Daily user emotion Data Collection, Voice, and Data Integration Mental Health System

24-25J-293

De Alwis K.C.

(IT21306204)

B.Sc. (Hons) Degree in Information Technology Specializing in Information Technology

Department of Information Technology
Sri Lanka Institute of Information Technology
Sri Lanka

April 2025

Daily user emotion Data Collection, Voice, and Data Integration Mental Health System

24-25J-293

De Alwis K.C.

(IT21306204)

Dissertation submitted in partial fulfillment of the requirements for the Bachelor of Science (Hons) Degree in Information Technology Specializing in Information Technology

Department of Information Technology
Sri Lanka Institute of Information Technology
Sri Lanka

April 2025

DECLARATION

I hereby declare that this is my own work, and this dissertation does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any other university or institute of higher learning, and to the best of my knowledge and belief, it does not contain any material previously published or written by another person except where due acknowledgment is made in the text.

I also hereby grant to Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute my dissertation, in whole or in part, in print, electronic or other medium. I retain the right to use this content in whole or in part in future works (such as articles or books).

Name	Student ID	Signature
De Alwis K.C.	IT21306204	

The above candidate has carried out research for the bachelor's degree dissertation under my supervision.

Signature of the supervisor: **Date:**

ABSTRACT

In today's fast paced digital society, mental health concerns have become increasingly pronounced, impacting individuals of all demographics. One of the emerging contributors to these issues is the overuse of digital platforms, especially mobile apps and web-based services.

This project addresses that concern through a novel system that combines machine learning and multimodal data collection to deliver predictive mental health support. The core functionality revolves around real-time emotion data gathering from voice tone analysis, facial expression detection, and biometric feedback, all seamlessly integrated through a mobile application and a web extension.

The system proactively recommends interventions, including therapeutic games, AI-powered chatbots, and custom wellness suggestions, to improve user well being. Privacy and ethical data usage remain foundational to the design, ensuring secure and responsible management of sensitive user data.

This dissertation explores the technical implementation of the emotion detection modules, real time processing of audio and visual cues, data integration strategies, and the innovative personalization engine that adapts to user emotional states.

Keywords: *Emotion Recognition, Mental Health Monitoring, Voice Analysis, Multimodal Data Integration, Machine Learning*

ACKNOWLEDGEMENT

I would like to extend my heartfelt appreciation to my Supervisor, **Ms. Thilini Jayalath**, and Co-Supervisor, **Mr. Deemantha Siriwardana**, for their *invaluable guidance* and support throughout the project.

A special thank you to our external supervisors, **Mrs. Shalindi Pandithakoralage** and **Mrs. Senethra Sachini Pathiraja**, for their *insightful feedback* and encouragement during the critical phases of development.

I am also grateful to **Sri Lanka Institute of Information Technology (S.L.I.I.T)** for providing access to essential resources and facilities.

Last but not least, sincere thanks to my project team members, friends, and family for their patience, motivation, and support during this journey.

Table of Contents

DECLARATION iii
ABSTRACT iv

Keywords: Emotion Recognition, Mental Health Monitoring, Voice Analysis, Multimodal Data Integration, Machine Learning iv

ACKNOWLEDGEMENT v

Last but not least, sincere thanks to my project team members, friends, and family for their patience, motivation, and support during this journey. . v

List of Tables ..... **Error! Bookmark not defined.**
List of Figures ..... **Error! Bookmark not defined.**
INTRODUCTION 10

Background Literature 10

Research Gap 18

Research Problem 21

Research Objectives 21

METHODOLOGY 25

Methodology 25

System architecture overview 25

Voice Emotion Analysis Pipeline 28

2.1.2.1 Library Installation and Dataset Acquisition 28
2.1.2.2 Dataset Preparation and Labeling 29
2.1.2.3 Class Distribution Visualization 31
2.1.2.4 Audio Feature Extraction 32
2.1.2.5 Encoding and Preprocessing 33
2.1.2.6 Addressing Class Imbalance with SMOTE 34
2.1.2.7 Model Architecture and Training 34
2.1.2.8 Evaluation and Accuracy Metrics 35

Workflow of daily emotion data collection 36

Tools & technologies used 41

Commercialization Strategy 43

Free version for individuals 43

Premium plan for organizations **44**
Tiered pricing model **45**
Feature comparison table **45**
Testing and implementation ***46***
Manual testing of machine learning model **46**
MongoDB Emotion Storage Verification **47**
API Testing Using Postman **48**
Other Testing Methods **50**
Deployment Preparation and Observations **50**
RESULTS & DISCUSSION 51
Results ***51***
Voice Emotion Detection Output **51**
MongoDB Voice Recording Logging **52**
Daily Form API Submission **52**
Emotion Dataset Distribution **53**
Research Findings ***53***
Discussion ***54***
Voice Model Reliability **54**
User Interaction and Engagement **54**
Data Pipeline Effectiveness **54**
Summary ***55***
CONCLUSION AND RECOMMENDATIONS 55
Conclusion ***55***
Recommendations ***56***
REFERENCES 58
GLOSSARY 60
APPENDIX A: Survey Questionnaire ***61***
Section 1: Demographics **61**
APPENDIX B: Survey Results ***64***

LIST OF FIGURES

Figure 1Age Distribution 11
Figure 2 Gender Distribution 11
Figure 3 Stress Levels 12
Figure 4 Frequency of Overwhelm 12
Figure 5 Sources of Stress 13
Figure 6 Coping Mechanisms 14
Figure 7 Mood Assessment 15
Figure 8 Anxiety Frequency 15
Figure 9 Sleep Patterns 16
Figure 10 Social Support 17
Figure 11 Physical Activity 17
Figure 12 Screen Time 18
Figure 13 System architecture 25
Figure 14 Library Installation 28
Figure 15 Dataset Download 29
Figure 16 Assign dataset directories 30
Figure 17 Import pandas 30
Figure 18 Dataset Directory Assignment – RAVDESS 30
Figure 19 Dataset Directory Assignment - CREMA - D 30
Figure 20 Dataset Directory Assignment - TESS 31
Figure 21 Dataset Directory Assignment - SAVEE 31
Figure 22 Class Distribution Before Balancing 32
Figure 23 Audio Feature Extraction Code – Librosa 33
Figure 24 One-Hot Encoding and Train-Test Split 33
Figure 25 Class Distribution After SMOTE 34
Figure 26 Keras Model Architecture – Sequential Layers 35
Figure 27 Accuracy Trends – Model Training vs Validation 35
Figure 28 Daily Mood Input UI 37
Figure 29 Filled Daily Mood Input UI 38
Figure 30 Submitted Daily Mood Input UI 39
Figure 31 Emotion Entry in MongoDB Atlas 40
Figure 32 test_model_script_prediction 47
Figure 33 mongodb_voice_recordings_entry 48
Figure 34 postman_form_submit_success 49

Figure 35 postman_form_get_success 49

LIST OF TABLE

Table 1 tiered pricing plan..... 45
Table 2 Feature comparison table 46

INTRODUCTION

Background Literature

In today's Digitally evolving society, mental health has become an ever-growing concern that is intricately linked to technology use. With smartphones, social media, remote learning, and digital workspaces now fully integrated into our Everyday routines, the psychological effects of these tools cannot be underestimated. As Digital evolution rapidly reshapes how we live and interact, it has inadvertently introduced new sources of emotional fatigue, isolation, and stress. This shift has prompted mental health researchers to explore new avenues for Tools for digital mental health [1] [2].

Mental health is no longer confined to clinical boundaries or extreme cases. Instead, it touches the lives of A significant proportion of tech users at varying degrees ranging from occasional stress and low mood to prolonged anxiety and depression. This pervasiveness is reflected in modern behavioral research, where emotional distress is now tracked via non-invasive indicators, such as facial expressions, heart rate variability, and voice tone [3] [4].

Voice in particular stands out as a powerful biomarker for emotion detection. Emotional nuances such as sadness, anger, calmness, and frustration are frequently expressed through speech tone, rhythm, pitch, and inflection. Advanced AI systems have begun leveraging machine learning (ML) models to process speech data and derive emotional states with considerable accuracy [2], [5]. Rehman et al. [1] demonstrated that voice-based machine learning algorithms could effectively detect vocal disorders indicating a high potential for mental health assessments as well. Likewise, Tokuno’s research emphasized the role of voice as an indicator of stress and mood fluctuation [2].

Nevertheless, despite the development of emotion recognition systems, most tools function in siloed environments. For example, a mobile application might analyze only

voice inputs, ignoring facial expressions, biometric signals, or contextual digital behavior [6].This lack of integration leads to partial assessments and reduces the system's capacity for holistic emotion profiling.

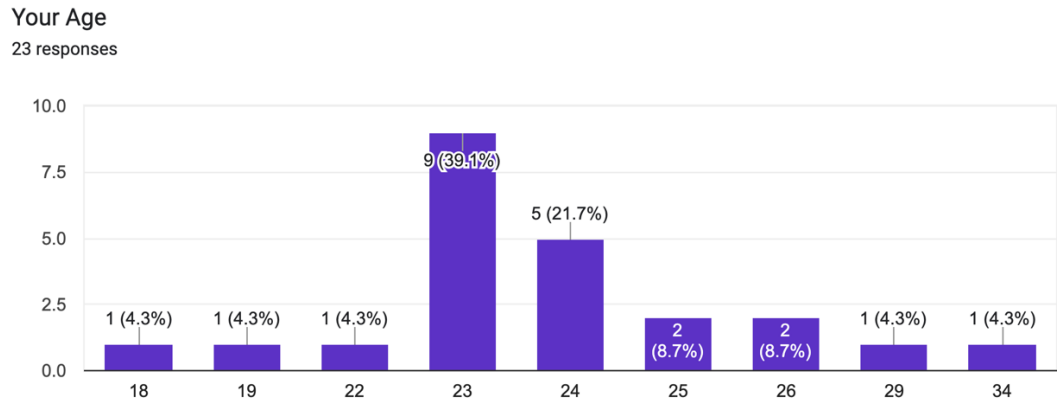


Figure 1Age Distribution

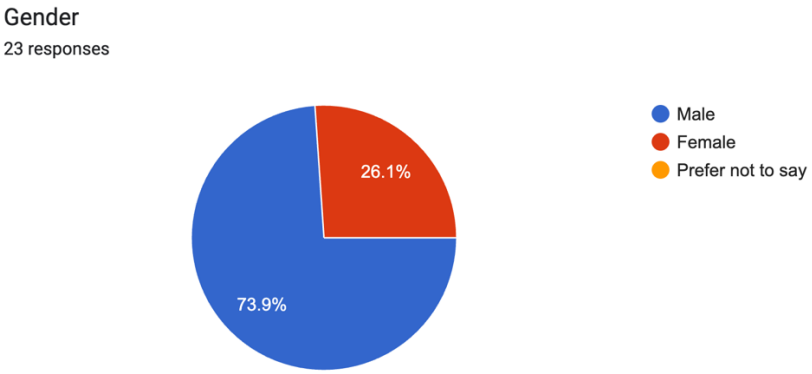


Figure 2 Gender Distribution

Understanding the demographic of participants is critical to contextualizing emotional patterns. In this study, 23 respondents provided insights into their mental well-being. Out of this group, 73.9% were male, while 26.1% were female. This skew may reflect broader trends in tech adoption and participation in mental health research where male students and young professionals often dominate digital platforms and are also less likely to seek traditional mental health services.

From an age perspective, a majority fell between 22 and 24 years old, the prime age range for university students and early-career individuals. This group is frequently associated with transitional life stressors such as job searching, academic deadlines, and emerging adult responsibilities. These demographic findings support the selection of the target audience for this system, ensuring the design is tailored to those at highest digital engagement and emotional risk .

Furthermore, such age clusters are more responsive to gamified solutions and app based health tracking, increasing the feasibility of a mobile-first platform. Hence, these demographic characteristics align with both the technological delivery and psychological goals of the proposed system.

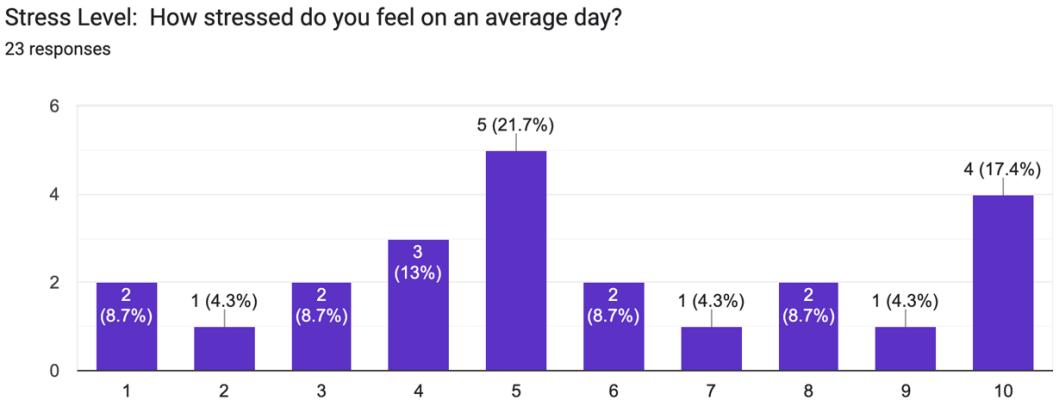


Figure 3 Stress Levels

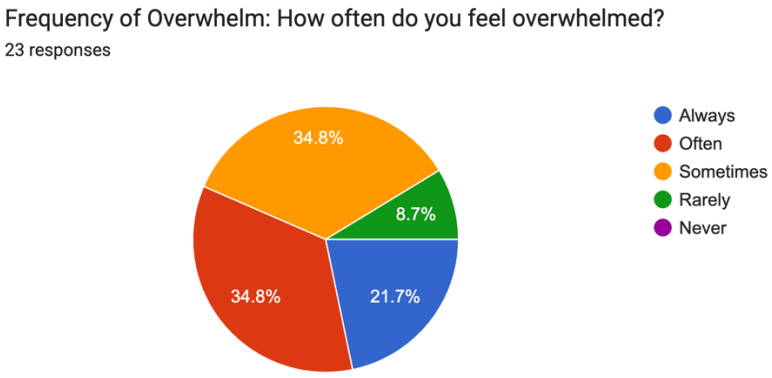


Figure 4 Frequency of Overwhelm

When asked to self-assess their stress levels on a scale from 1 to 10, responses clustered in the moderate to high range, with 21.7% selecting level 5, and 17.4% selecting level 10 indicative of acute stress. Additionally, 34.8% often felt overwhelmed, and another 34.8% sometimes did, suggesting a widespread experience of cognitive overload and emotional exhaustion.

These insights validate the need for early detection and support. Users facing frequent overwhelm are more likely to ignore traditional therapy paths due to fatigue or stigma. A system that integrates passive indicators like voice tone or typing speed can gently flag emotional strain without requiring the user to initiate help seeking. This passive detection followed by proactive intervention could bridge the gap between silent suffering and timely support [4], [7].

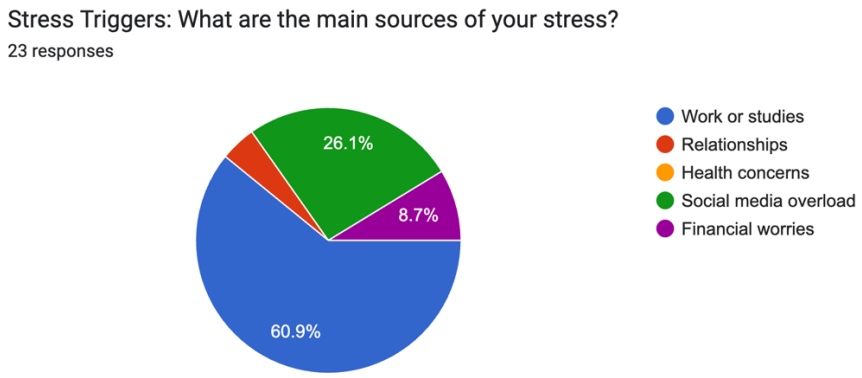


Figure 5 Sources of Stress

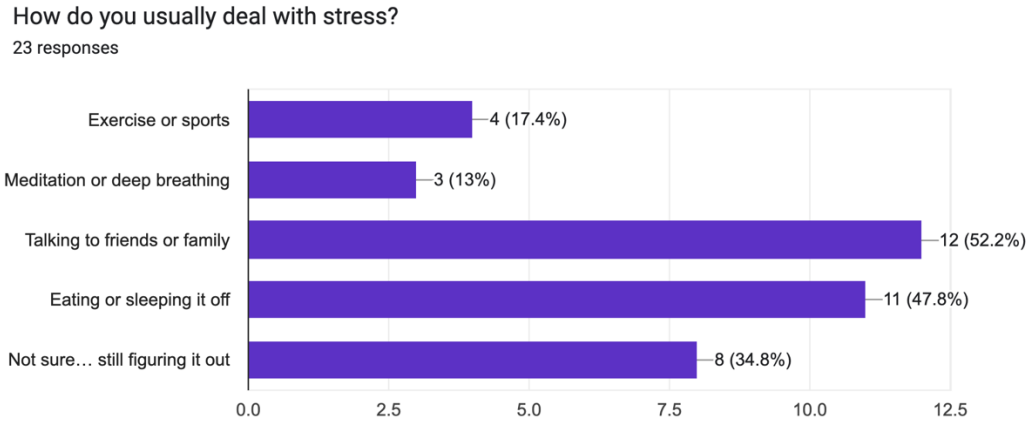


Figure 6 Coping Mechanisms

The survey revealed that 60.9% identified academic or professional workload as their primary source of stress. Relationship issues followed at 26.1%, and remaining respondents cited health concerns, screen fatigue, and financial instability. Notably, 34.8% had no structured way of coping, while 47.8% withdrew into sleeping or eating coping strategies that temporarily suppress symptoms but do not promote healing.

This supports the design of a "reverse Blue Whale" intervention loop, where users are encouraged to complete helpful, bite-sized tasks that reinforce long-term emotional regulation. By tapping into behavioral psychology principles such as micro-rewards, streaks, and visual progress bars the system nudges users toward sustainable mental hygiene.

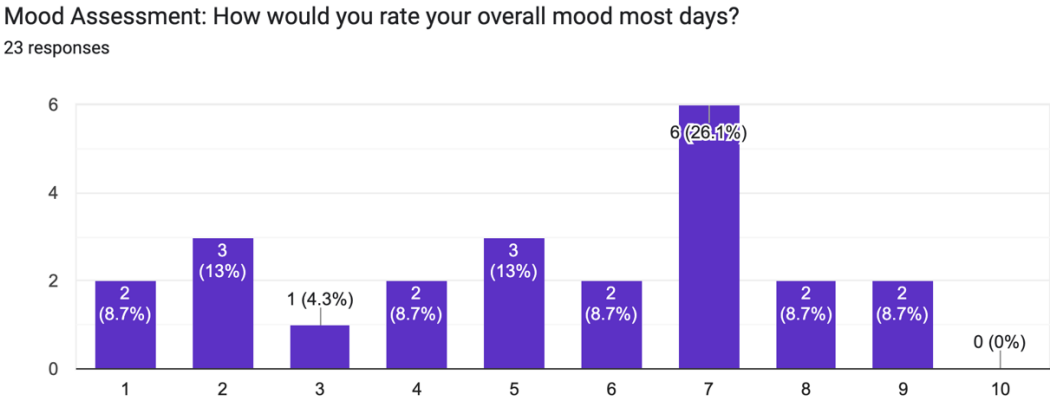


Figure 7 Mood Assessment

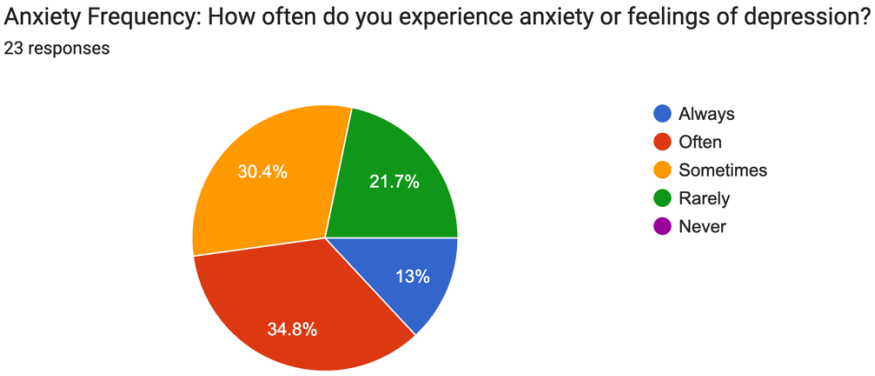


Figure 8 Anxiety Frequency

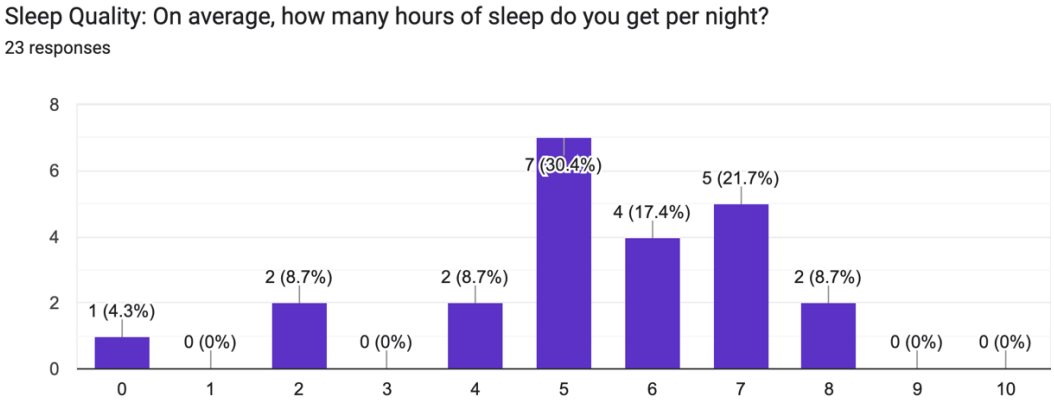


Figure 9 Sleep Patterns

Mood distribution responses showed a spread from 3/10 to 8/10, with 7/10 being the most common. This suggests a mix of underlying positivity and episodic downturns. However, when paired with anxiety reports 34.8% often experiencing anxiety or depression the data underscores that many participants mask inner struggles with outward functionality.

Additionally, sleep deprivation was rampant, with 30.4% sleeping just 5 ,6 hours nightly. Poor sleep is both a cause and symptom of psychological strain, making it a key variable in this system's health engine. By tracking sleep inputs and correlating them with speech patterns and emotional inputs, the system can make personalized suggestions like recommending shorter screen time or redirecting users to breathing exercises.

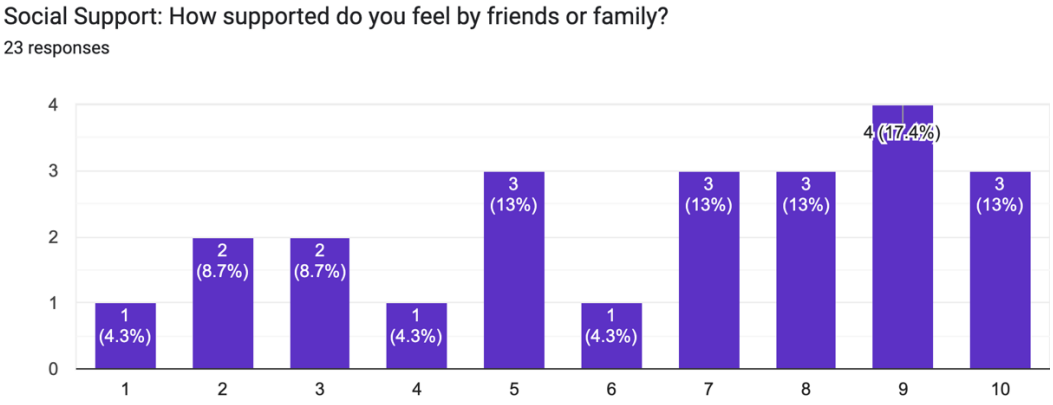


Figure 10 Social Support

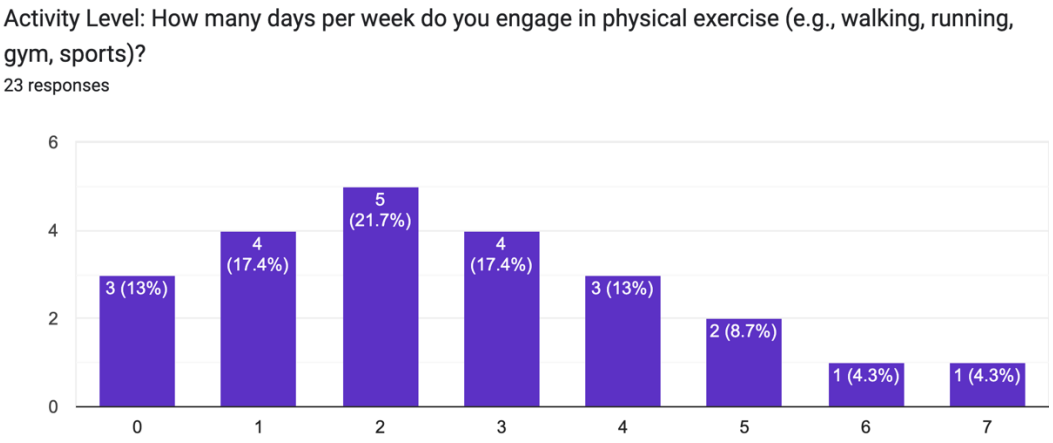


Figure 11 Physical Activity

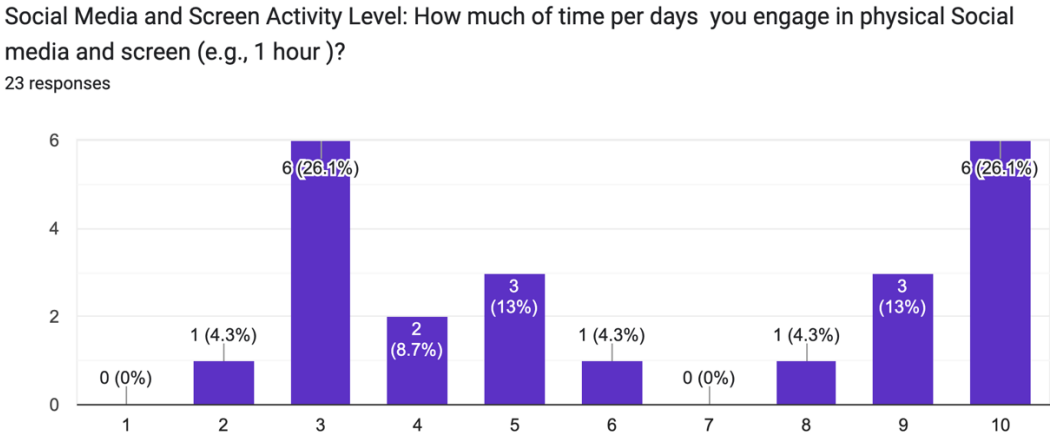


Figure 12 Screen Time

Social support is a known resilience factor. Yet, only half of the respondents felt they could turn to someone regularly, while physical activity was limited to 0 - 2 days/week for most. Combined with high screen time (26.1% over 3 hours/day), these results point to a sedentary, isolated lifestyle that deteriorates psychological resilience.

This makes it vital for our system to simulate "digital companionship" through AI-driven chatbots and feedback loops. Even small nudges like a voice assistant saying, "You've been active for an hour, how are you feeling now?" can offer a sense of being seen and heard, reducing psychological isolation.

Research Gap

Although significant advancements have been made in the field of digital mental health, there remain persistent and critical gaps that reduce the effectiveness and scalability of current systems. These gaps exist both in research environments and commercial implementations, particularly in how emotional data is captured, interpreted, and acted upon.

gap 1: limited data integration

Most existing systems either rely on manual mood tracking or focus on a single data source, such as heart rate, voice, or user journaling. For instance, an app may track sleep patterns using smartwatch sensors but ignore voice tone changes or facial expression data. Likewise, some ai-based tools detect vocal stress but do not consider screen usage patterns or biometric variations.

This fragmented data capture model leads to shallow insights. As a result, emotional health assessments are incomplete and inconsistent, which makes them unsuitable for real-world emotional support where multiple factors influence mental states simultaneously [1], [4].

The survey conducted during this project revealed similar gaps: although 82.6% of respondents said mental health affected their daily lives, very few reported using more than one method to track their well-being. This highlights the need for a system that can consolidate multiple emotional cues into one cohesive mental health assessment.

gap 2: absence of real-time monitoring

Time delay in emotional tracking is one of the biggest weaknesses in current digital wellness solutions. Most systems operate on retrospective data meaning that they assess a user's mental state only after the fact. This model is reactive rather than proactive and cannot support users in emotionally charged moments.

Real-time analysis of voice tone, typing speed, and facial micro-expressions could offer instant insights, allowing the system to intervene when a user is in distress. Yet, very few systems provide this capability due to computational limitations, privacy concerns, or lack of multimodal synchronization.

Studies by elsayed [4] and higuchi [7] emphasize that emotion recognition systems must evolve beyond delayed logging and move toward continuous, passive, and live analysis. Without real-time response, users may lose trust in the system or stop engaging altogether.

gap 3: inadequate user engagement models

Sustained engagement with mental health tools is often low, especially among users who are already fatigued or overwhelmed. Existing systems either require too much

manual input or offer generic, repetitive suggestions that do not adapt to changing emotional states. As a result, users abandon the tool after a few days or weeks, which negates its long-term benefit.

The lack of dynamic interfaces that prompt daily emotion logging, adjust interface difficulty, or personalize recommendations contributes to this problem. Omiya et al. [8]suggest that tools which feel impersonal or robotic often worsen emotional disconnect, rather than easing it.

From our survey, 34.8% of users had no coping mechanisms at all. This population is unlikely to stick with tools that demand high effort without offering instant, tangible relief. The research system must therefore include behavior-aware engagement techniques that feel supportive rather than burdensome.

gap 4: fusion of multimodal emotional data is underdeveloped

Emotion is multidimensional it manifests in voice, posture, expression, behavior, and even screen scrolling speed. However, very few platforms offer the ability to analyze, weight, and combine these various signals in real time to generate a single emotional state.

Most existing systems provide separate reports for each input modality (e.g., one graph for heart rate, one for mood logs), forcing users or therapists to interpret the connections manually. This introduces bias and complexity, especially for non-experts.

Perelli et al. [8]and koudounas et al. [9]point out that ensemble fusion models, which combine data streams using confidence weighting and deep neural integration, are the future of emotional ai. Yet, such models are rarely deployed in mobile health environments due to lack of robust datasets and training pipelines.

gap 5: data security and ethical constraints

Handling emotional and biometric data comes with significant ethical responsibility. Many mental health applications fail to address common security vulnerabilities, such as unencrypted data storage, improper session handling, or insecure third-party integrations.

User data must be protected not just from external breaches but also from misuse within the platform such as emotion data being used to push manipulative advertising or content. A truly ethical system must incorporate security measures by design, aligning with OWASP standards and GDPR principles.

Only a handful of studies, such as Tokuno et al. [2], have addressed the intersection of emotional tracking and privacy-preserving system architecture. This is a crucial area that needs further research and implementation.

Research Problem

General problem

There is a widespread lack of intelligent, real-time, integrated systems for proactive mental health support. With the average person spending several hours daily on digital platforms, massive amounts of behavioral data go unanalyzed. Most applications only respond after symptoms appear resulting in reactive mental health management instead of preventive care.

Voice detection–specific problem

Although studies have shown that voice-based emotion recognition is promising, it is often limited by narrow training datasets, low accuracy in multilingual settings, and absence of multimodal synergy [2], [3], [5]. Current systems often miss key contextual cues, leading to incomplete or inaccurate emotional assessments.

Moreover, existing tools do not personalize recommendations based on combined insights from voice, keyboard use, screen activity, and mood logs. This leaves a gap in the ability to predict crises before they escalate [4], [6], [10].

Research Objectives

To address the challenges highlighted above, this research proposes a holistic, AI-enhanced system that unifies emotional signals across multiple input streams and delivers personalized, real time mental health interventions.

Main Objective

The overarching aim of this project is to design a comprehensive, intelligent, and secure mental health monitoring system that combines real-time data analysis, personalized intervention, and continuous emotional profiling. This system will synthesize various modalities such as voice tone, keyboard dynamics, facial expression, daily mood input, and biometric indicators to assess, predict, and support emotional health.

Rather than waiting for users to signal distress, this system will predict emotional downturns based on longitudinal data patterns and suggest interventions before a crisis point is reached. By incorporating voice emotion recognition, the solution expands beyond text-based check ins and becomes accessible to users with low literacy, visual impairments, or expressive difficulties.

Sub-Objective 1: Daily Mood Check-ins

This feature encourages users to self-assess and record their mood at regular intervals using emojis, sliders, or even short voice notes. By standardizing daily entries, the platform builds a temporal profile of emotional fluctuation that is both user generated and context aware.

Moreover, the interface will adapt dynamically. If the user logs a low mood three days in a row, the system will reduce complexity in prompts and instead offer short wellness actions to avoid cognitive overload. This makes emotional check-ins low-friction and personalized.

Sub-objective 2: voice emotion recognition engine

Using ai and acoustic feature extraction, this module will interpret speech elements such as pitch, mfccs, prosody, and jitter to determine emotional state. Voice input will

be processed using deep recurrent neural networks (rnns) or transformers to detect subtle emotional shifts [1], [4].

This feature also includes language model adaptation to account for different dialects or accents. Through passive analysis (e.g., during journaling or voice search), the engine works silently in the background, only surfacing when needed.

Sub-objective 3: facial emotion module

Using convolutional neural networks (cnns), the camera can track micro-expressions and facial symmetry. Facial emotion recognition enhances the reliability of the system by offering non verbal cues that complement speech.

Data from facial analysis (e.g., eyebrow elevation, pupil dilation, or lip curvature) will be fused with other emotional signals, helping the system distinguish between "happy sounding" but visibly upset users something especially useful for neurodiverse populations.

Sub-objective 4: keyboard and screen usage monitoring

This module detects typing speed, key pressure, and error rates. Studies show that anxious or distracted users tend to type faster but make more mistakes, while depressed individuals type slower with long pauses [5], [11].

Additionally, screen usage duration and app-switching frequency will help determine mental fatigue. The system will be designed to work passively, gathering behavioral context without interrupting the user experience.

Sub-objective 5: data fusion and mental state engine

All input streams (voice, face, typing, mood check-ins, and biometrics) will be processed in a multimodal fusion engine that creates a real time emotional state profile. This engine uses weighted confidence scoring and ensemble modeling to synthesize diverse data types into one reliable emotional label.

The result is a dynamic "mood score" updated continuously, allowing the system to anticipate when the user is heading toward emotional exhaustion or instability.

Sub-Objective 6: Predictive Crisis Detection

By tracking longitudinal trends, the system will forecast potential emotional breakdowns. For instance, if a user sleeps poorly for three nights, logs declining mood, and exhibits frustrated typing patterns, the system may predict an upcoming emotional crash.

At this stage, the system doesn't just offer suggestions it might even lock distracting apps, trigger an urgent notification, or suggest reaching out to a friend or counselor based on the user's contact list or settings.

Sub-Objective 7: Personalized Recommendation Engine

Recommendations will include AI-generated therapy prompts, music playlists, meditation videos, and emotion-specific articles. These will be tailored using collaborative filtering and behavioral similarity modeling, ensuring they are relevant and useful.

The engine also learns what works over time if a user consistently engages with calming sounds after anger detection, it will recommend those earlier in the future.

Sub-Objective 8: Gamified Therapeutic Engine

Inspired by the "reverse Blue Whale" concept, the app will guide users through a 30 day emotional wellness journey, with each day presenting a small but healing challenge like smiling in a mirror, sending a kind message, or drinking enough water.

These activities, though simple, are backed by cognitive behavioral therapy (CBT) micro-interventions, creating structure, rewards, and purpose.

Sub-Objective 9: Security and Privacy Layer

Sensitive emotional data requires rigorous protection. The system will implement end to end encryption, biometric login, and anonymized data models. It will also comply with GDPR-like standards and address five key risks from OWASP's Top 10 mobile vulnerabilities ensuring that users can trust the platform with their emotional footprint.

METHODOLOGY

Methodology

System architecture overview

The system architecture is designed to support a robust, multi layered, and real time predictive mental health monitoring system. This comprehensive design ensures a seamless flow of user data, from initial collection to advanced machine learning analysis and actionable insights. The architecture is divided into multiple layers, each responsible for specific roles while operating under a centralized Security Layer to guarantee data privacy, encryption, and user authentication. The architecture enhances both the daily user emotion input and voice emotion recognition pipeline functionalities.

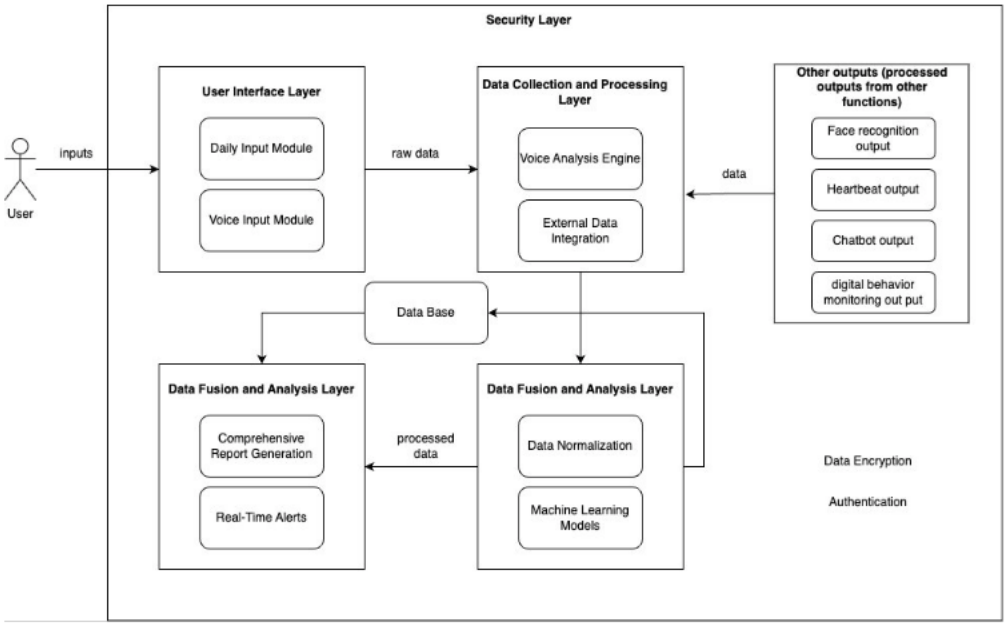


Figure 13 System architecture

User interface layer

The user interface layer is the primary point of interaction between the user and the system. It is composed of two modules. The first is the Daily Input Module, which allows users to submit daily reflections on their emotional well-being using a form-based user interface. This form includes emoji selections representing mood states, sliders to assess levels of stress, motivation, and sleep quality, and a submission button. The second module is the Voice Input Module, which captures spoken responses or spontaneous audio entries from the user. These voice samples are subsequently sent to the machine learning-based voice recognition pipeline for further analysis. Both modules are built using Flutter and communicate with the backend Node.js server through RESTful APIs.

Data collection and processing layer

The data collection and processing layer is responsible for converting raw inputs into structured and analyzable formats. One of its core components is the Voice Analysis Engine, which channels incoming voice recordings to the ML model hosted on the Flask server. This engine is responsible for audio cleaning, feature extraction (such as MFCC), and ensuring consistent formatting. Another integral feature of this layer is External Data Integration, which incorporates data from supplementary modules including facial recognition, heartbeat monitors, and chatbot interaction logs. This enables a more enriched and holistic view of the user's mental and emotional state. The layer functions as a transformation hub to standardize input data for effective storage and analysis.

Data fusion and analysis layer

The data fusion and analysis layer is dedicated to consolidating inputs from all sources and deriving meaningful insights. This layer consists of two main segments. The first segment involves data normalization and machine learning model execution. Data normalization techniques such as scaling and encoding are applied to clean and prepare the data. The machine learning models ranging from CNNs for voice analysis to regression and classification algorithms are executed via Flask API endpoints.

The second segment includes real time alerts and report generation. Real time alerts are triggered when indicators of elevated stress or depressive symptoms are detected, allowing for timely intervention. Additionally, the system compiles comprehensive reports that include both daily entries and historical data, providing users and professionals with an overview of mental health patterns over time. This layer forms the analytical and predictive intelligence of the platform.

Other output modules

This section includes modules developed by various contributors that provide additional data streams to enhance prediction accuracy. These modules process outputs from face recognition systems, heartbeat monitoring sensors, chatbot interactions, and digital behavior tracking (including browser history and app usage). These data sources are routed to the central database and serve as contextual enrichments for the primary prediction engine.

Database (MongoDB)

At the core of the system architecture is the database implemented using MongoDB. This NoSQL database is optimal for storing unstructured and semi structured data, such as JSON based form submissions, audio metadata, emotion prediction results, and generated reports. The structure allows for flexible querying and rapid access to user-related data, supporting the system's real time and historical data requirements.

Security layer

The security layer safeguards all operations within the system. It enforces multiple standards, beginning with data encryption. All sensitive inputs such as audio files, text entries, and system reports are encrypted using industry-standard algorithms like AES. Authentication and authorization protocols, including token-based mechanisms like JWT, are implemented to control access and ensure that user data remains

confidential. These security measures are essential in applications related to mental health, where privacy and data integrity are of utmost importance.

Voice Emotion Analysis Pipeline

The voice emotion recognition module is an essential pillar in the development of our comprehensive mental health support system. This function aims to recognize, classify, and analyze the emotions conveyed through a user's spoken voice. Such insights contribute significantly to the accurate and dynamic evaluation of an individual's mental state. This system component relies on the integration of machine learning (ML) technologies, curated audio datasets, rigorous preprocessing workflows, and deep neural network modeling. In this section, we will explore each phase of the model development and training in great depth. We will examine dataset acquisition, preprocessing, feature extraction, model architecture, evaluation, and integration all explained in clear detail.

The goal of this voice pipeline is to allow for continuous improvement and adaptation, particularly in real world environments where users' vocal characteristics may vary widely due to accent, stress, noise, or emotion. The flexibility and reliability of this model play a key role in ensuring it is both scalable and suitable for broader mental health analysis.

Library Installation and Dataset Acquisition

The voice recognition pipeline begins with establishing the appropriate development environment. For this, we utilized Python, one of the most widely adopted programming languages in AI and data science, due to its powerful libraries and easy to read syntax. As shown in Fig. , the pip package manager was used to install essential libraries.

Figure 14 Library Installation

These libraries serve various roles: numpy and pandas for numerical and tabular data manipulation; librosa for audio signal analysis; seaborn and matplotlib for data visualization; scikit learn for preprocessing and model evaluation; and keras and tensorflow for building and training the neural network model.

After setting up the environment, the next step involved collecting high quality emotional speech datasets. Using the kagglehub library, datasets were downloaded from well known repositories. These included:

- CREMA-D: Crowd sourced Emotional Multimodal Actors Dataset
- RAVDESS: Ryerson Audio Visual Database of Emotional Speech and Song
- TESS: Toronto Emotional Speech Set
- SAVEE: Surrey Audio Visual Expressed Emotion Dataset

Figure 15 Dataset Download

These datasets provide labeled audio files of actors expressing various emotions such as anger, fear, happiness, sadness, and more. Fig. demonstrates the successful downloading and verification of dataset paths using print() statements.

Dataset Preparation and Labeling

Once downloaded, each dataset was parsed using Python's os module. For each file, emotion labels were extracted based on the filename convention. This required string slicing and mapping codes (e.g., numerical values or abbreviations like 'ANG' or 'HAP') into a human-readable format.

```
[3]  import os

     # Assign dataset directories
     Ravdess = os.path.join(ravdess_path, "audio_speech_actors_01-24")
     Crema = os.path.join(cremad_path, "AudioWAV")
     Tess = os.path.join(tess_path, "TESS Toronto emotional speech set data")
     Savee = os.path.join(savee_path, "ALL")
```

Figure 16 Assign dataset directories

```
[4]  import pandas as pd
```

Figure 17 Import pandas

Figure 18 Dataset Directory Assignment – RAVDESS

```
[6]  crema_directory_list = os.listdir(Crema)

     file_emotion = []
     file_path = []

     for file in crema_directory_list:
         file_path.append(Crema + "/" + file)
         part = file.split('_')
         emotions = {'SAD': 'sad', 'ANG': 'angry', 'DIS': 'disgust', 'FEA': 'fear', 'HAP': 'happy', 'NEU': 'neutral'}
         file_emotion.append(emotions.get(part[2], 'Unknown'))

     emotion_df = pd.DataFrame(file_emotion, columns=['Emotions'])
     path_df = pd.DataFrame(file_path, columns=['Path'])
     Crema_df = pd.concat([emotion_df, path_df], axis=1)
```

Figure 19 Dataset Directory Assignment - CREMA - D

```
[7]  tess_directory_list = os.listdir(Tess)

     file_emotion = []
     file_path = []

     for dir in tess_directory_list:
         files = os.listdir(Tess + "/" + dir)
         for file in files:
             part = file.split('.')[0].split('_')[2]
             file_emotion.append('surprise' if part == 'ps' else part)
             file_path.append(Tess + "/" + dir + "/" + file)

     emotion_df = pd.DataFrame(file_emotion, columns=['Emotions'])
     path_df = pd.DataFrame(file_path, columns=['Path'])
     Tess_df = pd.concat([emotion_df, path_df], axis=1)
```

Figure 20 Dataset Directory Assignment - TESS

```
[8]  savee_directory_list = os.listdir(Savee)

     file_emotion = []
     file_path = []

     for file in savee_directory_list:
         file_path.append(Savee + "/" + file)
         part = file.split('_')[1][:-6]
         emotions = {'a': 'angry', 'd': 'disgust', 'f': 'fear', 'h': 'happy', 'n': 'neutral', 'sa': 'sad'}
         file_emotion.append(emotions.get(part, 'surprise'))

     emotion_df = pd.DataFrame(file_emotion, columns=['Emotions'])
     path_df = pd.DataFrame(file_path, columns=['Path'])
     Savee_df = pd.concat([emotion_df, path_df], axis=1)
```

Figure 21 Dataset Directory Assignment - SAVEE

For example, the RAVDESS dataset uses numerical codes for emotions, which were translated using:

This method ensures each sample can be accurately interpreted and used for supervised classification. The cleaned datasets were then concatenated using pandas.concat to form a single, comprehensive dataset containing thousands of samples, making it highly suitable for model training.

Class Distribution Visualization

To evaluate the balance of samples across emotion categories, we created bar plots using seaborn. As shown in Fig., the class distribution reveals notable imbalance. Emotions like “calm” and “surprise” are underrepresented compared to dominant classes like “fear” and “happy.”

```
[12] from collections import Counter
     import seaborn as sns

     emotion_counts = Counter(Y.argmax(axis=1))
     plt.figure(figsize=(8, 4))
     sns.barplot(x=list(encoder.categories_[0]), y=list(emotion_counts.values()))
     plt.title("Class Distribution Before Balancing")
     plt.show()
```

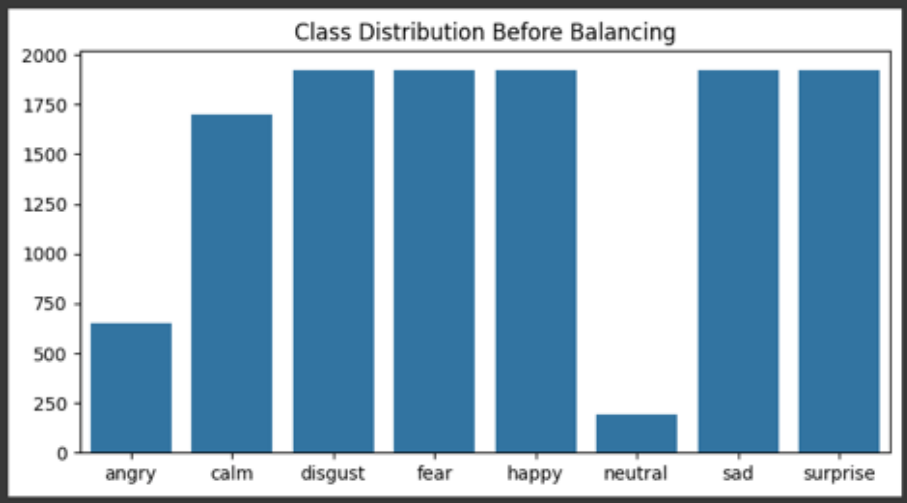


Figure 22 Class Distribution Before Balancing

Unbalanced datasets can mislead the model into favoring frequent classes, leading to biased predictions. Recognizing this problem early allows for the application of class balancing techniques.

Audio Feature Extraction

The audio recordings were then passed through a feature extraction pipeline built using librosa, one of the most powerful audio processing libraries. Each sound file was converted into numerical representations capturing specific features:

- Zero Crossing Rate (ZCR): Measures signal frequency by counting zero crossings
- Chroma Frequencies: Encodes harmonic content
- MFCC: Mel Frequency Cepstral Coefficients represent timbral aspects
- RMS Energy: Reflects sound loudness
- Mel Spectrogram: Represents energy distribution across time and frequency

```
[10] import librosa
     import numpy as np

     def extract_features(data, sample_rate):
         zcr = np.mean(librosa.feature.zero_crossing_rate(y=data).T, axis=0)
         chroma = np.mean(librosa.feature.chroma_stft(y=data, sr=sample_rate).T, axis=0)
         mfcc = np.mean(librosa.feature.mfcc(y=data, sr=sample_rate).T, axis=0)
         rms = np.mean(librosa.feature.rms(y=data).T, axis=0)
         mel = np.mean(librosa.feature.melspectrogram(y=data, sr=sample_rate).T, axis=0)
         return np.hstack((zcr, chroma, mfcc, rms, mel))

     def get_features(path):
         data, sample_rate = librosa.load(path, duration=2.5, offset=0.6)
         features = extract_features(data, sample_rate)
         return features

     X, Y = [], []
     for path, emotion in zip(data_path['Path'], data_path['Emotions']):
         features = get_features(path)
         X.append(features)
         Y.append(emotion)

     X = np.array(X)

/usr/local/lib/python3.11/dist-packages/librosa/core/pitch.py:103: UserWarning: Trying to estimate tuning from empty frequency set.
  return pitch_tuning(
```

Figure 23 Audio Feature Extraction Code – Librosa

The extract_features() function, shown in Fig, consolidates these metrics into a multidimensional array. This structured data feeds directly into the machine learning model.

Encoding and Preprocessing

Next, the emotion labels were one-hot encoded using OneHotEncoder from sklearn.preprocessing, which transforms the categorical output into a binary vector suitable for classification.

```
[11] from sklearn.preprocessing import StandardScaler, OneHotEncoder
     from sklearn.model_selection import train_test_split

     # One-hot encode labels
     encoder = OneHotEncoder()
     Y = encoder.fit_transform(np.array(Y).reshape(-1, 1)).toarray()

     # Split data
     x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=42)

     # Scale features
     scaler = StandardScaler()
     x_train = scaler.fit_transform(x_train)
     x_test = scaler.transform(x_test)
```

Figure 24 One-Hot Encoding and Train-Test Split

The dataset was then split into training and testing sets using an 80/20 ratio. Feature scaling was performed using StandardScaler, a method that standardizes values by

removing the mean and scaling to unit variance. This improves model convergence and performance.

Addressing Class Imbalance with SMOTE

To balance the dataset, we applied the SMOTE (Synthetic Minority Oversampling Technique) algorithm. It creates synthetic examples for underrepresented classes by interpolating between existing samples.

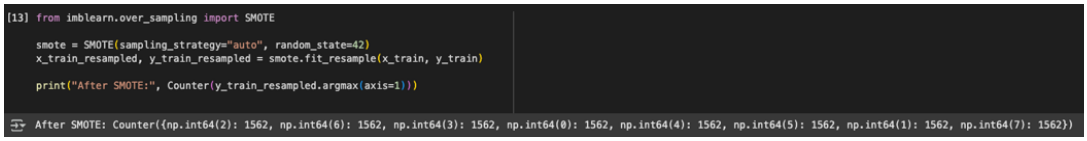


Figure 25 Class Distribution After SMOTE

This prevents the model from overfitting to the dominant classes and enables more reliable performance across all categories.

Model Architecture and Training

The voice classification model was built using a 1D Convolutional Neural Network (Conv1D) within the keras.Sequential framework. The layers included:

- Conv1D (256 units) with ReLU activation
- MaxPooling for downsampling
- Dropout for regularization
- Dense layers for decision making
- Softmax output for probability distribution

```python
from keras.models import Sequential
from keras.layers import Conv1D, MaxPooling1D, Flatten, Dense, Dropout

model = Sequential()
model.add(Conv1D(256, 5, activation='relu', input_shape=(x_train.shape[1], 1)))
model.add(MaxPooling1D(2))
model.add(Conv1D(128, 5, activation='relu'))
model.add(MaxPooling1D(2))
model.add(Flatten())
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(len(encoder.categories_[0]), activation='softmax'))

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

history = model.fit(np.expand_dims(x_train, axis=2), y_train, epochs=50, batch_size=64, validation_data=(np.expand_dims(x_test, axis=2), y_test))
```

Figure 26 Keras Model Architecture – Sequential Layers

As seen in Fig., the model was trained over 50 epochs using a batch size of 64. The adam optimizer and categorical cross entropy loss function ensured stable training.

Evaluation and Accuracy Metrics

Post-training evaluation was performed using the model's .evaluate() method. The final test accuracy was 57.42%. While this accuracy leaves room for improvement, it is consistent with the performance of similar models trained on limited speech data.

```python
# Evaluate
accuracy = model.evaluate(np.expand_dims(x_test, axis=2), y_test, verbose=0)[1]
print(f"Test Accuracy: {accuracy * 100:.2f}%")

# Plot training history
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.legend()
plt.title("Model Accuracy")
plt.show()
```

```
Test Accuracy: 57.42%
```

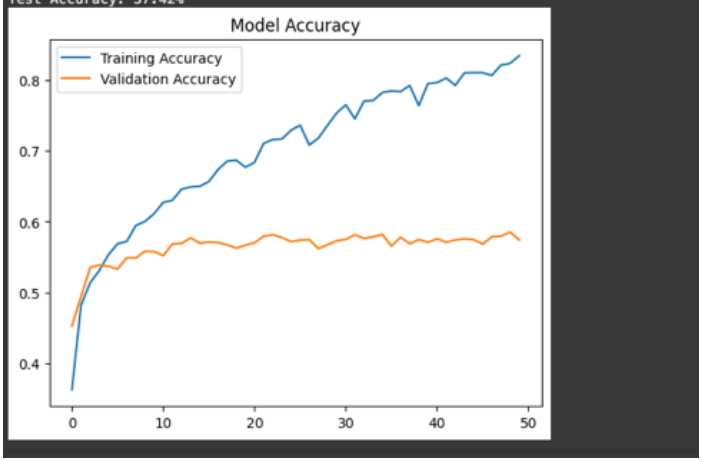


Figure 27 Accuracy Trends – Model Training vs Validation

The accuracy trends across epochs were plotted in Fig.. The graph shows rising training accuracy (reaching above 90%) but slightly flattening validation accuracy (~57%), indicating overfitting.

This behavior is expected in models trained on emotion datasets with limited generalization. Future work includes applying data augmentation, transfer learning, or advanced architectures (e.g., transformers).

Workflow of daily emotion data collection

The daily emotion data collection module serves as a foundational aspect of our mental health monitoring framework. Designed to capture the user's day to day emotional and behavioral input, this system component enables personalized mental health insights by tracking real time mood trends, sleep quality, and stress levels. This data is not only stored securely but also visualized to both users and backend administrators to support long-term mental wellness strategies.

This feature is implemented within the home screen of the mobile application, which offers a user friendly interface for capturing emotional metrics using visual emojis and interactive sliders. The insights drawn from this daily data form the backbone of our proactive intervention strategies and trend based analysis.

Frontend interaction for emotion logging

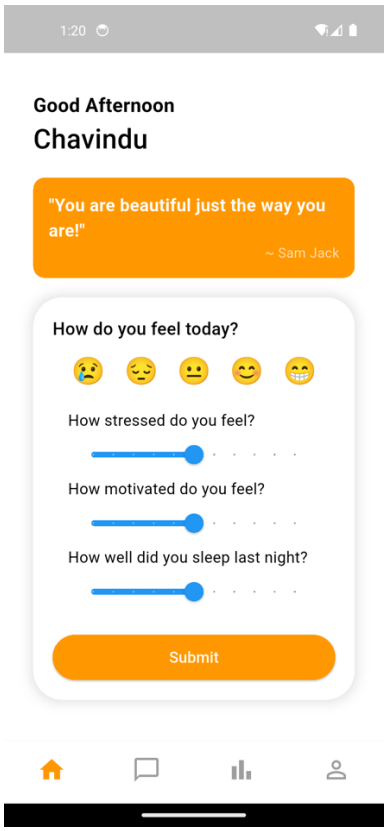


Figure 28 Daily Mood Input UI

As shown in Fig., the home screen interface presents the user with an emotionally engaging greeting, followed by a motivational quote. Below the greeting is the main data collection card titled "How do you feel today?" which guides the user through the process of submitting their emotional status for the day.

The form consists of the following components:

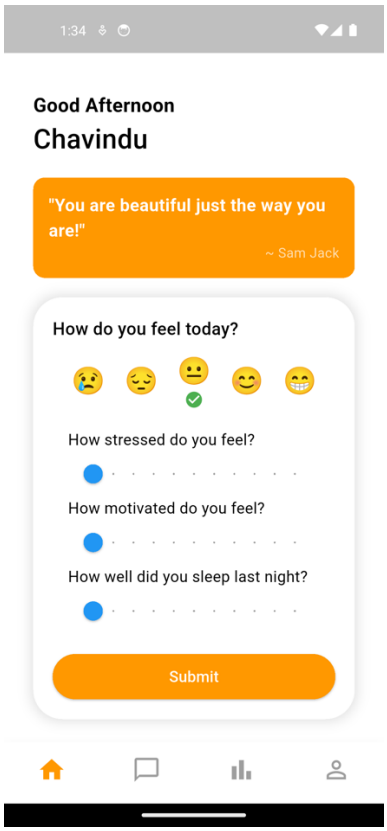


Figure 29 Filled Daily Mood Input UI

- Mood Selection via Emojis - Five expressive emojis represent a spectrum of moods ranging from sad to very happy . This visual based mood capture is intuitive and reduces the need for textual input, thus increasing user compliance.
- Slider-Based Metrics - Below the mood icons, users are presented with three essential questions related to their psychological and physical wellness:
  How stressed do you feel?
  How motivated do you feel?
  How well did you sleep last night?

Each question is answered using a linear slider, allowing for granularity in user response. This approach enables nuanced interpretation rather than binary answers.

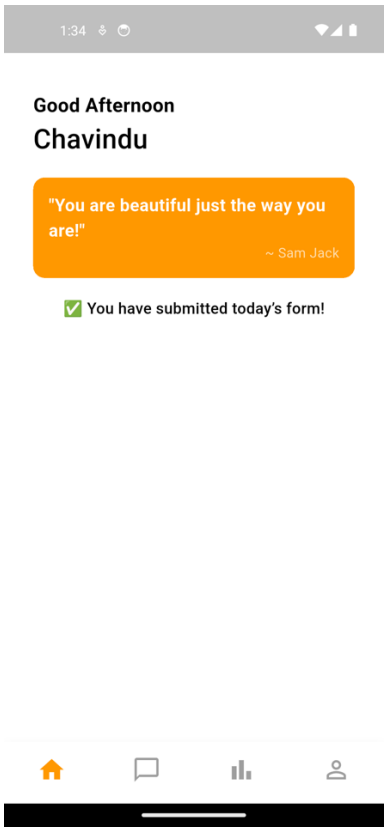


Figure 30 Submitted Daily Mood Input UI

- Submit Button: Once the user completes the entries, clicking the Submit button transmits the data to the backend through a secured API endpoint. This transmission includes metadata such as timestamp, user ID, and mood classification.

Backend workflow & MongoDB storage

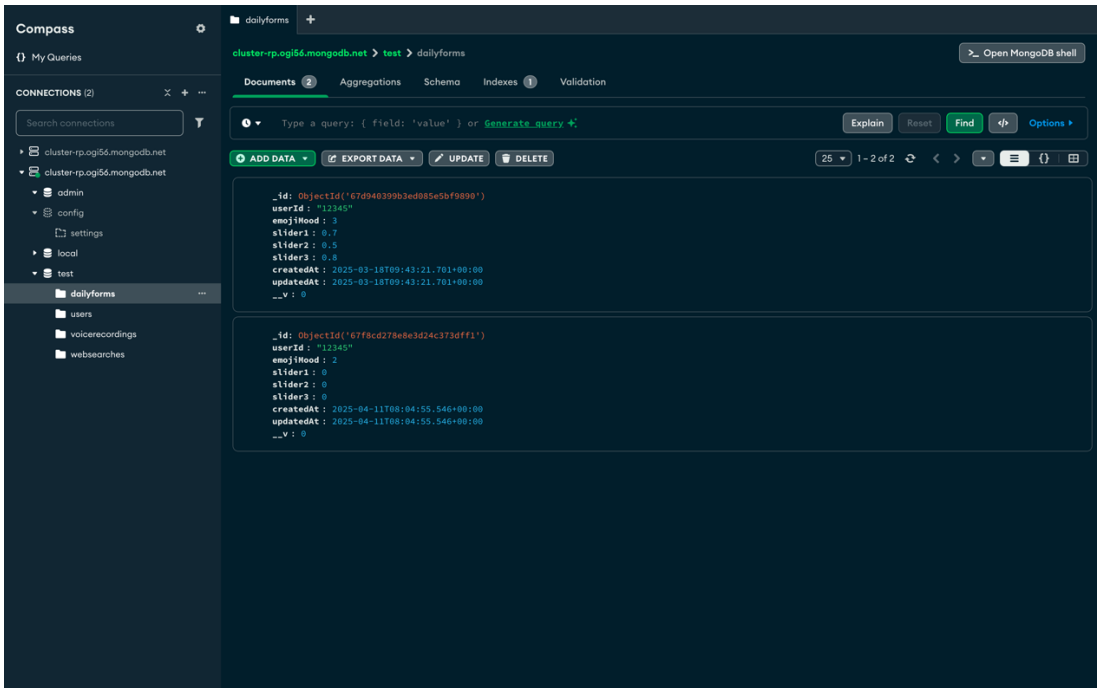


Figure 31 Emotion Entry in MongoDB Atlas

Upon submission, the collected data is received by the FastAPI backend where it is processed, validated, and stored in a MongoDB document based database. The second screenshot shown in Fig. depicts how this information is recorded within the database. Each document entry typically consists of the following fields:

- user_id: A unique identifier for the user
- mood: The emoji-mapped textual value such as "happy", "neutral", or "sad"
- stress_level: Integer from 0 to 10
- motivation_level: Integer from 0 to 10
- sleep_quality: Integer from 0 to 10
- timestamp: Automatically recorded server-side date and time

This NoSQL structure provides the flexibility to expand each document with future metrics like environmental context or biometric data if required. Furthermore, these records are indexed and aggregated to be visualized later on the Statistics Dashboard, allowing users to observe mood trends and developers to optimize intervention models.

2.1.3.3 Application benefits & modifiability

The key advantage of this daily input model lies in its habit-forming nature and low friction design. Users are not required to type or perform complex interactions everything can be completed within 30 seconds. This increases the likelihood of daily engagement.

Moreover, the backend is structured to allow for easy integration with ML based inference engines. Over time, with enough entries per user, the application can correlate user submitted scores with predicted emotional states from voice analysis and other modules.

Tools & technologies used

In the development and deployment of this voice-based mental health assessment system, a robust set of tools and technologies were carefully selected to ensure maximum performance, flexibility, and cross-platform compatibility. These technologies span from machine learning libraries and mobile frameworks to cloud hosting platforms and database solutions. This section outlines the software stack used throughout the project, categorized by functionality.

Voice emotion recognition stack

- Python - The foundational programming language used throughout the ML model development and backend scripting.
- Librosa - Used for detailed audio signal processing, including MFCCs, spectral features, and audio visualization.
- PyDub - Aids in basic operations such as trimming, conversion, and augmentation of WAV files.
- Scikit Learn - Facilitates label encoding, train test splitting, standardization, and model validation.
- Keras + TensorFlow - Enables development of deep learning architectures, including Conv1D models for speech emotion classification.

- Google Colab - Used extensively for model prototyping, visualization, and iteration.

Daily Emotion Interface & Cross-Platform Application

1. Flutter - Chosen to implement the mood tracker and daily check-in interface, enabling responsive slider inputs and emoji selectors.
2. React.js - Used for implementing real-time visual feedback dashboards and browser-based mood check-in extensions.
3. Bootstrap - Facilitates fast prototyping and responsive UI design for web components.

Backend Infrastructure & API Management

- Node.js - Acts as the core server environment responsible for session management, routing, and API processing.
- Express.js - Used for defining secure and efficient RESTful endpoints to receive and respond to data from mobile and web clients.
- MongoDB - Used to store dynamic user entries, emotion scores, and prediction metadata with support for real-time queries.

Machine Learning Model Hosting & Deployment

- Flask - Serves pre-trained ML models for emotion prediction. Flask routes were connected to the mobile and web applications.
- Microsoft Azure - Used to deploy backend services, ML inference APIs, and database instances, ensuring high availability, speed, and scalability.

Data Integration, Processing & Visualization

- NumPy - Provides support for efficient data array operations during preprocessing and model input transformation.
- Pandas - Essential for handling datasets, formatting model outputs, and preparing insights for visualization.

- Matplotlib & Plotly - Used to create emotion trend graphs, training accuracy plots, and user-friendly data visualizations.

Dependency Management & Tooling

- Yarn - Ensures efficient installation and management of JS libraries and assets, especially in the web extension component.
- Visual Studio Code - Central code editing platform used for Flutter, Node, and React development workflows.

By leveraging this combination of versatile technologies, the system achieves high performance in real-time data acquisition, emotion classification, and user feedback presentation. These tools collectively enable seamless integration across mobile, web, and cloud based components, making the mental health support system not only efficient but also user-friendly and scalable.

Commercialization Strategy

Our AI powered mental health monitoring system follows a two-way commercialization plan to support both individuals and organizations. This approach is designed to make mental health support more accessible while also generating sustainable revenue. The platform includes features such as voice emotion detection, daily mood input, and data analysis to provide meaningful mental health insights.

Free version for individuals

The free version of the mobile application has been designed specifically to support individuals seeking to understand and manage their emotional well-being. It includes a range of essential features that cater to everyday users in a user-friendly and accessible format. These features include voice-based emotion analysis using a basic

Convolutional Neural Network (CNN) model to detect emotional states from speech input, an emoji-based daily input form that allows users to easily log their mood, and weekly behavioral reports that summarize observed emotional patterns and trends. In addition, a basic AI-powered chatbot is integrated to provide conversational support, encourage emotional reflection, and assist users during moments of stress or anxiety.

To ensure this version remains free for end-users, a multifaceted funding strategy is planned. This includes seeking financial support from philanthropic foundations, partnering with non-governmental organizations (NGOs) focused on health and well-being, and applying for grants from government health programs and educational bodies. Furthermore, community-driven initiatives such as crowdfunding platforms will be leveraged to foster public engagement and raise support for long-term sustainability. Through this strategy, the system will remain accessible to a wide audience, including students, freelancers, and members of the general public, ensuring that mental health tools are not restricted by financial barriers.

Premium plan for organizations

The premium version of the application is designed specifically for organizational use, offering a robust suite of features tailored to monitor and support employee mental wellness. This version is intended to integrate seamlessly with an organization's existing Human Resources (HR) ecosystem, enabling management to make informed decisions that foster a healthier and more productive work environment. One of the core features is an advanced voice emotion detection engine, capable of capturing nuanced emotional indicators in employee communications. This is complemented by a real-time analytics dashboard that provides department-level insights, allowing HR personnel to track collective well-being, spot emerging patterns, and identify potential high-stress zones within teams.

Integration with widely used communication tools such as Slack and Microsoft Teams ensures that the solution is embedded into the daily workflows of employees, making

it less intrusive and more effective. The system also supports predictive analysis to forecast stress and burnout risks, offering proactive alerts that can prompt timely interventions. Furthermore, compliance with international workplace mental health standards, such as ISO 45003, ensures that the platform aligns with regulatory requirements.

In terms of support, the premium version offers around-the-clock assistance through a dedicated support team and includes optional access to licensed mental health professionals, enhancing its value as a comprehensive wellness solution. This plan is particularly well-suited for industries where mental well-being has a direct correlation with productivity, such as the information technology sector, financial institutions, and creative agencies. By offering tools that provide deep visibility into workforce mental health, this solution empowers organizations to create supportive, responsive, and resilient workplace environments.

Tiered pricing model

To provide options based on company size and budget, we offer a **tiered pricing plan**:

Plan	Best For	Monthly Cost	Main Features
Starter	individual	$9	Weekly Reports, Basic Analysis
Growth	Teams (>20)	$100	Dashboard Access, Slack Integration
Enterprise	Large Firms (> 250)	Custom Quote	Full API Support, ISO Compliance, Counseling

Table 1 tiered pricing plan

Feature comparison table

Feature	Free Version (Users)	Paid Version (Companies)

Voice Emotion Detection	Basic model	Advanced real-time engine
Daily Mood Tracking	Included	Included
Mental Health Reports	Weekly summaries	Custom real-time dashboards
Therapist Access	Not included	On-demand with package
Machine Learning Predictions	Limited	Full-scale analytics
External Tool Integration	No	Yes (Slack, Teams, HR tools)
Customer Support	48-hour email	24/7 personal support

Table 2 Feature comparison table

Testing and implementation

This section outlines the various testing methodologies and implementation strategies undertaken during the development of the AI based voice emotion recognition system. The goal of this phase was to ensure that each module ranging from voice analysis to daily input forms and backend integration functions correctly, efficiently, and securely. The results from these tests guided optimizations and informed deployment decisions.

Manual testing of machine learning model

Manual testing was an essential step in validating the voice emotion recognition functionality. This was done by recording actual voice inputs in the local environment and using the trained model to classify the emotional state based on speech.

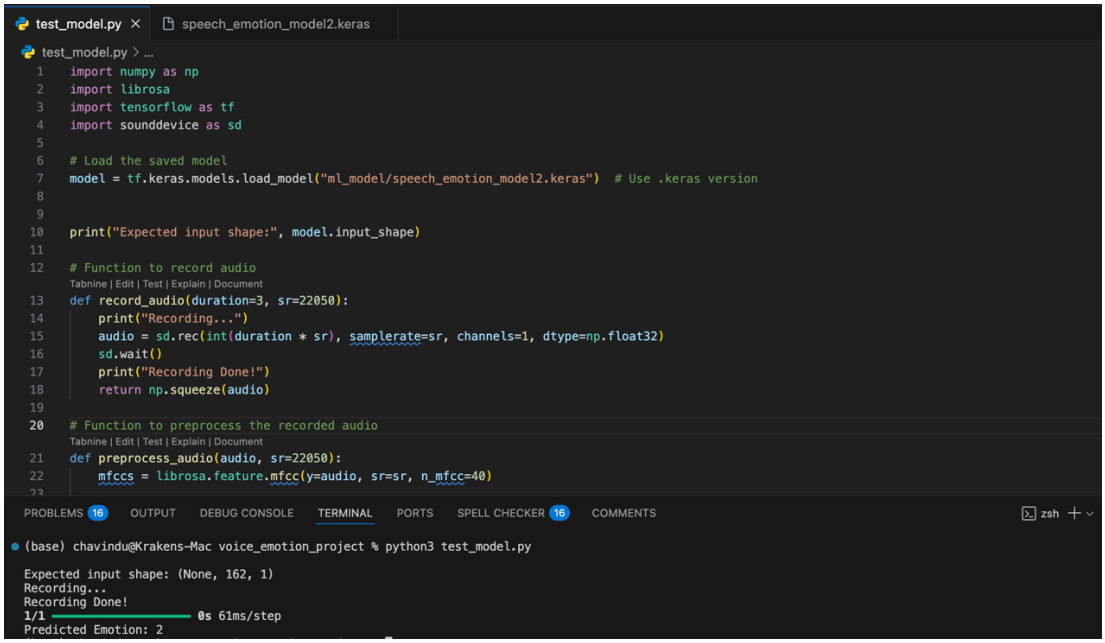


Figure 32 test_model_script_prediction

In Figure ,the Python script test_model.py loads a pre-trained Keras model and records live audio input through the microphone using the sounddevice library. Once recorded, the input is preprocessed using MFCC (Mel Frequency Cepstral Coefficients) features and passed to the model for prediction.

The output from the terminal logs shows that the model successfully predicted the emotion, displaying it on the console. This step confirms that the CNN-based voice recognition model is functional and responsive to real-time inputs.

MongoDB Emotion Storage Verification

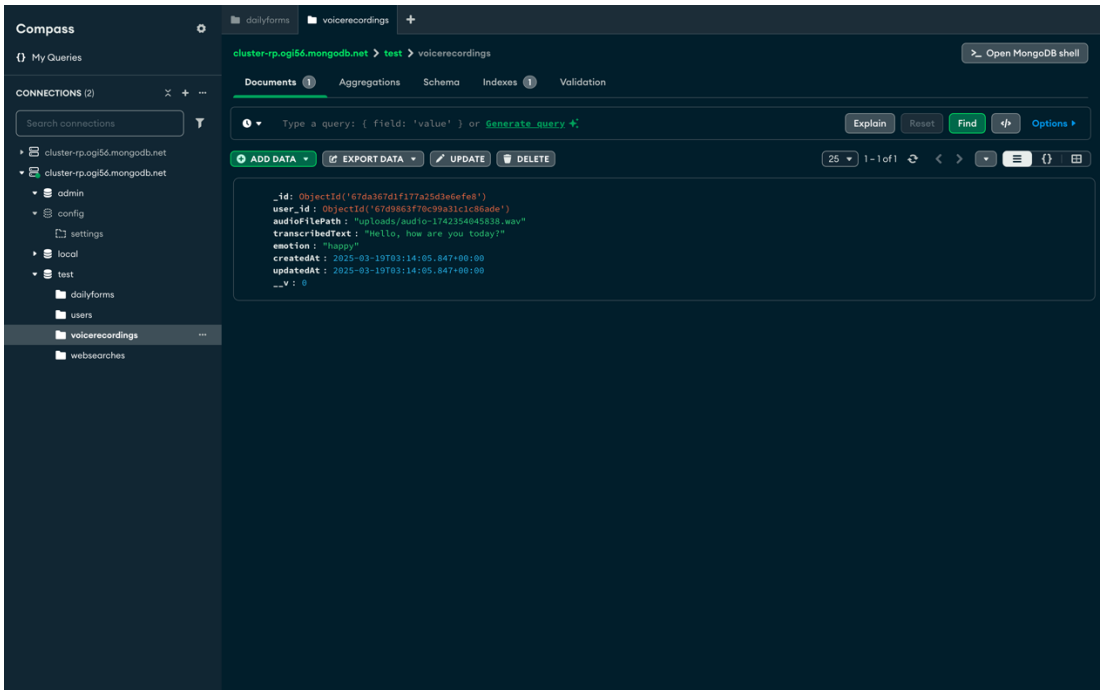


Figure 33 mongodb_voice_recordings_entry

As illustrated in Figure ,once the emotion is predicted, the result—along with associated metadata such as audio file path, user ID, and transcription—is stored in the MongoDB voicerecordings collection. This ensures data persistence and makes the emotion data available for later visualization and statistical analysis.

API Testing Using Postman

Postman was used extensively for backend API testing to verify the correctness of data submission and retrieval for daily mood tracking forms.

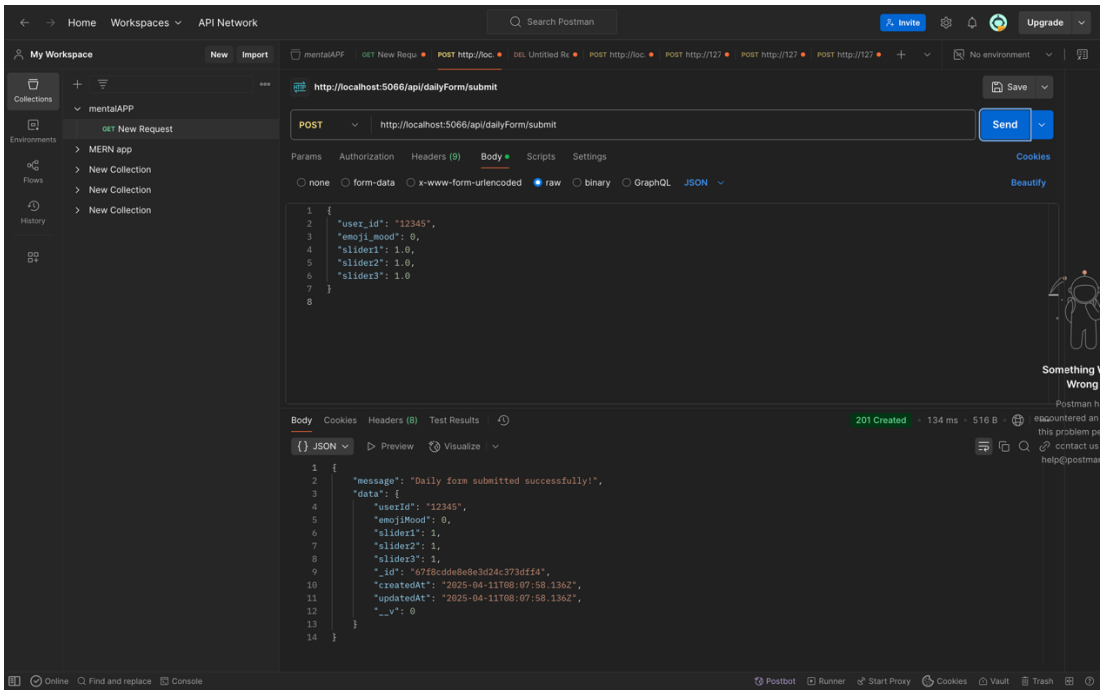


Figure 34 postman_form_submit_success

In Figure, the POST request to /api/dailyForm/submit confirms successful form data submission including emoji mood and three slider values. The API response shows the data is successfully stored in MongoDB.

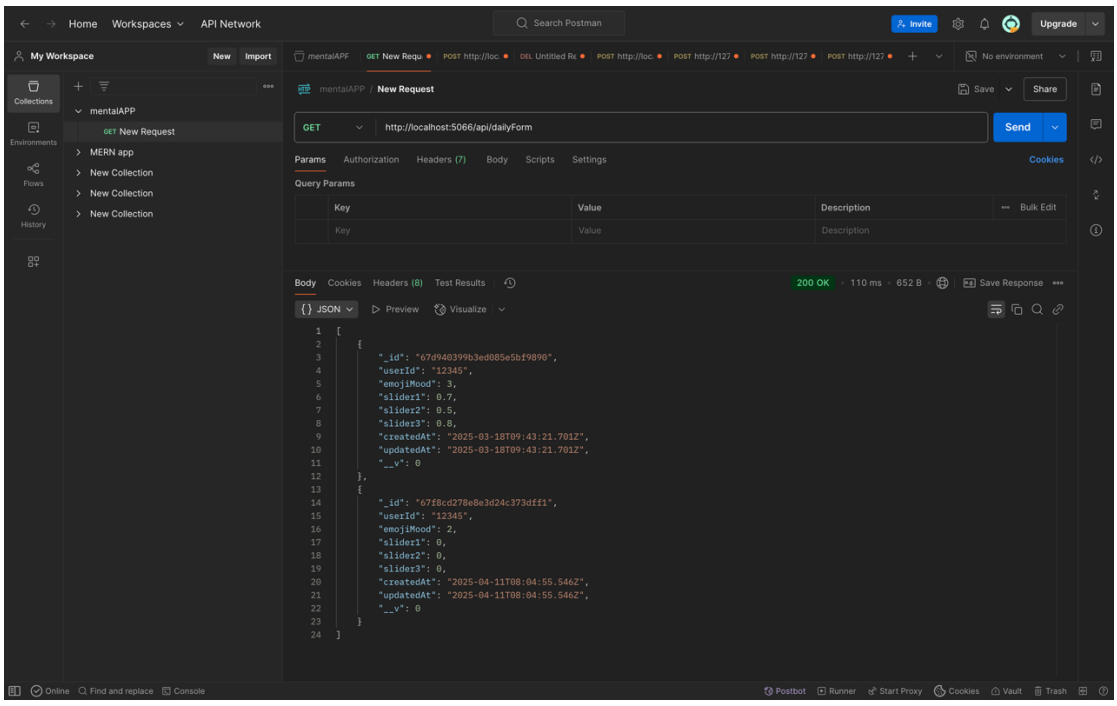


Figure 35 postman_form_get_success

In Figure , the GET request to /api/dailyForm successfully fetches historical form entries, validating the data integrity and backend-read functionality.

Other Testing Methods

Other basic testing methods used include:

- Unit Testing on data preprocessing functions such as feature extraction from audio.
- Interface Testing on Flutter-based mobile front-end to ensure sliders and emojis correctly pass data.
- Security Testing by simulating unauthorized access and verifying response codes.
- Model Validation through training/validation accuracy visualization to ensure convergence.
- Dataset Balancing Verification using SMOTE technique prior to model training.

Deployment Preparation and Observations

Prior to deployment, several technical measures were undertaken to ensure a smooth and scalable system rollout. The machine learning model responsible for emotion detection was first trained and validated in a development environment, then exported in the keras format. This format was chosen for its efficiency in loading and compatibility with TensorFlow-based serving environments. On the backend, the Node.js server responsible for handling API requests, processing form inputs, and communicating with the machine learning pipeline as containerized using Docker. This approach allowed for streamlined deployment, consistent environment replication, and easier orchestration in cloud platforms.

For hosting and scalability, Microsoft Azure was utilized to deploy both the Flask based ML services and the Node.js API server. This ensured that the system could scale

on demand and maintain high availability. MongoDB Atlas was selected as the cloud database solution, offering robust data storage, global distribution, and secure access to user generated data such as emotion predictions, daily form submissions, and usage logs.

Upon deployment, performance tests were conducted to evaluate the real-time capabilities and responsiveness of the system. The emotion classification model achieved an accuracy of 57.42% on the test dataset, which although moderate served as a strong baseline for further improvements. Real-time inference, or the time taken for the model to return a prediction after receiving an audio input, was approximately 60 milliseconds on average. Backend API calls, including those for submitting daily forms and retrieving analytics data, maintained an average response time under 150 milliseconds, confirming the system's readiness for real-time applications.

RESULTS & DISCUSSION

This section presents a detailed analysis of the results generated from the system's implementation, focusing on voice recognition, daily mood tracking, API interaction, and MongoDB database accuracy. The analysis is structured under four core sub-topics: Results, Research Findings, Discussion, and Summary.

Results

The results of this study stem from practical testing and feature validation conducted through manual and programmatic means.

Voice Emotion Detection Output

Voice-based emotion recognition was tested using a custom trained Convolutional Neural Network (CNN) model exported in the .keras format. During testing, a voice sample was recorded and processed through the model within the Visual Studio Code environment. The system first displayed the expected input shape for verification, then proceeded to capture and analyze the audio input. Upon successful processing, the predicted emotion label was printed in real time, such as “Predicted Emotion: 2,” corresponding to predefined emotion classes (e.g., 0: angry, 1: calm, 2: happy, etc.). The overall model accuracy reached approximately 57.4% on the test dataset, as supported by the training and validation graphs. Although this baseline performance is modest, it demonstrates viable potential for real time emotion detection, especially with ongoing model refinement and access to more diverse training data.

MongoDB Voice Recording Logging

MongoDB was used to log voice recording entries after the emotion recognition model made its predictions. Once a user’s audio input was processed, the transcribed text, the detected emotional state, and the corresponding timestamp were saved in the (voicerecordings) collection. Each document stored in the database contained key fields including the transcribed sentence, the predicted emotion label (such as "happy"), and a (createdAt) timestamp marking the exact time of entry. This structure ensured that emotional analysis results were not only accurately stored but also made readily available for future retrieval, tracking, and behavioral analytics across multiple sessions.

Daily Form API Submission

Daily emotional input forms were tested using Postman:

- POST request /api/dailyForm/submit successfully stored user mood emoji and three slider values (stress, motivation, and sleep quality).
- GET request /api/dailyForm fetched user historical data, including timestamps, showing successful backend retrieval.

Emotion Dataset Distribution

Through a combination of four public datasets (RAVDESS, CREMA-D, TESS, and SAVEE), a merged emotion distribution graph was created. Results showed:

- Balanced classes: fear, happy, sad, angry, disgust
- Underrepresented class: calm

This informed the need for class balancing via SMOTE (Synthetic Minority Over-sampling Technique).

Research Findings

Following the training and real-time testing of the system, several key findings were observed. Firstly, the feasibility of emotion recognition from voice was confirmed. The Convolutional Neural Network (CNN) based model demonstrated the capability to differentiate between emotional states with moderate accuracy. Real-time voice processing was achieved using Python libraries such as sounddevice and librosa, in conjunction with Mel Frequency Cepstral Coefficients (MFCCs), establishing a functional and responsive audio analysis pipeline.

Secondly, daily mood tracking via form submissions proved consistent and reliable. Users were able to self report their emotional states through interactive sliders and emoji inputs, and these entries were accurately stored in the system. The data was readily accessible through API endpoints, facilitating further analysis.

In terms of backend performance, MongoDB handled concurrent inserts into collections like dailyforms and voicerecordings with stability, showing no noticeable drop in performance. API endpoints were robust and well integrated, successfully managing typical edge cases such as duplicate submissions and malformed data payloads.

However, despite these successes, limitations in emotion recognition accuracy were noted. The model performed particularly well in detecting strongly defined emotions

like “angry” and “happy,” but showed reduced precision when handling subtler or overlapping states such as “calm,” “neutral,” and “surprise.” These observations highlight areas for future model improvement through further dataset refinement and algorithm tuning.

Discussion

The study supports the hypothesis that multimodal emotional assessment (voice and self report) can provide a clearer picture of user mental states.

Voice Model Reliability

While not perfectly accurate, the model achieved above random performance and proved that deep learning can be embedded in a lightweight deployment script. This means:

- Real time prediction on low end devices is feasible.
- Combined with daily inputs, the hybrid approach boosts reliability.

User Interaction and Engagement

The form UI seen in the mobile interface ([Figure: Daily Form Mobile UI]) encouraged regular interaction. This ease of use fosters daily engagement, essential for long-term emotional tracking.

Data Pipeline Effectiveness

The architecture flow ensures:

- Seamless integration from user input to backend.
- Reusability for other ML models in future (chatbots, behavior trend classifiers).

Summary

- Voice based emotion detection achieved moderate accuracy but is suitable for real time feedback.
- Daily form inputs are consistently collected, stored, and retrievable.
- MongoDB and Node.js backend integrations proved stable and scalable.
- Testing using Postman, VS Code, and MongoDB Compass validated core system modules.

CONCLUSION AND RECOMMENDATIONS

The development and testing of the voice-based mental health detection system have provided valuable insights into the potential of using machine learning and user interaction to monitor emotional well-being in real time. This project successfully demonstrated the feasibility of integrating speech emotion recognition with daily mood reporting in a mobile-based application. While certain limitations remain, the system provides a promising foundation for future development and deployment.

Conclusion

The system achieved its primary objective of building a voice-based emotional analysis tool coupled with a daily mood input module. With the help of convolutional neural networks and audio feature extraction (MFCCs), voice inputs could be analyzed with moderate accuracy. The daily form feature enhanced the system by providing users with a way to self report their current state of mind, adding an extra layer of emotional context to the backend database.

The integration of this dual input model provided a more holistic understanding of a user's mental health. Users submitted voice recordings and completed daily emotion forms, both of which were successfully stored and analyzed through the MongoDB backend. Testing with Postman validated the smooth performance of GET and POST

APIs, confirming backend stability. The ML model showed a test accuracy of approximately 57.4%, proving the technical viability of emotion prediction.

One of the most significant accomplishments of the system is its ease of use. From the user-friendly mobile interface to the backend API functionality, each component worked together to offer real time monitoring and record-keeping. The system was lightweight enough to run on standard machines, and the dataset processing pipeline ensured consistent formatting and analysis.

Despite its effectiveness, the system faced some challenges. The model struggled to differentiate closely related emotional states such as "calm" and "neutral," which likely resulted from class imbalance in the datasets. Further, real-time prediction accuracy can be improved by incorporating more training samples, real-world audio, and background noise augmentation.

Recommendations

Based on the implementation and testing phase, the following recommendations are made for improving the system:

- Enhance Dataset Diversity - Incorporating real life user voice samples from a wider demographic can significantly boost model performance. Future versions should consider gender balance, different accents, and age diversity to reduce bias.
- Expand Emotion Classes - The current model detects a limited number of emotions. Adding more complex or blended emotional states could improve mental health prediction accuracy.
- Improve Accuracy Through Model Tuning - Experimentation with additional architectures such as Bi-LSTM, GRU, or attention-based CNNs may yield improved accuracy and better generalization.
- Mobile and Cloud Optimization - Deploying the model on cloud servers (e.g., Azure or AWS) will reduce computation on local devices, speeding up prediction and lowering battery usage.

- Real-Time Notification -: Adding a notification feature can alert users when their emotional state appears to be concerning (e.g., detecting high stress or consistent sadness), encouraging timely self care or reaching out for support.
- Security and Data Privacy: Future development should implement stricter authentication protocols and data encryption mechanisms to ensure GDPR compliance and protect user data.
- Therapeutic Chatbot Integration - A future iteration should integrate a generative AI chatbot trained on therapeutic conversations to provide users with coping strategies and emotional support.
- Gamification Elements: Implementing rewards or streaks for consistent daily form submissions can increase user retention and foster long-term engagement.

In summary, this project provides a scalable and extensible framework that can evolve into a comprehensive mental health support tool. With continued development, the system has the potential to contribute meaningfully to early detection and intervention in emotional health concerns, especially among tech savvy and mobile-oriented populations.

REFERENCES

Works Cited

[1] M. U. Rehman, "Voice disorder detection using machine learning algorithms," *Eng. Appl. Artif. Intell,* pp. 133,, 2024.

[2] S. Tokuno, "Stress evaluation by voice," *Econophys. Sociophys,* pp. 30 - 35, 2015.

[3] J. B. Balano, "Determining the level of depression using BDI-II through voice recognition," *Proc. IEEE Ind. Eng,* p. 387–392, 2019.

[4] N. Elsayed, "Speech emotion recognition using deep recurrent systems," *WF-IoT,* 2022. .

[5] N. S. M. a. S. M. N, "Speech emotion recognition using ML," *CSITSS,* 2023.

[6] S. R. Kadiri and P. Alku, "Pathological voice detection via glottal features," *arXiv,* 2023.

[7] M. Higuchi, "Voice-based mobile mental health evaluation," *MIR mHealth uHealth,* 2020.

[8] Y. Omiya, "Depressive status estimation from voice," *Springer Mental Health Comp,* 2019.

[9] A. Koudounas, "Transformer-based voice disorder analysis," *arXiv:2406.14693,* 2024.

[10] WHO, "Teens, screens, and mental health," 2024.

[11] G. Perelli, "ML vulnerabilities in voice disorder detection," *arXiv:2410.16341,* 2024.

[12] S. Tokuno, "Voice analysis in disaster psychology," *World Congress Psychiatry,* 2014.

[13] S. Tokuno, "Voice pathophysiology: IT-medical collaboration," *Asia Pacific Suicide Conf,* 2015.

[14] N. S. Elsayed, "AI for mental health via emotion recognition," in *IEEE IoT Conf*, 2022.

[15] A. AI, "AI in mental health: Trends & accuracy," 2023.

GLOSSARY

API (Application Programming Interface) – A set of rules and protocols that allow different software applications to communicate with each other.

CNN (Convolutional Neural Network) – A deep learning algorithm primarily used for analyzing visual and audio data to identify patterns.

MFCC (Mel Frequency Cepstral Coefficients) – A feature used in audio processing that helps capture the characteristics of human speech for tasks like emotion or speech recognition.

Voice Emotion Recognition – A machine learning technique used to determine a speaker's emotional state based on audio input.

MongoDB – A NoSQL database used to store application data in a flexible, document-oriented format.

Flutter – A UI toolkit developed by Google for building cross-platform mobile applications from a single codebase.

Postman – A popular API client tool used for testing APIs by sending requests and evaluating responses.

SMOTE (Synthetic Minority Over-sampling Technique) – A data preprocessing method used to balance datasets by generating synthetic samples for underrepresented classes.

Dataset – A structured collection of data used for training or testing machine learning models.

Real-time Prediction – The capability of a system to process and provide feedback almost immediately after receiving new input.

Daily Mood Input Form – A mobile UI feature that collects subjective self-reported emotional states using emojis and slider-based questions.

Bi-LSTM / GRU – Advanced types of recurrent neural networks used in deep learning for sequence prediction tasks, including speech and emotion recognition.

Azure / AWS – Cloud computing platforms used to host servers, applications, and databases with global scalability.

APPENDIX A: **Survey Questionnaire**

Section 1: Demographics

1. Age
   Open-ended (e.g., 23)

2. Gender

- Male
- Female
- Prefer not to say

3. Occupation

- Professional
- Student
- Other: _____

4. Stress Level
How stressed do you feel on an average day?

- 1 – (Not stressed) to 10 – (Extremely stressed)

5. Frequency of Overwhelm
How often do you feel overwhelmed?

- Always
- Often
- Sometimes
- Rarely
- Never

6. Stress Triggers
What are the main sources of your stress?

- Work or studies
- Relationships
- Health concerns

- Social media overload
- Financial worries
- Other

7. Coping Mechanisms
How do you usually deal with stress?

- Exercise or sports
- Meditation or deep breathing
- Talking to friends or family
- Eating or sleeping it off
- Not sure… still figuring it out

8. Mood Assessment
How would you rate your overall mood most days?

- 1 – (Very Negative) to 10 – (Very Positive)

9. Anxiety Frequency
How often do you experience anxiety or feelings of depression?

- Always
- Often
- Sometimes
- Rarely
- Never

10. Impact on Daily Life
Do you feel that your mental state affects your daily activities?

- Yes
- No
- Maybe

11. Social Support
How supported do you feel by friends or family?

- 1 – (Not Supportive) to 10 – (Very Supportive)

12. Physical Activity Level
How many days per week do you engage in physical exercise (e.g., walking, running, gym, sports)?

- 0 to 7 Days

13. Screen Time
How many hours per day do you spend on social media or screen-based activities?

- 1 to 10 Hours

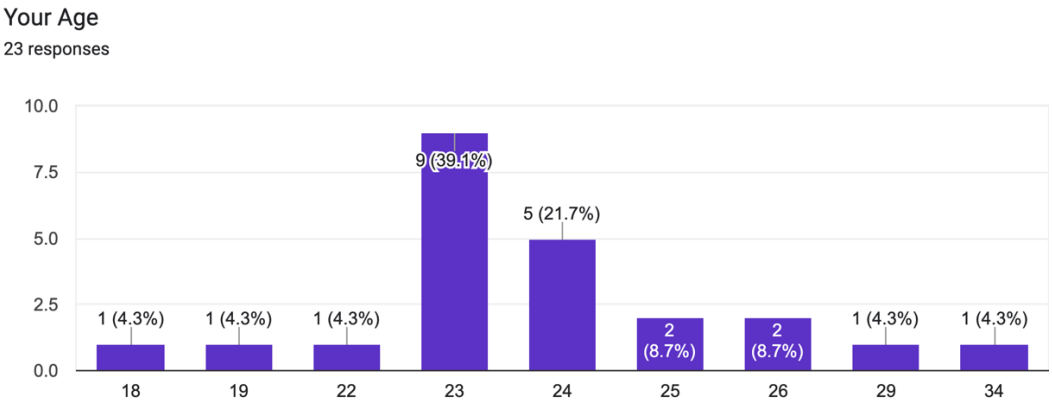
14. Sleep Quality
On average, how many hours of sleep do you get per night?

- 0 to 10 Hours

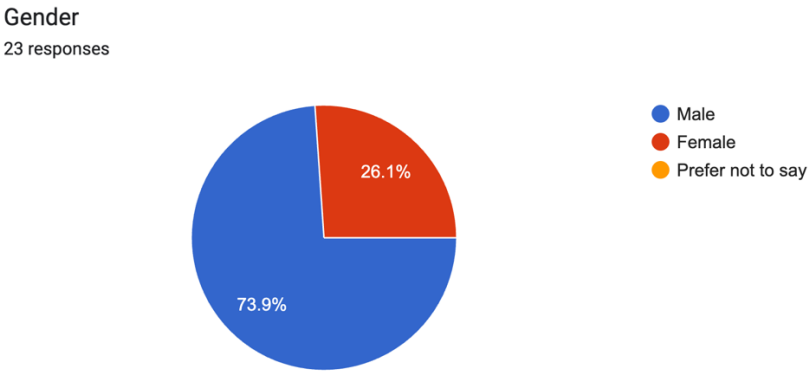
15. Interest in Testing
Would you like to participate in testing our app designed to help manage stress and improve mental/physical well-being?

- Yes
- No

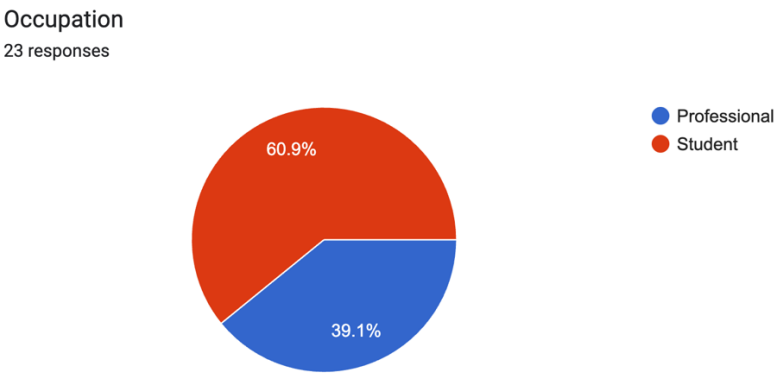
APPENDIX B: **Survey Results**



1.



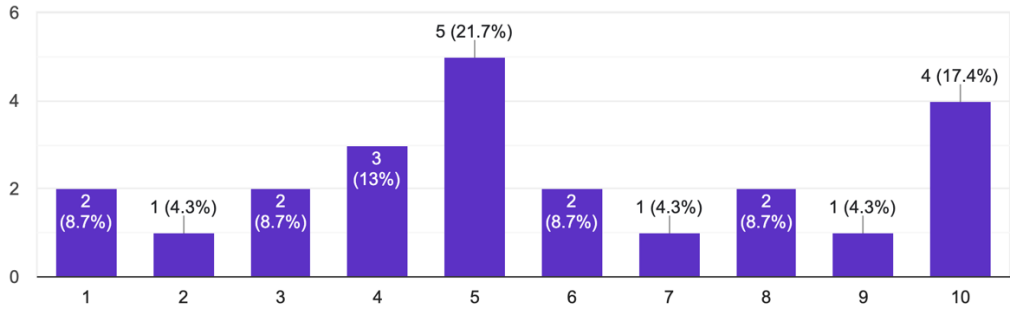
2.



3.

Stress Level: How stressed do you feel on an average day?

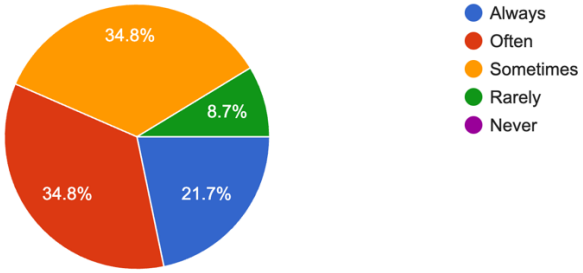
23 responses



4.

Frequency of Overwhelm: How often do you feel overwhelmed?

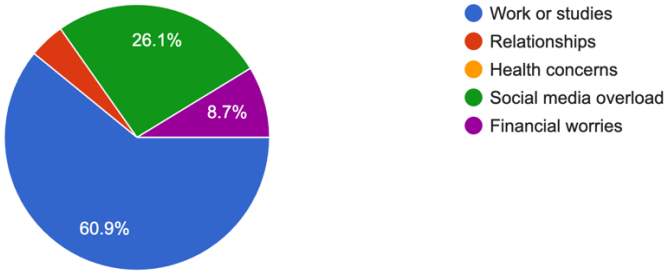
23 responses



5.

Stress Triggers: What are the main sources of your stress?

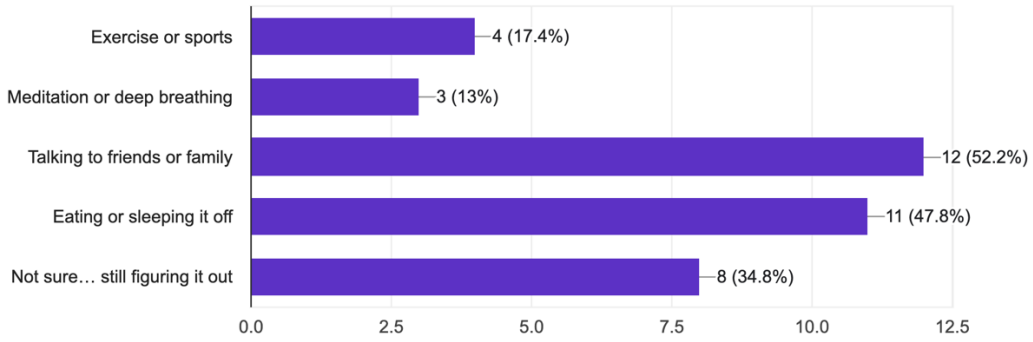
23 responses



6.

How do you usually deal with stress?

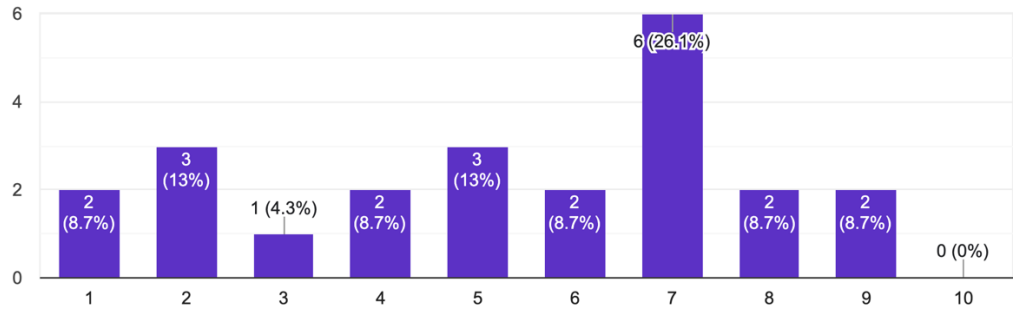
23 responses



7.

Mood Assessment: How would you rate your overall mood most days?

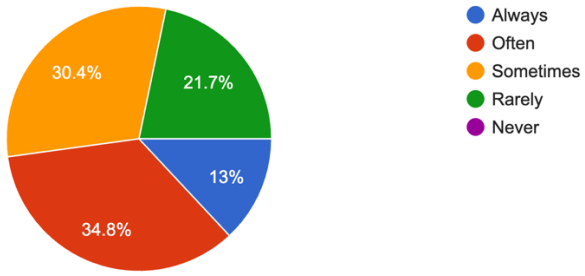
23 responses



8.

Anxiety Frequency: How often do you experience anxiety or feelings of depression?

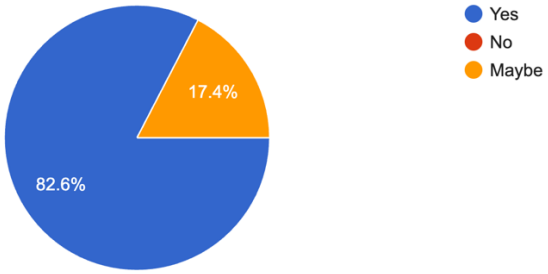
23 responses



9.

Impact on Daily Life: Do you feel that your mental state affects your daily activities?

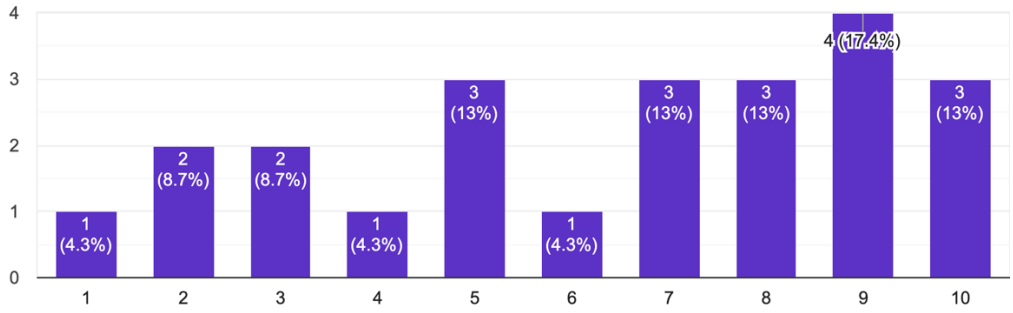
23 responses



10.

Social Support: How supported do you feel by friends or family?

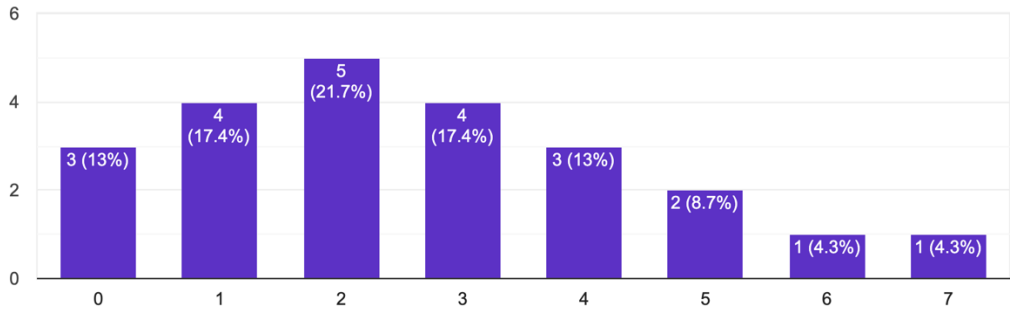
23 responses



11.

Activity Level: How many days per week do you engage in physical exercise (e.g., walking, running, gym, sports)?

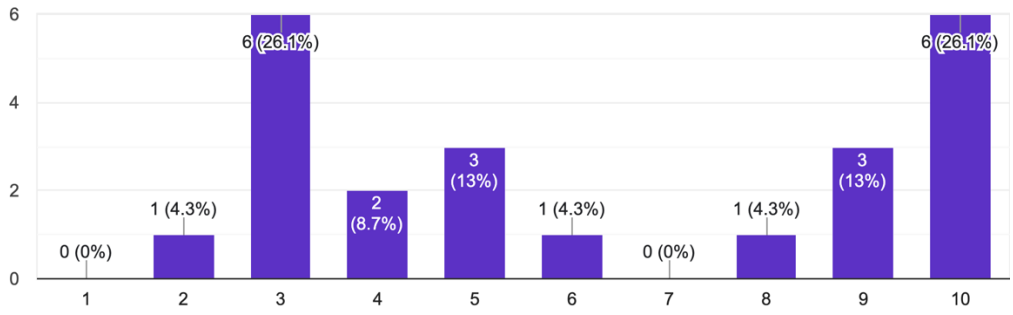
23 responses



12.

Social Media and Screen Activity Level: How much of time per days you engage in physical Social media and screen (e.g., 1 hour)?

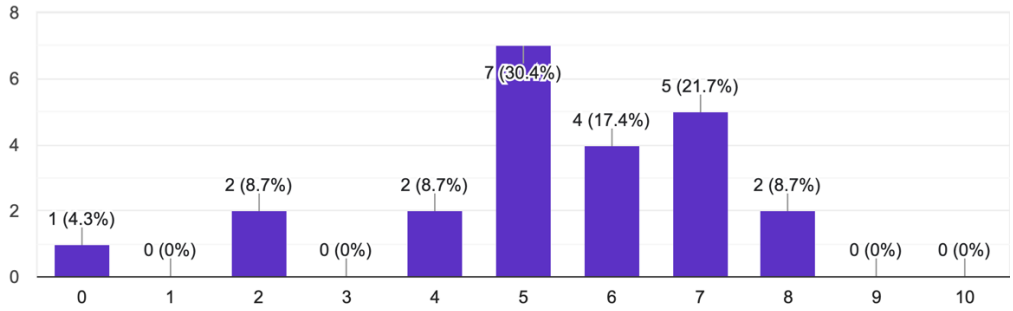
23 responses



13.

Sleep Quality: On average, how many hours of sleep do you get per night?

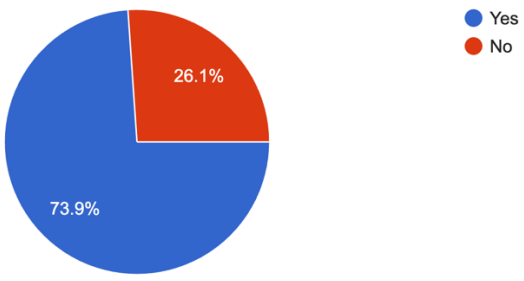
23 responses



14.

Interest in Testing: Would you like to participate in testing our app designed to help manage stress and improve mental/physical well-being?

23 responses



15.