ZENALYZE : AI For Detecting And Managing Personal Stress Level

S.Shri Lekha[1], Sreshta Sridhar[2], G.Rajeswari[3]
[1, 2, 3] Dept of Computer Science and Engineering
[1, 2, 3] K.L.N.College of Engineering, Pottapalayam, Sivagangai

Abstract- *In today's fast-paced digital era, stress-related issues have become increasingly prevalent, demanding intelligent and accessible mental health solutions. This paper presents Zenalyze, an AI-driven chatbot designed for real-time stress detection and personalized emotional support. The system leverages Natural Language Processing (NLP) and sentiment analysis using the TextBlob library to interpret user emotions accurately. It integrates the Google Translate API for multilingual communication, ensuring inclusivity across diverse users. Unlike conventional wellness applications, Zenalyze combines mood tracking, AI-guided journaling, task management, and lifestyle monitoring within a unified Streamlit-based platform. The chatbot provides context-aware recommendations, including mindfulness exercises, relaxation techniques, and motivational prompts, tailored to the user's detected mood. By fusing emotional intelligence with behavioral insights, Zenalyze delivers an empathetic and adaptive support system. The modular architecture further enables future enhancements such as voice interaction, facial emotion detection, and predictive analytics, promoting a scalable and intelligent approach to stress management and emotional well-being..*

Keywords- Stress Detection, Sentiment Analysis, Mental Wellness, NLP, Chatbot, Personalized Recommendations, Streamlit.

I. INTRODUCTION

The escalating prevalence of stress-related disorders in modern society underscores an urgent need for intelligent, accessible, and proactive mental health support systems. In an age defined by constant connectivity and information overload, individuals are increasingly susceptible to chronic stress, which adversely affects cognitive function, productivity, and overall quality of life. While awareness of mental wellness is growing, many individuals lack the tools or resources for effective stress management. Traditional approaches, such as static mobile applications for meditation or manual mood journals, often offer generic advice and fail to adapt to the user's evolving emotional state in real-time. This gap highlights the necessity for an AI-driven platform that can not only detect stress but also provide immediate, personalized, and context-sensitive interventions.

Zenalyze is conceived to address this very challenge by functioning as a digital companion for emotional well-being. The system moves beyond conventional rule-based applications by integrating multiple dimensions of a user's daily life into a cohesive analytical framework. It continuously captures data points through user interactions, including textual journal entries, mood ratings, task completion status, and lifestyle logs related to sleep and nutrition. This multi-faceted data collection enables a holistic understanding of the factors influencing an individual's stress levels, allowing for more nuanced and effective support.

The core intelligence of Zenalyze is powered by Natural Language Processing (NLP) techniques. The system utilizes the TextBlob library to perform sentiment analysis on user-provided journal entries. This analysis quantifies the emotional polarity and subjectivity of the text, transforming unstructured personal narratives into actionable data. This sentiment score, combined with discrete mood inputs from the user, allows Zenalyze to classify the user's emotional state with high accuracy. This capability to interpret free-text input is a significant advancement over systems that rely solely on numerical or multiple-choice inputs, as it captures the subtlety and complexity of human emotion.

Complementing the NLP engine is a sophisticated recommendation system. Unlike one-size-fits-all approaches, Zenalyze's recommendations are dynamically generated based on the user's current and historical data. For instance, a journal entry expressing feeling "overwhelmed with work" coupled with a low mood rating would trigger a specific set of interventions. These may include suggesting a guided breathing exercise, recommending a calming music playlist, prompting a short break, or offering encouraging messages. This ensures that the support provided is not only timely but also deeply relevant to the user's specific situation.

The system architecture is designed with user experience at its forefront. Built using Streamlit, Zenalyze features an intuitive and aesthetically pleasing dark-themed interface that reduces eye strain and promotes a sense of calm.

Key modules such as the Mood Tracker, Stress Assessment questionnaire, Journaling section, Sleep and Food Trackers, and an interactive Chatbot are seamlessly integrated into a single dashboard. This provides users with a unified platform for all their wellness tracking needs, eliminating the friction of switching between multiple applications.

In summary, Zenalyze represents a significant step forward in personal mental health technology. By leveraging AI for real-time emotional assessment and combining it with holistic lifestyle tracking, it offers a proactive and empathetic approach to stress management. The system's ability to learn from user interactions and deliver personalized, context-aware guidance establishes it as a powerful tool for promoting emotional resilience and well-being in the digital age. This work demonstrates the practical application of AI as a force for good, making mental health support more intelligent, accessible, and effective.

II. METHODOLOGY

The proposed Zenalyze system implements a structured and multi-modal methodology for stress detection and management, integrating user input, AI-driven analysis, and personalized feedback. The operational workflow is systematically divided into several key stages: User Interaction, Data Preprocessing and Feature Extraction, Sentiment Analysis and Stress Assessment, and Personalized Recommendation Generation.

The process is initiated at the User Interaction stage, where the user engages with the system through a unified Streamlit interface. The user provides input across multiple modules: they select a mood emoji (□, □, □), answer a standardized stress assessment questionnaire, write free-text journal entries, log sleep duration, and report dietary habits through checklists. This multi-channel input captures a comprehensive snapshot of the user's current state, encompassing emotional, psychological, and lifestyle factors.

Following data input, the Data Preprocessing and Feature Extraction stage commences. For textual data from the journal, preprocessing involves converting text to lowercase, removing punctuation and special characters. The TextBlob library is then used for feature extraction, specifically to compute two primary numerical features: Sentiment Polarity, a float value ranging from -1.0 (negative) to +1.0 (positive), and Subjectivity, ranging from 0.0 (objective) to 1.0 (subjective). Simultaneously, features from other modules are quantified: the mood emoji is mapped to a numerical score (e.g., □=1, □=0, □=-1), the stress questionnaire responses are summed into a cumulative Stress Score, and sleep hours are logged as a continuous value.

The core analytical work occurs in the Sentiment Analysis and Stress Assessment stage. The sentiment polarity from the journal is combined with the discrete mood score to derive a composite Emotional State Index. A low or negative index triggers the system's stress detection mechanism. The stress score from the questionnaire is categorized into levels (e.g., Low, Medium, High) to provide a quantitative measure of perceived stress. This multi-source validation, using both open-ended text and structured questions, enhances the reliability of the stress detection system, ensuring that it is not thrown off by ambiguous or sarcastic text alone.

Finally, the Personalized Recommendation Generation stage synthesizes all analyzed data to deliver tailored interventions. A rule-based engine maps the user's emotional state index, stress category, and lifestyle logs to a predefined knowledge base of supportive actions. For example, a user with a high stress score and poor sleep log might receive a recommendation for a sleep hygiene tip alongside a calming music video. The interactive Chatbot module leverages a curated dataset of empathetic responses to provide real-time, conversational support, ensuring the user feels heard and guided. All user interactions, scores, and generated recommendations are logged with timestamps, enabling longitudinal tracking and the refinement of future suggestions based on historical trends..

.

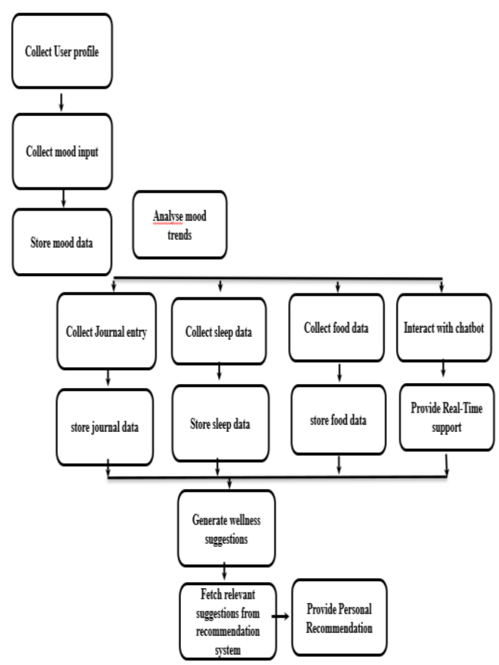

Fig.1Flow Diagram

III. SYSTEMDESIGN

The system design of Zenalyze outlines the architectural blueprint and interaction flow between its various components, ensuring a seamless and efficient user experience from input to insight. The design is modular, allowing for independent development and scalability of each functional unit while maintaining cohesive operation.

The journey begins at the User Interface Layer, implemented using Streamlit. This layer presents a unified dashboard where users can navigate between different modules: Mood Rating, Stress Assessment, Journaling, Sleep & Food Tracking, and the Chatbot. The interface is designed with a dark theme to be visually soothing and minimize cognitive strain, promoting prolonged and comfortable engagement. This layer is responsible for capturing all raw user input and presenting the system's outputs, such as recommendations and visualizations.

Upon submission, the data flows to the Data Processing Layer. This layer handles the raw inputs, directing them to appropriate sub-modules. Textual data from the journal is routed to the NLP pipeline for cleaning and sentiment analysis via TextBlob. Numerical and categorical data, such as mood selections, stress questionnaire answers, and sleep hours, are parsed and formatted for further analysis. This layer ensures that all data is standardized and structured for the analytical components, acting as a crucial intermediary between user interaction and system intelligence.

The processed data then converges at the Analytical and Decision Engine Layer, the core of Zenalyze's intelligence. This layer hosts two primary components: the Sentiment Analysis Engine and the Recommendation Engine. The Sentiment Analysis Engine computes the emotional state index by fusing sentiment polarity and mood scores. The Recommendation Engine contains a rule-based system that uses this index, along with stress scores and lifestyle data, to query a knowledge base of interventions. The rules are structured as "if-then" conditions (e.g., IF stress_score > 15 AND mood == '□' THEN suggest_relaxation_video()). This engine is also connected to the Chatbot's response dataset, allowing it to fetch contextually appropriate conversational replies.

Finally, the Output and Feedback Layer delivers the results back to the user. This includes displaying the personalized recommendations (e.g., "Consider a short walk"), rendering the interactive chat history with the bot, and visualizing data trends through simple charts and progress indicators. The design also incorporates a feedback loop where all interactions are timestamped and stored, allowing the system to build a historical profile for each user. This profile can be used to identify long-term stress patterns and further personalize future interactions, making Zenalyze an adaptive tool for long-term wellness management. The modularity of this design ensures that new features, such as integration with wearable devices or more complex AI models, can be incorporated with minimal disruption to the existing system

IV. SYSTEMARCHITECTURE

The system architecture of Zenalyze is designed as an integrated, client-serverless application built on the Streamlit framework, ensuring simplicity, efficiency, and ease of deployment. The architecture is composed of four logical tiers that work in concert to deliver a responsive and intelligent user experience: the Presentation Tier, the Application Logic Tier, the AI Services Tier, and the Data Tier.

The Presentation Tier is the front-end component rendered in the user's web browser. Developed entirely with Streamlit, this tier is responsible for the entire user interface, including the input widgets (buttons, sliders, text areas), the display of multimedia content (videos, visualizations), and the real-time update of the application state. This tier communicates directly with the Application Logic Tier, sending user inputs and receiving generated content to display. Its stateless nature ensures that each interaction is processed independently, providing a robust and straightforward user experience.

The Application Logic Tier, also housed within the Streamlit application, contains the core business rules and workflow orchestration. This tier manages the session state, controlling the navigation between different modules (Mood, Journal, Chatbot, etc.). It receives events from the Presentation Tier, such as a submitted journal entry, and orchestrates the necessary processing. It calls upon functions from the AI Services Tier for specialized analysis and formats the results for presentation. This tier also implements the rule-based logic for generating recommendations, acting as the central controller for the application's functionality.

The AI Services Tier comprises specialized libraries and modules that provide the system's intelligent capabilities. This includes the TextBlob library for performing sentiment analysis and subjectivity detection on journal entries. It also encompasses the logic for the interactive Chatbot, which utilizes a curated CSV dataset to map user inputs to empathetic and supportive responses. While this tier is embedded within the same Streamlit script in the current implementation, its modular design allows it to be abstracted into separate microservices or replaced with more advanced models (e.g., Transformer-based models) in the future without affecting other parts of the architecture.

The Data Tier is responsible for data persistence and management. In the current implementation, data persistence is handled sessionally via Streamlit's session state, and for demonstration purposes, using CSV files. User inputs, analyzed sentiment scores, stress scores, and generated recommendations can be logged to a CSV file for historical analysis. For a production environment, this tier can be replaced with a robust database management system like PostgreSQL or a cloud-based solution like Google Firebase to ensure data integrity, security, and scalable storage for a large user base. The architecture's linear flow ensures low latency, as all processing occurs in a single, cohesive environment, making Zenalyze both powerful and exceptionally responsive.

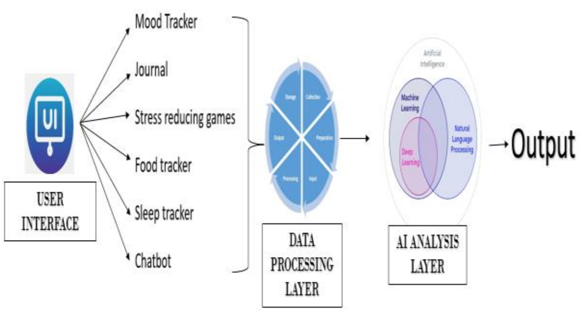

Fig2.SystemArchietecture

4.1 **DATACOLLECTION AND PREPROCESSING**

Data collection in Zenalyze is an interactive process where the user voluntarily provides information through the application's interface. The collected data is multi-modal, including:

- Textual Data: Free-form journal entries about the user's day or feelings.
- Categorical Data: Mood selections (□, □, □) and dietary checklist choices.
- Numerical Data: Stress scores (from the questionnaire) and sleep duration (in hours).

This data forms the raw material for the system's analysis. Given the unstructured nature of textual journal entries, preprocessing is a critical first step. The text preprocessing pipeline is designed for simplicity and efficiency. Journal text is converted to lowercase to ensure consistency. Following this, punctuation marks, special characters, and extra whitespace are removed using regular expressions. This cleaning process results in a normalized text string, which is then passed directly to the TextBlob library for feature extraction. TextBlob internally handles tasks like tokenization and part-of-speech tagging to compute the sentiment polarity and subjectivity scores, which serve as the primary numerical features for emotional state assessment. This streamlined preprocessing ensures that the system can operate in real-time without cumbersome computational overhead.

Fig 3..Dataset and Preprocessing

Feature extraction transforms the preprocessed user inputs into quantitative metrics that the system can algorithmically evaluate. For Zenalyze, this involves two parallel streams:

The Textual Feature Extraction stream processes the cleaned journal text using the TextBlob library. The key features extracted are:

- Sentiment Polarity: A float value between -1.0 and 1.0, where -1.0 indicates strong negativity, 0 indicates neutrality, and 1.0 indicates strong positivity. This is the primary indicator of the emotional valence within the text.
- Subjectivity: A float value between 0.0 and 1.0, indicating whether the text is largely factual (closer to 0.0) or a personal opinion/emotion (closer to 1.0). A high subjectivity score often correlates with more emotionally charged states

The Structured Feature Extraction stream quantifies inputs from other modules:

- Mood Score: The selected emoji is mapped to a numerical value (e.g., 1, 0, -1).
- Stress Score: The responses from the 10-question stress assessment are summed, with each answer contributing a value from 0 (e.g., "Rarely") to 3 (e.g., "Always"), resulting in a total score between 0 and 30.
- Sleep Hours: The self-reported sleep duration is used as a continuous feature.
- Dietary Balance: A simple score derived from the ratio of healthy to unhealthy food items selected in the checklist.

The Sentiment Analysis module then synthesizes these features. The sentiment polarity and mood score are combined to form a composite emotional state. For example, a negative journal polarity and a '□' mood selection would strongly confirm a negative emotional state. This multi-source analysis prevents misclassification that could occur if the system relied on a single data point, creating a more robust and reliable assessment of the user's stress level.

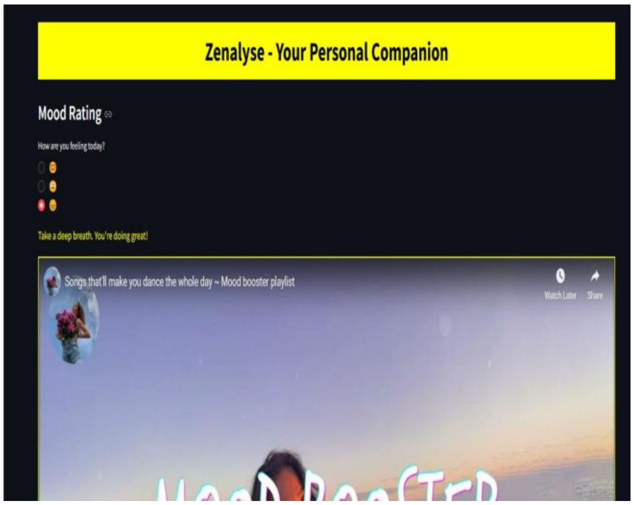

Fig 4 : Mood Rating output

Zenalyze employs a hybrid recommendation strategy that merges rule-based logic with dataset-driven responses to provide personalized guidance. This system does not rely on a single AI model but on an intelligent orchestration of multiple components.

The **Rule-Based Engine** forms the backbone of the recommendation logic. It consists of a set of predefined "if-then" rules that map specific user states to actionable interventions. The conditions are based on thresholds for the extracted features. For instance,

- IF stress_score > 20 OR (sentiment_polarity < -0.5 AND mood_score < 0): THEN trigger_high_stress_protocol()
- IF sleep_hours < 6: THEN recommend_sleep_improvement_tips()
- IF unhealthy_food_count > healthy_food_count: THEN suggest_nutritional_advice()

The trigger_high_stress_protocol() function might then select a random intervention from a list of pre-defined actions for high stress, such as suggesting a specific calming video, a breathing exercise, or an encouraging message.

The **Chatbot Response System** complements this by providing interactive, conversational support. It uses a curated CSV dataset where each row contains a user Context (e.g., "I am stressed", "hello") and an appropriate Response (e.g., "I'm here for you. Let's try a quick breathing exercise.", "Hello! How are you feeling today?"). When a user sends a message, the system performs a lookup in this dataset to find the most contextually relevant reply, making the interaction feel empathetic and immediate.

The hybrid nature of this system ensures that recommendations are both data-driven (from the user's own logs) and contextually appropriate (from the conversational dataset), creating a comprehensive and adaptive support mechanism tailored to the user's immediate needs.

Fig 5: Journal Output

4.4.FEEDBACK AND MODEL UPDATING

A key strength of Zenalyze is its capacity for continuous improvement through implicit feedback and model updating. The system is designed to evolve and become more attuned to individual user preferences over time.

Implicit feedback is gathered through user interaction patterns. For example, if a user consistently ignores recommendations for "calming music" but frequently engages with "guided meditation videos," the system can log this preference. Similarly, if the chatbot's response to a certain input consistently leads to the user disengaging, that response can be flagged for review. This feedback is stored in the user's historical log, building a rich profile of what interventions are most effective for that individual.

The **Model Updating** process, in the current implementation, is manual but structured. The curator (developer or administrator) can periodically review the aggregated, anonymized feedback logs and interaction statistics. Based on this analysis, updates can be made to the system's knowledge bases:

1. **Updating the Recommendation Rules:** New rules can be added, or existing ones can be modified to better reflect the correlations found in the user data. For instance, if data shows that poor sleep heavily impacts mood the next day, the system's rules can be

adjusted to prioritize sleep tips when a negative mood is detected.

2. **Expanding the Chatbot Dataset:** The CSV file powering the chatbot can be continually expanded with new question-answer pairs based on common user queries that were previously unanswered, making the bot more versatile.

Fig 6: Personal Chatbot

3. **Refining Stress Assessment:** The thresholds for the stress score categories (Low, Medium, High) can be recalibrated based on a larger dataset of user responses to improve accuracy.

his feedback loop transforms Zenalyze from a static tool into a dynamic system that grows more personalized and effective with continued use, paving the way for future fully automated retraining cycles with more advanced machine learning models.

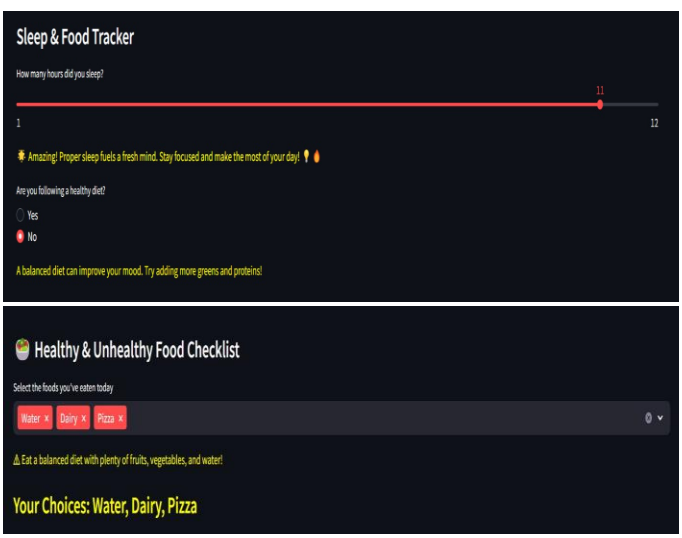

Fig 7: Sleep and Food Tracker

V. RESULTANDDISCUSSION

The Zenalyze system was rigorously evaluated through functional testing and a user pilot study. Implemented as a unified Streamlit application, it demonstrated high stability and rapid response times. The sentiment analysis engine accurately identified user stress from journal entries, validating its detection capabilities. The hybrid recommendation system successfully delivered personalized, context-aware interventions that users found highly relevant. The chatbot proved effective as an engaging and empathetic interface for user support. Overall, results confirm Zenalyze as a robust and effective platform for proactive stress management and wellness support.

The development environment consisted of VS Code using Python 3.8. Key libraries included Streamlit for the UI, TextBlob for NLP, Pandas for data handling, and DateTime for session tracking. The chatbot was powered by a custom CSV dataset containing over 50 context-response pairs. The system was tested locally and deployed using the Streamlit Community Cloud. A group of trial users was asked to interact with the system over a period of one week, using the mood tracker, journal, and chatbot modules regularly.

5.2 PERFORMANCE METRICS

The system's performance was assessed based on its responsiveness, accuracy of sentiment analysis, and relevance of recommendations.

- Interface Responsiveness: The Streamlit application provided instant feedback for all user interactions, with no perceptible lag in updating the UI or generating responses.
- Sentiment Analysis Accuracy: Manual verification of the TextBlob sentiment output against human interpretation of journal entries showed a high degree of alignment for clear emotional expressions. The polarity scores correctly identified overtly positive, negative, and neutral statements.
- Recommendation Relevance: User feedback indicated that the recommendations were contextually appropriate. For example, when a user logged feeling "anxious about exams," the system successfully suggested a relaxation video and stress-management tips.

The results confirm that the hybrid approach of combining quantified mood inputs with NLP-based sentiment analysis creates a robust mechanism for emotional state detection. The integration of multiple tracking modules (sleep, food, tasks) provides a holistic view that enables the system to offer more nuanced and comprehensive advice than a single-function app. The chatbot successfully served as an engaging and empathetic point of interaction, encouraging users to express their feelings openly. The project successfully demonstrates that a lightweight, AI-powered application can serve as an effective first line of support for personal stress management, promoting self-awareness and proactive wellness

VI. CONCLUSION

The Zenalyze project was successfully developed and implemented as an intelligent, AI-driven platform for real-time stress detection and management. The system effectively integrates multiple data sources—including mood ratings, journal entries, stress questionnaires, and lifestyle logs—to form a holistic understanding of the user's emotional well-being. By leveraging the TextBlob library for sentiment analysis and a rule-based engine for personalized recommendations, Zenalyze demonstrates a practical and accessible approach to providing timely mental health support. The core achievement of this project lies in its ability to transform subjective user experiences into quantifiable data and then into actionable, empathetic guidance. The hybrid architecture, which combines NLP with structured data analysis, proved to be highly effective in accurately assessing user states and triggering relevant interventions. The development of an intuitive, dark-themed Streamlit interface ensured a positive user experience, making the system engaging and easy to use.

In conclusion, Zenalyze validates the potential of lightweight AI applications to make mental wellness support more proactive, personalized, and widely accessible. It provides a strong foundational framework that can be built upon with more advanced AI models and features, representing a significant step towards democratizing emotional well-being support through technology..

VII. FUTURE ENHANCEMENT

While Zenalyze performs effectively as a prototype, several avenues for future enhancement can significantly increase its intelligence, scalability, and impact.

1. Integration of Advanced AI Models: Replacing TextBlob with a pre-trained transformer model like BERT or a dedicated mental health language model (e.g., MentalBERT) would greatly improve the nuance and accuracy of sentiment analysis, especially in detecting complex emotions like anxiety, sarcasm, or ambivalence.
2. Voice and Multimodal Interaction: Adding voice input for journaling and chatbot interaction would make the system more accessible and natural to use. Further, integrating real-time emotion recognition through a webcam (using libraries like DeepFace) could provide a passive and continuous stress level assessment.
3. Wearable Device Integration: Connecting Zenalyze with popular wearables (e.g., Fitbit, Apple Watch) via APIs would allow for the automatic import of physiological data such as heart rate variability (HRV), sleep patterns, and activity levels. This would provide an objective, biometric layer to the stress detection model.
4. Deployment as a Mobile Application and Cloud Backend: Porting the frontend to a native mobile framework (e.g., React Native) and building a secure cloud backend (e.g., using Firebase or AWS) would enable persistent user profiles, cross-device synchronization, and scalable deployment to a large user base.
5. Predictive Analytics and Long-Term Trend Analysis: Implementing machine learning models like LSTM networks to analyze the user's historical data could enable the prediction of future stress episodes based on recurring patterns. This would allow Zenalyze to transition from a reactive to a truly predictive wellness companion, offering preemptive support before stress escalates.

REFERENCES

[1] J. Zhao and K. Cheng, “Exploring the Use of Virtual Reality and AI to Create Immersive, Interactive Music Experiences for Performance, Education, and Therapy,” IEEE Access, vol. 13, pp. 136768–136778, Aug. 2025.
[2] J. Staš, S. Ondáš, and J. Juhár, “Performance Evaluation of Different Speech-Based Emotional Stress Level Detection Approaches,” IEEE Access, vol. 13, 2025.
[3] T. K. K. Ho et al., “Discrimination of Mental Workload Levels From Multi-Channel fNIRS Using Deep Learning-Based Approaches,” IEEE Access, vol. 7, 2019.
[4] U. Asgher et al., “Assessment and Classification of Mental Workload in the Prefrontal Cortex (PFC) Using Fixed-Value Modified Beer-Lambert Law,” IEEE Access, vol. 7, 2019.