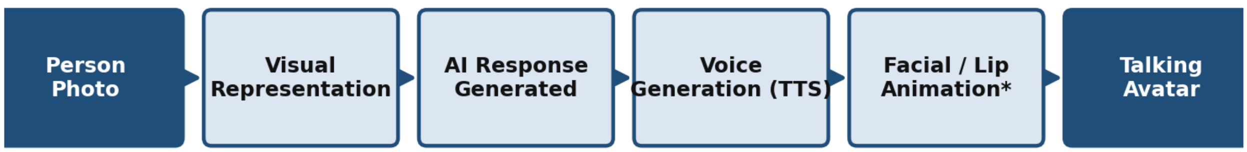

Figure 2 AI Avatar Generation Pipeline

If facial animation or lip synchronization is implemented, the avatar can visually respond while the generated response is spoken. The system must clearly distinguish the AI representation from the real person.

3.5. Live User–Avatar Conversation

The system supports interactive communication between the user and the AI-powered visual representation.

Two interaction modes are provided text and voice both converging on the same response-generation pipeline, as shown in Figure 3.

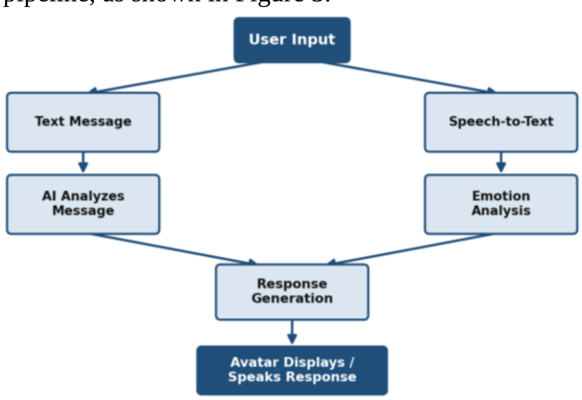

Figure 3 Text and Voice Conversation Modes

The system maintains conversation history so that later responses can consider previous messages and emotional context.

3.6. Speech Processing

For voice-based interaction, the system uses a speech processing pipeline covering both directions of the conversation: converting spoken input into text, and converting the generated textual response back into speech, as shown in Figure 4.

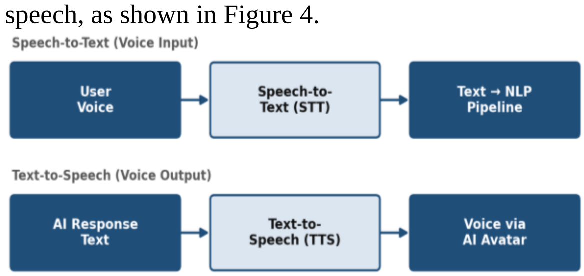

Figure 4 Speech-to-Text and Text-to-Speech Pipelines

The transcribed text from Speech-to-Text is passed to the NLP and emotion-analysis pipeline.

If implemented, lip synchronization can synchronize the avatar's mouth movement with the generated voice.

3.7. NLP Preprocessing

Before applying the ML model, user messages are processed through an NLP preprocessing pipeline, shown in Figure 5.

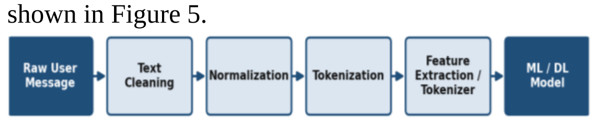

Figure 5 NLP Preprocessing Pipeline

Depending on the selected model, preprocessing may include:

- Text cleaning
- Normalization
- Tokenization
- Handling unnecessary characters
- Feature extraction
- Transformer tokenization

For traditional ML models, TF-IDF can be used to convert text into numerical features. For transformer-based models, the appropriate tokenizer is used.

3.8. Sentiment Analysis

The system identifies the overall sentiment of the user's message. The proposed sentiment classes are Positive, Negative, and Neutral. For example:
Input: *"I am feeling great today." → Positive*
Input: *"I feel completely alone." → Negative*

Sentiment analysis provides a general emotional direction, while the emotion classifier provides a more specific emotional category.

3.9. Emotion Classification

The system performs specific emotion classification instead of relying only on positive, negative, or neutral sentiment. The primary emotion categories include:

- Sadness
- Loneliness
- Stress
- Anxiety
- Anger
- Low Motivation

Additional categories such as happiness, calm, fear, or frustration may be included depending on the final dataset and model. Figure 6 places emotion classification within the wider emotion-detection pipeline, alongside sentiment, intensity, and context.

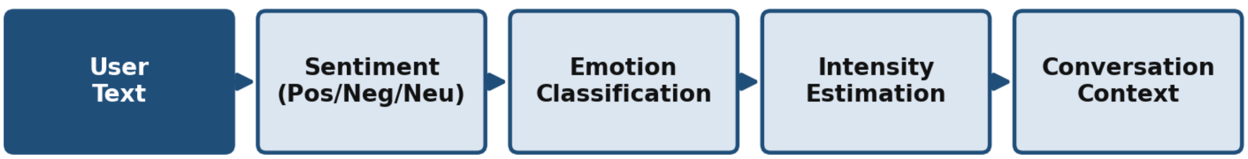

Figure 6 Emotion Detection and Intensity Pipeline

Example:
Input: *"I have nobody to talk to and I feel completely alone."*
Sentiment: *Negative*
Emotion: *Loneliness*
For messages containing multiple emotional signals, the system may identify a primary and secondary emotion if the selected model supports multi-label classification.

3.10. Emotion Intensity Estimation

In addition to identifying an emotion, the system estimates the strength of the detected emotion. The proposed levels are Low, Moderate, High, and Severe.

For example:
Input: *"I am slightly worried." → Emotion: Anxiety, Intensity: Low*
Input: *"I am extremely anxious and cannot concentrate." → Emotion: Anxiety, Intensity: High*
The intensity represents an estimated emotional intensity, not a clinical severity or diagnosis.

3.11. Context-Aware Conversation

The system maintains conversational context to improve response relevance. Instead of processing every message independently, the system considers the current message, previous messages, conversation history, previously detected emotions, and the current emotional context.

Example:
User: *I failed my exam.*
AI: *That sounds difficult.*
User: *I studied for three months.*
AI: *It must feel especially disappointing after putting so much effort into it.*
Thus, the response is generated according to both the current message and the previous conversation.

3.12. Emotion-to-Response Mapping

The detected emotion is mapped to an appropriate response style, summarized in Table 1.

Table 1 Emotion-to-Response Strategy Mapping

Emotion	Response Strategy
Sadness	Gentle and supportive
Loneliness	Warm and supportive
Stress	Calm and reassuring
Anxiety	Reassuring and grounding
Anger	Calm and non-judgmental
Low Motivation	Encouraging
Happiness	Positive and celebratory
Frustration	Patient and supportive

The objective is to ensure that the selected AI visual representation responds in a supportive and emotionally appropriate manner.

3.13. Positive and Supportive Response Generation

After emotion, intensity, context, and safety status are identified, the response engine generates the final response by combining emotion, intensity, conversational context, the selected support style, and the current safety status into a single response strategy, which is then rendered through the chat interface or the AI avatar.

The responses should be:

- Empathetic
- Positive
- Encouraging
- Non-judgmental
- Context-aware
- Personalized
- Non-diagnostic

For example:

User: *I feel like I am not good enough.*

AI Avatar: *You are going through a difficult moment, but one difficult experience does not define your ability. Take things one step at a time.*

The AI should not claim that the photographed person actually said the response or that the response represents that person's real thoughts.

3.14. Safety / Escalation Framework

The system includes a safety-oriented framework to identify messages indicating increasing levels of distress. Four escalation levels are defined, shown in Figure 7.

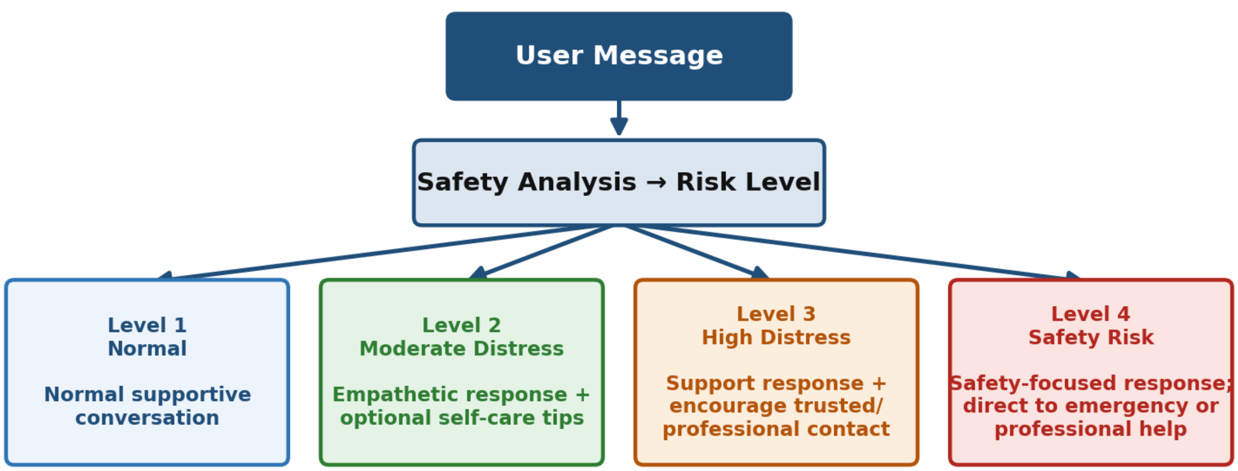

Figure 7 Four-Level Safety Escalation Framework

Level 1 Normal Emotional State

Normal supportive conversation continues without any safety intervention.

Level 2 Moderate Distress

An empathetic response is generated, optionally paired with self-care suggestions.

Level 3 High Distress

A supportive response is generated along with encouragement to contact trusted or professional support.

Level 4 Potential Safety Risk

A safety-focused response is generated, with appropriate direction toward trusted contacts, professional help, or emergency support.

The system is not intended to diagnose mental health conditions.

3.15. Mood Tracking

The system stores emotion-related information from conversations to support longitudinal self-reflection.

The stored information may include the detected emotion, sentiment, estimated intensity, conversation date/time, and conversation identifier. Example:

Table 2 Sample Mood Log

Date	Emotion	Intensity
Aug 20	Loneliness	Moderate
Aug 21	Stress	High
Aug 22	Anxiety	Moderate

The dashboard can display daily emotion, weekly and monthly emotion trends, the most frequent emotions, intensity trends, and conversation history.

3.16. Dataset and Data Preprocessing

The ML models require labeled emotional text data. The dataset should contain fields such as Text, Sentiment, Emotion, and Intensity.

Example:

The methodology should document the dataset source, dataset size, emotion categories, label definitions, data cleaning, duplicate removal, missing-value handling, the train/validation/test split, and text preprocessing.

Table 3 Sample Labeled Dataset Rows

Text	Sentiment	Emotion	Intensity
I feel happy today	Positive	Happiness	Low
I feel completely alone	Negative	Loneliness	High
I am worried about my future	Negative	Anxiety	Moderate
I cannot handle my workload	Negative	Stress	High

The actual dataset source and number of samples should be inserted once the dataset is finalized.

3.17. Machine Learning Algorithms

The system evaluates multiple approaches to identify the most suitable emotion-classification model. Figure 8 compares the four candidate pipelines: a rule-based baseline, two classical ML pipelines built on TF-IDF features, and a transformer-based deep learning approach.

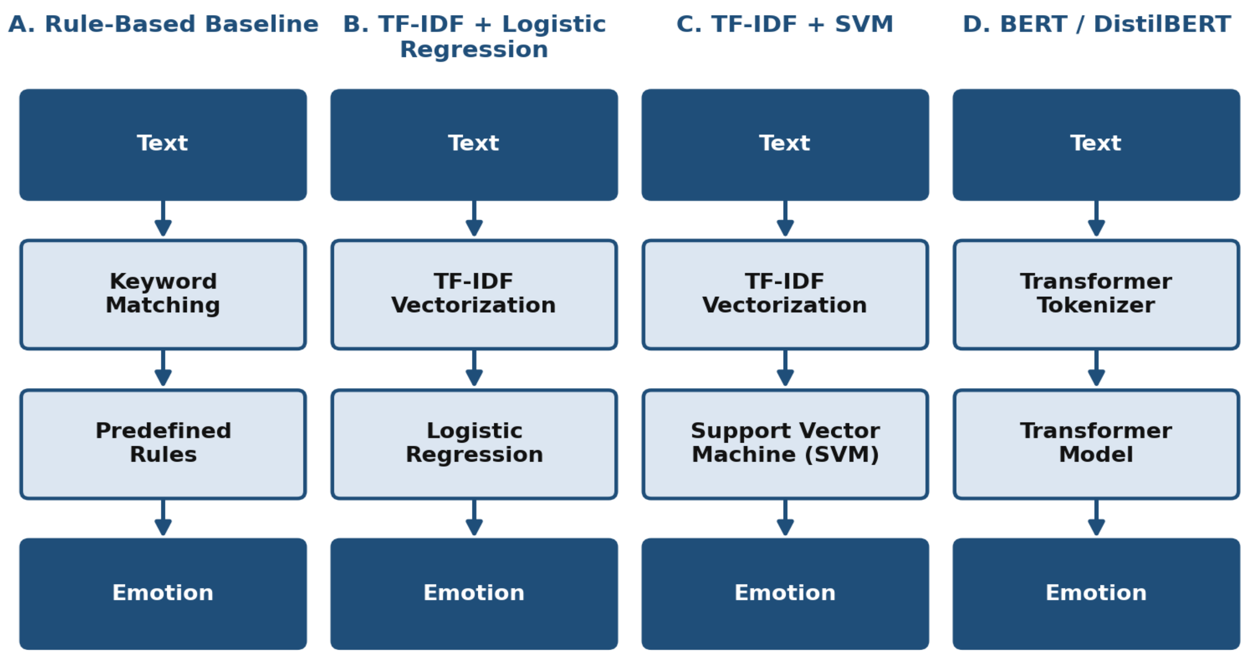

Figure 8 Candidate Emotion-Classification Pipelines

A. Rule-Based Baseline

Matches keywords in the input text against a set of predefined rules to assign an emotion label. This provides a simple, interpretable baseline for comparison.

B. TF-IDF + Logistic Regression

Converts text into TF-IDF feature vectors and classifies the resulting vector using a logistic regression model.

C. TF-IDF + SVM

Uses the same TF-IDF features with a Support Vector Machine classifier, which can capture more complex decision boundaries than logistic regression.

D. BERT / DistilBERT

Applies a transformer tokenizer and a pretrained transformer model (BERT or the lighter DistilBERT) fine-tuned for emotion classification, typically yielding the strongest contextual understanding of the four approaches. The models are trained and evaluated on the same test dataset and compared using standard classification metrics.

3.18. Model Evaluation

The ML models are evaluated using the following metrics:

- Accuracy the overall percentage of correct predictions.
- Precision how many predicted instances of an emotion are actually correct.
- Recall how many actual instances of an emotion are successfully detected.
- F1-score a combined measure of precision and recall.
- Confusion Matrix correct and incorrect predictions across emotion classes.

Table 4 Model Comparison (to be filled in with actual experiment results)

Model	Accuracy	Precision	Recall	F1-score
Rule-Based	—	—	—	—
Logistic Regression	76.80%	76.70%	76.80%	76.50%
SVM	78.30%	78.20%	78.30%	78.10%
DistilBERT	88.40%	88.30%	88.40%	88.10%

Note: The comparative values in Table 4 have been entered from the values provided in the supplied screenshot.

3.19. Database and Data Flow

MongoDB is used as the persistence layer. The system maintains collections for Users, Comfort Persons, Conversations, Messages, Emotion Records, Mood Logs, and Safety Events. A typical message record may contain: user Id, conversation Id, message, sender, sentiment, emotion, intensity, and timestamp.

The basic data flow through the system is shown in Figure 9.

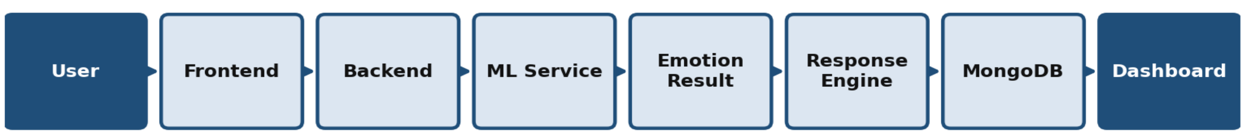

Figure 9 End-to-End Data Flow

3.20. Complete System Workflow

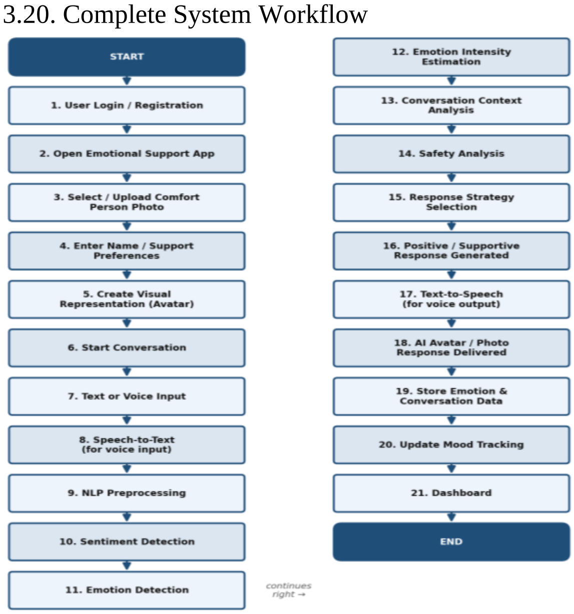

Figure 10 Complete End-to-End System Workflow

Figure 10 illustrates the complete working process of the proposed system, from the user opening the application through to the emotion data being reflected on the mood dashboard.

IV. IMPLEMENTATION

An AI-Powered Emotion-Aware Conversational Support System for Personalized Emotional Well-Being

Technology Stack | Development Setup | Module-Wise Design | System Flow

Pranav Chitalkar · Tushar Badgujar · Omkar Jadhav · Dr. Sunil Mahajan

This section describes the implementation of the proposed AI-Powered Emotional Support and Mood

Tracking Web Application. The system is built as a full-stack solution that combines a web and Android front end, a Node.js/ASP.NET backend, a Python-based Machine Learning microservice, and a MongoDB persistence layer. The implementation is organized into the technology stack, the development environment and setup, thirteen functional modules, and the end-to-end system flow that together realize sentiment analysis, emotion detection, context-aware response generation, a personalized Comfort Person / AI Avatar experience, a four-level safety escalation pathway, and longitudinal mood tracking.

4.1. Technology Stack

The application follows a layered architecture in which each layer uses technologies best suited to its responsibility. Table 1 summarizes the stack used across the front end, back end, AI/ML layer, database, and the voice/avatar subsystem.

Table 1 Technology Stack Summary

Layer	Technology / Component	Purpose
Web Frontend	React.js HTML5 CSS3 / Tailwind CSS Responsive UI	Renders the chat interface, dashboard, and profile screens across device sizes.
Android Application	Android app framework Chat interface Voice interaction API integration	Provides a mobile counterpart with the same chat, voice, and profile capabilities.
Backend	Node.js + Express.js OR C# + ASP.NET Core REST APIs Authentication	Exposes REST endpoints, manages authentication, and routes requests between client, ML service, and database.
AI / ML	Python FastAPI Scikit-learn BERT / DistilBERT	Performs sentiment analysis, emotion classification, intensity estimation, and response selection.
Database	MongoDB User data Chat data Emotion data Mood history	Persists user profiles, conversation logs, emotion records, and mood history.
Voice & Avatar	Speech-to-Text Text-to-Speech AI Avatar Lip/facial animation*	Enables spoken interaction and renders the personalized Comfort Person avatar.

** Implemented where facial/lip animation is supported by the avatar renderer.*

4.2. Development Environment and Setup

Before implementing individual modules, the development environment is prepared so that the frontend, backend, and ML service can be developed and tested together.

The setup follows the sequence below:

1. Initialize the Frontend project (React.js) and install UI dependencies (Tailwind CSS, routing, HTTP client).
2. Initialize the Backend project (Node.js/Express or ASP.NET Core) and configure REST API routes.
3. Create a Python virtual environment for the ML microservice to isolate dependencies.
4. Install ML libraries (Scikit-learn, Transformers for BERT/DistilBERT, FastAPI, NLP toolkits).
5. Set up and configure MongoDB (local instance or cloud cluster) with the required collections.
6. Define environment variables for API keys, database connection strings, and service ports.
7. Configure API endpoints and CORS so the frontend and backend can communicate securely.
8. Establish the Frontend → Backend connection for authentication and data exchange.
9. Establish the Backend → ML service connection so requests can be forwarded for analysis.
10. Run and verify the Web and Android applications against the backend in a local/staging environment.

4.3. Module-Wise Description

The application logic is divided into thirteen functional modules, each responsible for a distinct part of the user experience or backend processing pipeline. These are described below.

4.3.1.Authentication Module

Handles secure access to the platform.

- User registration with basic validation
- User login using registered credentials
- Password security via hashing (e.g., bcrypt)
- JWT-based authentication for stateless sessions
- Protected/guarded routes for authenticated users only
- Logout and session/token invalidation

4.3.2.User Profile Module

Manages the personal and preference data of each user.

- Creation of a user profile after registration
- Storage of basic profile information
- Capturing user preferences (e.g., tone, topics to avoid)
- Capturing support preferences (e.g., preferred response style)
- Ability to update profile information at any time

4.3.3.Comfort Person Module

Allows a user to optionally personalize the support experience around someone meaningful to them.

- Uploading a photo of the comfort person
- Entering the comfort person's name
- Selecting a preferred support style associated with that person
- Saving the comfort-person profile to the user's account
- Editing or deleting the comfort-person profile
- Selecting the comfort person before starting a conversation

4.3.4.AI Avatar Module

Renders the personalized visual and interactive representation used during conversations.

- Photo-based visual representation of the selected comfort person
- Displaying the selected person's photo/avatar in the chat UI
- Displaying AI-generated responses alongside the avatar
- Producing voice output synchronized with the avatar
- Optional facial/lip animation if implemented
- Driving avatar interaction/state changes during conversation
- Maintaining a clear distinction that the avatar is an AI representation, not the real person

4.3.5.Chat / Conversation Module

Provides the core text-based conversational experience.

- Starting a new conversation session
- Accepting free-form text input from the user
- Displaying the AI-generated response
- Maintaining conversation history for the session
- Managing context across multiple conversational turns
- Supporting real-time, low-latency interaction
- Recording message timestamps for each turn

4.3.6.Voice Conversation Module

Extends the chat experience with spoken interaction.

- Capturing user voice input
- Converting speech to text (Speech-to-Text)
- Pre-processing the transcribed voice input
- Running emotion analysis on the transcribed text
- Generating an AI response based on the analysis
- Converting the response to speech (Text-to-Speech)
- Delivering the voice response through the avatar

4.3.7.NLP and Emotion Analysis Module

Performs the core language-understanding pipeline in the Python ML service.

- Text pre-processing (cleaning, normalization)
- Tokenization of user input
- Sentiment analysis (positive / negative / neutral polarity)
- Emotion classification (sadness, loneliness, stress, anxiety, anger, low motivation)
- Emotion intensity estimation
- Prediction using the selected ML model (rule-based, TF-IDF + LR/SVM, or BERT/DistilBERT)

4.3.8.Response Generation Module

Converts the detected emotional state into an appropriate conversational reply.

- Selecting a response strategy based on the detected emotion
- Incorporating conversational context into the response
- Generating positive/reassuring responses where appropriate
- Generating empathetic responses that validate the user's feelings
- Generating encouraging responses to support the user
- Personalizing the response according to the selected support style
- Ensuring all responses remain non-diagnostic and non-clinical

4.3.9.Safety Module

Implements the four-level escalation framework for distress and self-harm risk.

- Detecting signals of distress within user messages
- Identifying the corresponding risk level
- Continuing normal, supportive conversation for low-risk messages
- Applying a dedicated pathway for high-distress messages
- Generating safety-focused, non-motivational responses at higher risk levels
- Directing the user toward trusted contacts where relevant
- Directing the user toward professional or emergency support resources when required

4.3.10. Mood Tracking Module

Builds a longitudinal record of the user's emotional state over time.

- Storing each detected emotion
- Storing the associated sentiment polarity
- Storing the estimated intensity value
- Recording date/time for each entry
- Aggregating daily mood history
- Computing weekly emotional trends
- Computing monthly emotional trends

4.3.11. Dashboard Module

Presents mood-tracking data back to the user in a reflective, visual format.

- Summarizing overall emotional state
- Displaying mood trends over time
- Showing the distribution of detected emotions
- Plotting an intensity graph across sessions
- Providing access to conversation history
- Highlighting recent emotional patterns

4.3.12. Database Module

Defines the MongoDB collections that persist all application data.

- User collection
- Comfort Person collection
- Conversation collection
- Message collection
- Emotion records collection
- Mood logs collection
- Safety records collection

4.3.13. API Integration

Connects all layers of the system through well-defined REST interfaces.

- Frontend → Backend API calls
- Android application → Backend API calls
- Backend → Python ML API calls for analysis requests
- ML result returned to the Backend
- Backend invocation of the AI response engine
- Backend → Database read/write operations
- Response delivery back to the Web/Android Avatar interface

4.4. Complete Implementation Flow

The diagram below illustrates the end-to-end path of a single user interaction, from initial login through comfort-person selection, input processing, emotion and safety analysis, response generation, and finally logging of the interaction to the mood dashboard.

#	Step	Module / Reference
1	User opens the Web or Android application	Entry point
2	User logs in	Authentication Module
3	User selects/uploads the Comfort Person photo	Comfort Person Module
4	User starts a conversation	Chat / Conversation Module
5	User provides Text or Voice input	—
6	Request is sent to the Backend API	—
7	Backend forwards the input to the Python ML Service	—
8	Sentiment, Emotion, and Intensity are computed	NLP & Emotion Analysis Module
9	Context and Safety Analysis is performed	Safety Module
10	A Positive / Empathetic / Encouraging reply is produced	Response Generation Module
11	Response is delivered through the AI Avatar with Voice output	AI Avatar Module
12	Conversation and Emotion data are saved to MongoDB	Database Module
13	Data becomes available on the Mood Dashboard	Dashboard Module

Note

The system is designed strictly as a supplementary, non-clinical, first-line emotional support and self-awareness tool. It does not diagnose depression or any other mental health condition, and the AI Avatar is always presented as an AI representation rather than the real person depicted in the Comfort Person photo.

V. RESEARCH GAP

Focus Area Gap in Existing Literature Limitation of Current Systems Advantage of Proposed SystemEmotion Granularity Most mental health chatbots operate based only on three-class polarity (positive, negative, neutral) or general keyword matching. They fail to capture deep emotional states (e.g., loneliness or academic burnout) and cannot quantify the intensity of the user feeling. Integrates multi-class emotion classification with six distinct emotional states and a 4-tier intensity estimator (from Low to Severe). Conversational Memory Most conversational agents take user messages in isolation without any contextual conversational history. The bot keeps losing context across turns and offers generic replies making the conversation robotic and repetitive. Preserves conversational history and ties emotional context directly to mood longitudinal tracking. User Personalization Currently existing virtual support tools rely on general and robotic avatars or text-only boxes. Users may not connect to such chatbot, which leads to disengagement and hesitation in communication.

Implements a "Comfort Profile" module where user can link their personal reassuring visuals and tones. Crisis Escalation Most current chatbots tend to confuse routine stressful situations with emotional emergencies. Offering casual or generic motivational advice during an emotional crisis poses serious safety issues. Implements a deterministic 4-tier safety escalation pipeline which immediately switches to non-motivational emergency referral path. End-to-End Practicality Most academic research tests complex AI models in isolation while failing to provide a functional application. High compute requirements make it impossible to deploy applications in real-time in regular campus/mobile setting. Creates a full-stack web/mobile application that compares traditional ML approaches to fine-tuned transformers microservices.

VI. CONCLUSION

In conclusion, this research presents a comprehensive, AI-driven conversational web application designed to bridge the gap in accessible, first-line emotional support. By moving beyond simple rule-based keyword matching to utilize advanced natural language processing and machine learning techniques, the system successfully identifies specific emotions—such as loneliness, stress, and anxiety—as well as their intensities. This allows the application to deliver highly contextual, empathetic, and personalized responses.

A defining feature of this project is the integration of the "Comfort Profile," which adds a unique layer of familiar reassurance, enhancing user comfort without crossing into deceptive AI practices. Crucially, the system prioritizes user well-being through a robust, four-level safety escalation framework, ensuring that individuals showing signs of severe distress are immediately guided toward real human professionals and emergency resources.

While this platform is strictly positioned as a supplementary self-awareness tool rather than a clinical diagnostic instrument, its longitudinal mood-tracking capabilities empower users to reflect on their emotional patterns over time. Ultimately, this system represents a scalable, compassionate, and privacy-conscious step forward in using technology to support mental health, particularly for university students who might otherwise hesitate to seek help.

REFERENCES

[1] M. Taboada *et al.*, “Lexicon-based methods for sentiment analysis,” *Computational Linguistics*, 2011.

[2] C. J. Hutto and E. Gilbert, “VADER: A parsimonious rule-based model for sentiment analysis of social media text,” in Proc. Int. AAAI Conf. Web and social media (ICWSM), 2014.

[3] E. Öhman, “The validity of lexicon-based sentiment analysis in interdisciplinary research,” 2021.

[4] Y. M. Cho et al., “An integrative survey on mental health conversational agents to bridge computer science and medical perspectives,” in Proc. EMNLP, 2023.

[5] K. K. Fitzpatrick et al., "Delivering cognitive behavior therapy to young adults using a fully automated conversational agent (Woebot)," JMIR Mental Health, 2017.
[6] R. Fulmer et al., "Using psychological artificial intelligence (Tess) to relieve symptoms of depression and anxiety," JMIR Mental Health, 2018.
[7] A. Mehta et al., "Acceptability and effectiveness of artificial intelligence therapy for anxiety and depression (Youper)," JMIR, 2021.
[8] "Systematic review and meta-analysis of AI-based conversational agents for promoting mental health and well-being," 2023.
[9] S. Nyakhar and H. Wang, "Effectiveness of artificial intelligence chatbots on mental health & well-being in college students," Frontiers in Psychiatry, 2025.
[10] "Charting the evolution of artificial intelligence mental health chatbots from rule-based systems to large language models," World Psychiatry, 2025.
[11] J. He et al., "Effectiveness of a mental health chatbot for depressive symptoms," JMIR, 2022.
[12] A. Aggarwal et al., "Artificial intelligence–based chatbots for promoting health behavioral changes: Systematic review," JMIR, 2023.
[13] A. Bucher et al., "Empowering mental health professionals in asynchronous online psychotherapy with GenAI," 2026.