A Project Report on

Cross-Modal Therapy Companion

Submitted by:

Sidra Aiman	**-**	**100522733042**
Einstein Ellandala	**-**	**100522733089**
Rohith Kodam	**-**	**100522733093**

Under the supervision of

PROF. P. V. SUDHA

Assistant Professor, Department of CSE,

UCE(A), OU, Hyderabad-500007

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,

UNIVERSITY COLLEGE OF ENGINEERING(A),

OSMANIA UNIVERSITY, HYDERABAD-500007

JUNE-2025

2022 – 2026

UNIVERSITY COLLEGE OF ENGINEERING (AUTONOMOUS)

OSMANIA UNIVERSITY, HYDERABAD – 500007

CERTIFICATE

This is to certify that the Mini Project work titled "**Cross-Modal Therapy Companion**" submitted by **Sidra Aiman, Einstein Ellandala, Rohith Kodam** students of the Department of Computer Science and Engineering, University College of Engineering, Osmania University, is a record of the bonafide work carried out by them during the academic year 2025-26. The project is submitted in partial fulfilment of the requirements for the award of the degree of Bachelor of Engineering in Computer Science and Engineering. The work has been carried out under the supervision of **PROF. P. V. SUDHA**, and it is to the best of my knowledge that the work is original and reflects the sincere efforts of the students.

Signature of the Supervision	Signature of the Head of the Dept.
PROF. P. V. SUDHA	**PROF. P. V. SUDHA**
Department of CSE, University College of Engineering, Osmania University	Department of CSE, University College of Engineering, Osmania University

DECLARATION

We **Sidra Aiman, Einstein Ellandala, Rohith Kodam** students of the Department of Computer Science and Engineering, University College of Engineering, Osmania University, hereby declare that the work presented in this Mini Project titled "**Cross-Modal Therapy Companion**" is an original contribution carried out by us during the academic year 2025-26. This project report is submitted in partial fulfilment of the requirements for the degree of Bachelor of Engineering in Computer Science and Engineering. The project work has not been submitted elsewhere for the award of any degree or diploma.

We affirm that no part of this report is plagiarized, and wherever references have been made, they have been appropriately cited. The findings and analysis presented in the report are based on our genuine and authentic work under the guidance of **PROF. P. V. SUDHA**.

We further declare that we have adhered to ethical practices throughout the research and project development process, maintaining academic integrity at every stage. The data, analysis, and outcomes of this report are factual to the best of our knowledge, and we take full responsibility for the contents of this submission.

Sidra Aiman	**-**	**100522733042**
Einstein Ellandala	**-**	**100522733089**
Rohith Kodam	**-**	**100522733093**

ACKNOWLEDGEMENT

We would like to express our deep sense of gratitude and whole-hearted thanks our project guide **PROF. P. V. SUDHA**, Professor at Department of Computer Science and Engineering, University College of Engineering Osmania University, for giving us the privilege of working under his guidance, with tremendous support and cogent discussion, constructive criticism, and encouragement throughout this dissertation work carrying out the project work. We also thank him for his support from the department and for making all the resources available to us students. We also extend our thanks to the entire faculty of the Department of Computer Science and Engineering, University College of Engineering, Osmania University who encouraged us throughout the course of our Bachelor's degree and allowed us to use the many resources present in the department. Our sincere thanks to our parents and peers for their valuable suggestions, morals, strength, and support for the completion of our project.

ABSTRACT

The increasing prevalence of mental health awareness has catalyzed the development of digital tools aimed at promoting well-being. However, a significant gap exists in the market, with most applications relying on subjective, single-modality inputs like manual mood logging. These methods often lack the objectivity and holistic perspective necessary for profound self-reflection. This project, titled "Mindful Companion," addresses this gap by presenting a novel, cross-modal web application designed as a private, data-driven dashboard for personal emotional analysis. The system provides users with a comprehensive and insightful tool to track, visualize, and understand the intricate connections between their emotional state, biometric expressions, and lifestyle habits.

The core innovation of the "Mindful Companion" lies in its multi-modal data integration framework. The application captures data from three distinct sources to build a rich, contextualized user profile. Firstly, it employs a **Computer Vision** module for real-time **Facial Emotion Recognition**, utilizing the robust DeepFace library to analyze images captured from a user's webcam. This provides an objective assessment of expressed emotion. Secondly, an **Audio Processing** module analyzes the user's vocal tone through a state-of-the-art **Wav2Vec2 Transformer model**, a deep learning architecture that can discern emotional cues from the prosodic features of speech. Thirdly, the system incorporates **self-reported data**, allowing users to log crucial lifestyle factors such as hours of sleep and daily physical activity level.

The system's backend is developed using **Python** with the **Flask** micro-framework, creating a RESTful API to handle all data processing and analysis tasks. The frontend is a responsive, single-page application built with standard **HTML, CSS, and JavaScript**, ensuring broad accessibility across all modern devices and browsers. The communication between client and server is handled asynchronously via the `fetch` API, with data exchanged in JSON format. For data persistence in this prototype, the system utilizes the Pandas library to manage a local CSV file, which serves as a secure, on-device logbook.

A key deliverable of this project is a custom **heuristic assessment algorithm** that synthesizes the multi-modal inputs into a single, easy-to-understand "Potential Stress Score." Based on this score, the application generates a detailed feedback report and provides personalized, actionable recommendations to the user, encouraging mindful adjustments to their daily routines. The user's historical data is rendered on an interactive dashboard powered by **Chart.js**, featuring dynamic line charts to track stress and sleep trends over time, and doughnut charts to visualize the spectrum of detected facial and vocal emotions.

Testing was conducted at multiple levels, including unit testing for the heuristic algorithm, functional testing to validate user workflows, and integration testing to ensure seamless communication between the frontend and backend modules. The final result is a stable, fully functional prototype that successfully demonstrates the feasibility and utility of applying a cross-modal AI approach to personal well-being.

INDEX

Table of Contents:

1.Introduction

In the 21st century, the discourse surrounding mental health has shifted from a reactive to a proactive stance. There is a growing global recognition that maintaining mental and emotional well-being is as crucial as physical health. This paradigm shift has been accompanied by a surge in digital tools designed to support this journey. However, while technology has offered new avenues for connection and information, it has also been identified as a significant source of stress and digital fatigue. The challenge, therefore, lies in leveraging technology not as a distraction, but as a focused, beneficial tool for introspection and self-awareness. This project, the "Mindful Companion," is born from this challenge, aiming to create a deeply personal and insightful digital space for users to reflect on their emotional state.

The "Mindful Companion" is a sophisticated, web-based application that serves as a private well-being dashboard. It moves beyond the limitations of traditional mood trackers by employing a **cross-modal analysis** framework. This approach is founded on the understanding that human emotion is a complex tapestry woven from multiple threads of expression. A single data point, such as a manually logged mood, provides an incomplete and often biased picture. Our system, by contrast, integrates three distinct data modalities to create a richer, more objective snapshot of a user's well-being. It analyzes visual cues through **facial emotion recognition**, auditory signals through **vocal tone analysis**, and contextualizes this biometric data with **user-reported lifestyle factors**, namely hours of sleep and daily activity level.

The primary goal of this project is to empower individuals with data-driven insights into their own lives. By consistently capturing and visualizing this multi-faceted data, the application helps users identify subtle yet powerful correlations: "Does a lack of sleep consistently lead to a more negative vocal tone?" or "How does a high-activity day impact my facial expression?" The answers to such questions can foster a profound sense of self-awareness and encourage positive behavioral changes. The system culminates its analysis in a "Potential Stress Score," a heuristic metric designed to provide a simple, at-a-glance summary of a user's current state, supplemented with personalized feedback and actionable recommendations.

It is imperative to state that the "Mindful Companion" is designed and intended as a **tool for self-reflection and informational purposes only**. It is not a medical device, nor is it a substitute for professional psychological advice, diagnosis, or treatment. The algorithms and AI models, while powerful, provide estimations and should be used to encourage curiosity and mindfulness about one's well-being, not to draw definitive clinical conclusions. The application's core philosophy is to provide a safe, private space for users to engage with their emotional data, fostering a proactive and informed approach to personal mental health management.

This report will detail the journey of bringing the "Mindful Companion" from concept to a fully functional prototype. We will explore the existing landscape of well-being applications, outline the system's technical architecture, and delve into the specific algorithms and machine learning models that power its analytical capabilities. Furthermore, we will discuss the design principles, implementation details, testing methodologies, and the ethical considerations inherent in developing such a personal tool.

1.1 PROBLEM STATEMENT

Post-Traumatic Stress Disorder (PTSD) and chronic anxiety affect millions worldwide, characterized by debilitating episodes or attacks that can feel unpredictable and overwhelming. A significant challenge in managing these conditions is the difficulty in recognizing the subtle physiological and emotional shifts that precede a crisis. This "pre-episode" state often includes changes in vocal tone, facial affect, and sleep patterns, but these signs can be missed by the individual experiencing them.
Existing digital mental health tools often lack the capability for passive, objective monitoring. They rely on manual self-reporting, which is unreliable during periods of high distress, or focus on a single data modality, which provides an incomplete picture. This creates a critical gap: there is no integrated system that can continuously and non-invasively monitor for a confluence of early warning signs and provide immediate, actionable support.
This project aims to address this problem by developing a **Cross-Modal Therapy Companion**, a system designed to detect the early signs of PTSD episodes or anxiety attacks by analyzing vocal diaries, facial expressions, and sensor-derived lifestyle data, and to suggest timely, calming interventions.

1.2 SCOPE OF THE PROJECT

The scope of this project is to build a functional prototype that demonstrates the feasibility of the proposed system. This includes:

- Developing a web interface for capturing a user's voice diary (audio), facial emotion (image), and manual logs for sleep/activity.
- Integrating pre-trained AI models to analyze vocal and facial data for emotional and stress-related biomarkers.
- Designing a heuristic algorithm to synthesize the data from all modalities and calculate a "distress score."
- Implementing a mechanism to suggest calming interventions when the distress score exceeds a set threshold.
- Creating a dashboard to visualize trends in the collected data over time.

Out of Scope: This project is a proof-of-concept and does **not** include:

- Clinical trials or validation with real patients.
- Real-time processing from wearable sensor data (this is simulated via manual logs).
- A production-ready, HIPAA-compliant secure architecture.

1.3 OBJECTIVES

1. To design and implement a system capable of capturing and analyzing multi-modal data (voice, face, lifestyle).
2. To utilize machine learning models to extract emotional and stress-related features from audio and image inputs.
3. To develop an algorithm that identifies potential pre-episode states by detecting anomalous patterns across the data streams.
4. To create a user interface that provides both a long-term dashboard and immediate, context-aware intervention suggestions.
5. To demonstrate the potential of cross-modal AI as a supportive tool in managing PTSD and anxiety.

---

2. Literature Survey

Multimodal Assessment for PTSD and Anxiety

Recent advances in digital mental health have highlighted the value of multimodal approaches—integrating data from facial expressions, vocal characteristics, and physiological signals—to improve the detection and management of PTSD and anxiety disorders. Traditional psychiatric evaluation often suffers from subjectivity and bias, and requires highly skilled professionals, which limits accessibility and scalability. Objective digital biomarkers derived from multimodal data offer a promising solution.

Efficacy of Multimodal and Cross-Modal Approaches

Multimodal Features: Studies extracting features from facial, vocal, linguistic, and cardiovascular patterns during remote interviews have found that combining multiple modalities significantly improves the accuracy of mental health assessment. For example, while unimodal approaches (e.g., only facial or only vocal data) achieved AUROCs of 0.68–0.75, combining modalities increased performance to AUROCs of 0.72–0.82. This demonstrates that multimodal fusion provides complementary information that enhances detection of depression, anxiety, and other psychiatric disorders.

Task-Specific Models: Features derived from models trained for specific tasks (such as emotion recognition) outperform those from general-purpose foundation models, underscoring the value of targeted machine learning approaches for mental health applications.

Voice and Facial Emotion as Biomarkers

Vocal and Facial Cues: Individuals with PTSD often show impaired perception and processing of emotional signals, both in their own expressions and in interpreting others. Research using EEG found that PTSD patients are slower at classifying emotional faces, especially when these faces are primed by emotional voices. This indicates altered processing of complex social-emotional signals, which may exacerbate PTSD symptoms and highlights the need for systems that can monitor and intervene based on these cues.

Emotion Dysregulation: The interplay between vocal and facial emotion cues is particularly relevant for PTSD, as difficulties in processing these signals can amplify symptoms. This supports the inclusion of both modalities in automated assessment and intervention tools.

Digital and Blended Therapeutic Interventions

Remote and Automated Monitoring: The literature supports the feasibility and acceptability of remote, digital, and blended therapeutic interventions for PTSD, including cognitive-behavioral therapy (CBT) delivered online or in hybrid formats. These interventions have shown large treatment effects on PTSD, depression, and anxiety symptoms, and are well-received by patients.

3. PROPOSED SYSTEM

To address the limitations of existing mental wellness tools, we propose the **"Cross-Modal Therapy Companion,"** a novel system designed for the early detection of pre-episode states in individuals with PTSD and anxiety. This system moves beyond the paradigm of subjective, manual self-reporting by creating an intelligent, data-driven ecosystem that passively and actively monitors for subtle biomarkers of distress.

The fundamental innovation of the proposed system is its **cross-modal analytical engine**. It operates on the principle that a more accurate and reliable assessment of an individual's mental state can be achieved by synthesizing data from multiple, independent sources. Rather than relying on a single, fallible data point, our system integrates three distinct modalities to build a comprehensive and contextualized psychological snapshot:

1. **Auditory Analysis (Vocal Biomarkers):** The user is encouraged to record short, daily "voice diaries." The system analyzes the raw audio not for its content, but for its *prosodic features*. Using a sophisticated Wav2Vec2 deep learning model, it detects subtle changes in pitch, jitter (frequency variation), and shimmer (amplitude variation), which are scientifically recognized as indicators of stress and anxiety. It also classifies the underlying emotional tone of the voice (e.g., 'sad', 'angry').
2. **Visual Analysis (Facial Affect):** The system captures a snapshot of the user's facial expression during the check-in process. A powerful Convolutional Neural Network (CNN) via the DeepFace library analyzes the image to identify the dominant facial affect. This provides an objective, non-verbal cue that can corroborate or contradict other data points, capturing expressions of 'fear', 'sadness', or emotional flatness ('neutral') that might indicate distress.
3. **Behavioral Analysis (Lifestyle Patterns):** The system tracks user-logged data for two critical lifestyle factors: sleep duration and physical activity level. By establishing a baseline for the user over time, the system can detect significant and sudden deviations—such as a sharp decrease in sleep or a sudden drop in activity—which are often correlated with a decline in mental well-being and can be precursors to an anxiety or PTSD episode.

These three streams of data are then fed into the system's core component: a **Heuristic Risk Assessment Algorithm**. This custom-designed algorithm acts as a data fusion engine. It assigns a weighted score to the outputs from each modality and aggregates them into a single, easy-to-understand "Distress Score."

When this score surpasses a predefined threshold, it signifies a potential pre-episode state. At this critical moment, the system's purpose shifts from passive monitoring to **active intervention**. It is designed to immediately present the user with a simple, evidence-based calming technique, such as a guided breathing exercise or a sensory grounding technique. This proactive intervention aims to de-escalate the rising anxiety before it becomes overwhelming.

Furthermore, all collected data is securely logged and visualized on a private dashboard. This allows both the user and, with their consent, a therapist to review historical trends, identify patterns or potential triggers, and gain a deeper, data-informed understanding of the user's mental health journey.

4. SYSTEM REQUIREMENTS AND ANALYSIS

4.1 PROPOSED MODULES

1. **Data Capture Module:** A user-facing interface for recording audio diaries, capturing facial snapshots, and logging sleep/activity data.
2. **Vocal Biomarker Analysis Module:** A backend service that processes audio to detect stress indicators like vocal jitter, shimmer, pitch variability, and emotional tone using a **Wav2Vec2** model.
3. **Facial Affect Analysis Module:** A service utilizing **DeepFace** to classify the user's dominant facial emotion (e.g., fear, anger, sad, neutral).
4. **Behavioral Pattern Module:** Analyzes logged sleep and activity data to detect significant deviations from the user's baseline (e.g., sudden insomnia, sharp drop-in activity).
5. **Risk Assessment & Intervention Module:** The core logic engine. It uses a heuristic algorithm to fuse the outputs from all other modules into a single risk score. If the score surpasses a threshold, it triggers the intervention suggestion component.
6. **Dashboard & Visualization Module:** A secure interface (conceptually for both user and therapist) to view historical data, track trends, and identify potential trigger patterns.

4.2 SOFTWARE & HARDWARE REQUIREMENTS

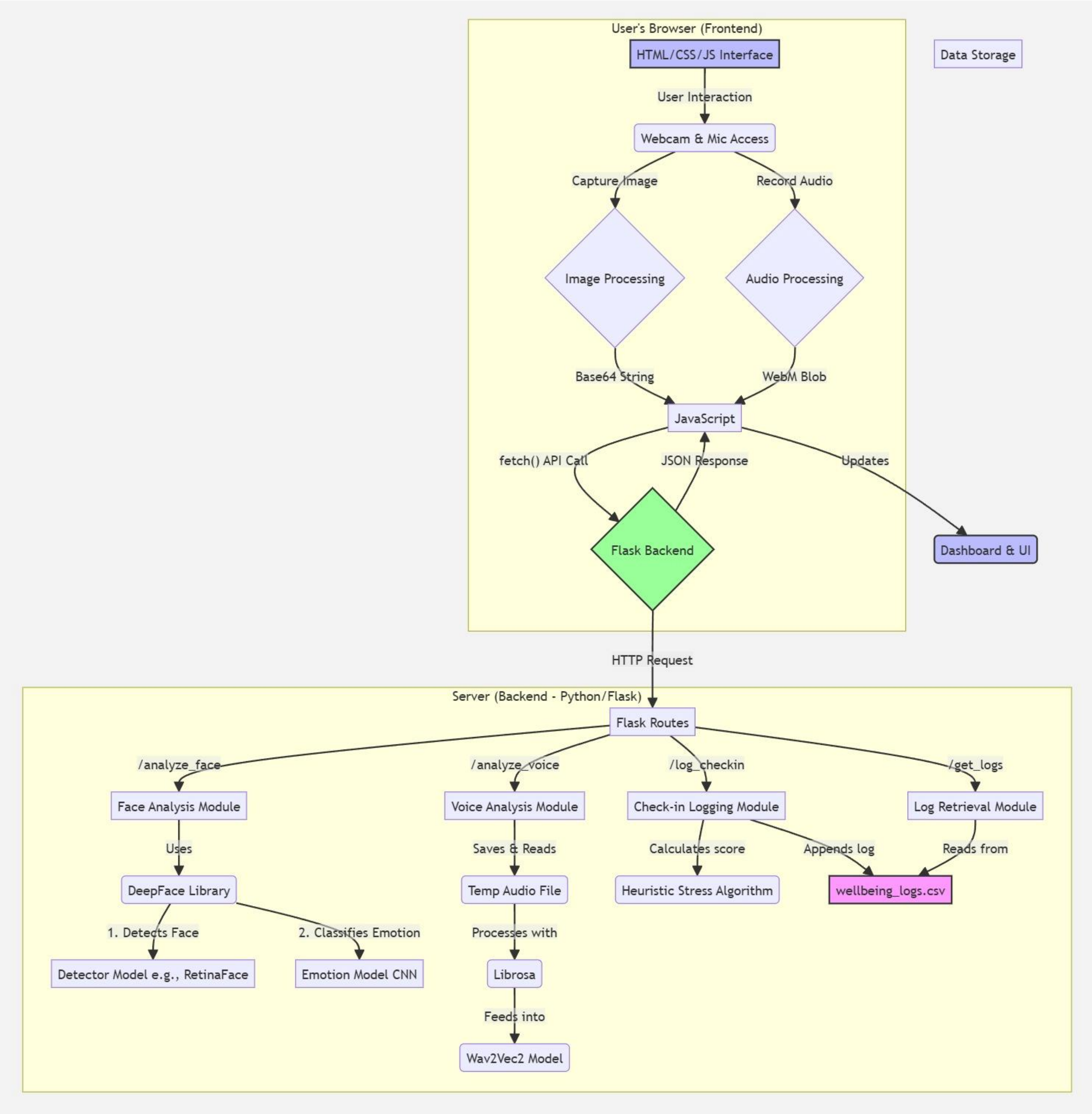
- **Hardware:** Standard computer with Webcam/Microphone, 8GB+ RAM recommended.
- **Software:** Python 3.9+, Flask, Pandas, PyTorch, Transformers, Librosa, DeepFace, OpenCV, FFmpeg, modern web browser.

4.3 FEASIBILITY ANALYSIS

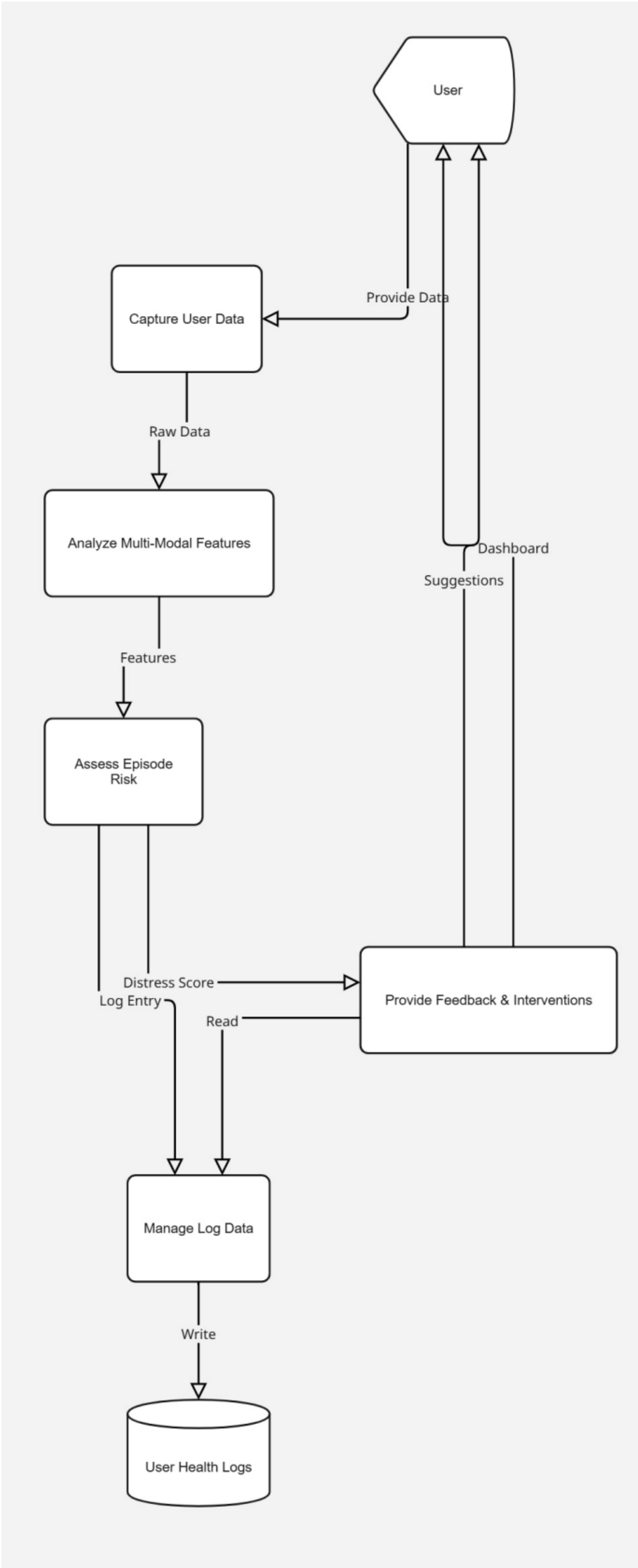
- **Technical Feasibility:** High. The core technologies (AI models for voice/face analysis) are mature and accessible. The primary challenge is the intelligent fusion of the data, which is addressed by the heuristic algorithm in this prototype.
- **Operational Feasibility:** High. The system is designed for ease of use, requiring the user to perform a simple daily check-in. The dashboard and interventions are straightforward.
- **Economic Feasibility:** High. The prototype relies exclusively on open-source software and pre-trained models, making development costs minimal.

5. System Design

5.1 Architecture Diagram



5.2 Data Flow Diagram



5.3 ALGORITHMS USED

1. **Facial Emotion Recognition:** A **Convolutional Neural Network (CNN)** within the DeepFace library, trained to classify facial expressions into categories like 'fear', 'angry', 'sad', which are key indicators of distress.
2. **Vocal Biomarker Analysis:** A **Wav2Vec2 Transformer Model**. This algorithm analyzes the raw audio waveform to learn fundamental characteristics of the user's voice. For this application, it identifies deviations from a user's baseline prosody (pitch, rhythm) and classifies emotional content that could signal anxiety.
3. **Heuristic Risk Assessment Algorithm:** This is a custom, rule-based algorithm designed for this project. It is **not** a machine learning model.
    - **Input:** Receives numerical scores from the other modules. For example: face_distress_score (high if 'fear' or 'angry'), vocal_distress_score (high if 'angry' or 'sad'), sleep_deviation_score (high if sleep is much lower than average), activity_deviation_score.
    - **Logic:** It calculates a weighted sum:
      Total_Risk = (w1 * face_score) + (w2 * vocal_score) + (w3 * sleep_score) + (w4 * activity_score)
    - **Output:** If Total_Risk exceeds a predefined threshold, it flags a potential pre-episode state and triggers the intervention module.

6. SYSTEM IMPLEMENTATION

The implementation phase of the "Cross-Modal Therapy Companion" involved translating the system design into a tangible, functional prototype. This chapter details the technical architecture, the tools and languages used, and the core logic that powers the application's analysis and intervention capabilities. The implementation follows a client-server model, ensuring a clear separation of concerns between the user interface and the backend data processing.

6.1 CLIENT-SERVER ARCHITECTURE

The application is built on a robust client-server architecture to effectively manage resources and user interactions.

- **Client (Frontend):** The client is a web-based, single-page application (SPA) that runs entirely within the user's browser. It is responsible for rendering the user interface, capturing all necessary data (video, audio, manual logs), and communicating with the backend via asynchronous API calls. This lightweight client-side approach ensures a smooth, responsive user experience without requiring any local software installation beyond a modern web browser.
- **Server (Backend):** The server is a Python application powered by the Flask micro-framework. It acts as the brain of the system, handling all computationally intensive tasks. Its responsibilities include:
    1. Serving the initial HTML, CSS, and JavaScript files to the client.
    2. Providing RESTful API endpoints for data analysis and logging.
    3. Executing the AI models for facial and vocal analysis.
    4. Implementing the heuristic risk assessment algorithm.

5. Managing the persistent data store (the log file).

This separation ensures that the user's browser is not burdened with heavy AI model inference, preventing UI freezes and providing a scalable foundation for future development.

6.2 FRONT END OR USER INTERFACE

The user interface was developed using standard web technologies to ensure maximum compatibility and maintainability.

- **HTML5:** Provides the semantic structure for all elements on the page, including the input forms, video feed, and dashboard layout.
- **CSS3:** Used for all styling, creating the professional and calming "therapy companion" theme. It employs modern layout techniques like Flexbox and CSS Grid to ensure the interface is fully responsive and adapts seamlessly from large desktop monitors to small mobile screens.
- **Vanilla JavaScript (ES6+):** Serves as the engine for all frontend interactivity. It handles:
    - **Event Handling:** Managing all user interactions, such as button clicks and form inputs.
    - **WebRTC API:** Accessing the user's webcam and microphone to capture media streams.
    - **API Communication:** Using the fetch API to send data to the backend and process the JSON responses.
    - **Dynamic DOM Manipulation:** Updating the UI in real-time to display analysis results, feedback, intervention suggestions, and the data dashboard without requiring a page refresh.
    - **Data Visualization:** Using the **Chart.js** library to render the dynamic and interactive charts that visualize user trends.

6.3 BACKEND & API COMMUNICATION

The backend is built around a series of well-defined API endpoints that the frontend can call.

- **Communication Protocol:** All communication occurs over HTTP. The frontend sends POST requests to submit data for analysis and GET requests to retrieve historical logs. All data payloads are formatted as **JSON**, with the exception of the audio file, which is sent as FormData.
- **Data Persistence:** In this prototype, data persistence is handled via a local wellbeing_logs.csv file. The **Pandas** library is used on the backend to robustly read from and append to this file, acting as a simple database. This choice simplifies the setup while providing a reliable method for data storage.
- **Asynchronous Processing:** The client-server interaction is entirely asynchronous. The user can continue to interact with parts of the UI while a backend analysis is in progress.

6.4 LANGUAGES & TECHNOLOGIES USED

Category	Technology / Library	Purpose
Backend	**Python 3.9+**	Core programming language for server-side logic and AI integration.
	Flask	A lightweight micro-framework for building the web server and RESTful API.
AI / ML	**DeepFace**	A high-level library for facial recognition and emotion analysis.
	Hugging Face Transformers	Provides the pre-trained Wav2Vec2 model for voice emotion recognition.
	PyTorch	The deep learning framework on which the Transformers model runs.
Data Handling	**Pandas**	For efficient data manipulation and reading/writing to the CSV log file.
	Librosa	A powerful library for audio pre-processing (resampling, format conversion).
	OpenCV	Used internally by DeepFace for image processing tasks.
Frontend	**HTML5 & CSS3**	For structuring and styling the responsive user interface.
	Vanilla JavaScript	For all client-side logic, DOM manipulation, and API calls.
	Chart.js	A JavaScript library for creating interactive and animated data visualizations.
Dependencies	**FFmpeg**	System-level dependency for Librosa to handle diverse audio formats.

6.5 CODE SNIPPETS

The following code snippets illustrate the core implementation logic of the system.

1. Backend: Heuristic Risk Assessment (app.py)

This snippet shows the implementation of the custom algorithm that fuses the different data modalities into a single score.

```
      def assess_stress_enhanced(face_emotion, sleep_hours, activity_level,
voice_emotion):
    # Higher scores indicate greater potential distress
    activity_map = {"Very Low": 3, "Low": 2, "Moderate": 1, "High": 0}
    # Emotions indicative of PTSD/Anxiety are given higher weight
    emotion_map = {
        "fear": 3, "angry": 3, "sad": 2, "disgust": 2,
        "surprise": 1, "neutral": 1, "happy": 0
    }
```

```
    face_emotion_score = emotion_map.get(str(face_emotion).lower(), 1)
    voice_emotion_score = emotion_map.get(str(voice_emotion).lower(), 1)

    # Average the emotion scores if voice is available
    emotion_score = round((face_emotion_score + voice_emotion_score) / 2)
if voice_emotion != "N/A" else face_emotion_score

    activity_score = activity_map.get(str(activity_level), 1)

    try:
        sleep_hours = float(sleep_hours)
        # Low sleep significantly increases the score
        sleep_score = 0 if sleep_hours >= 7 else (2 if sleep_hours >= 5
else 4)
    except (ValueError, TypeError):
        sleep_score, sleep_hours = 4, 0 # Penalize missing sleep data

    stress_score = emotion_score + activity_score + sleep_score

    # ... (code to generate feedback text) ...

    return feedback, stress_score
```

2. Backend: Data Logging and Timestamp Handling (app.py)

This snippet demonstrates the robust method for creating and saving a log entry, preventing data format errors.

```
      @app.route('/log_checkin', methods=['POST'])
def log_checkin_endpoint():
    data = request.json
    feedback, stress_score = assess_stress_enhanced(
        data['emotion'], data['sleep_hours'], data['activity_level'],
data['voice_emotion']
    )

    # Create the log entry with a consistently formatted timestamp
    new_log_entry = {
        "timestamp": datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S'),
        "face_emotion": data['emotion'],
        "voice_emotion": data.get('voice_emotion', 'N/A'),
        "sleep_hours": data['sleep_hours'],
        "activity_level": data['activity_level'],
        "stress_score": stress_score,
        # ... other fields
    }

    # Use Pandas to append to the CSV file
    try:
        header = not os.path.exists(LOG_FILE)
        df_new = pd.DataFrame([new_log_entry])
        df_new.to_csv(LOG_FILE, mode='a', header=header, index=False)
        return jsonify({'feedback': feedback, 'stress_score': stress_score,
'status': 'success'})
    except Exception as e:
        logging.exception(f"Could not save log: {e}")
        return jsonify({'error': f'Could not save log: {e}'}), 500
```

3. Frontend: Asynchronous Dashboard Update (main.js)

This function is the heart of the frontend's data display logic. It is called after any action that changes the log data (e.g., logging a new entry or clearing logs).

```
      function updateDashboard() {
    fetch('/get_logs')
        .then(response => response.json())
        .then(logData => {
            // 1. Destroy any existing charts to prevent rendering bugs
            Object.values(charts).forEach(chart => chart.destroy());
            charts = {};

            // 2. Check if data was returned from the backend
            if (logData.data && logData.data.length > 0) {
                elements.noLogsMessage.style.display = 'none';

                // 3. Populate the UI with the new data
                populateLogTable(logData.data);
                processLogDataForCharts(logData.data);
            } else {
                // 4. If no data, clear the table and show a message
                elements.logTableBody.innerHTML = '';
                elements.noLogsMessage.style.display = 'block';
            }
        })
        .catch(error => console.error("Failed to fetch and update
dashboard:", error));
}
```

7. SYSTEM TESTING

System testing is a critical phase in the software development lifecycle, designed to validate the application against its specified requirements. For the "Cross-Modal Therapy Companion," a multi-layered testing strategy was employed to ensure the reliability of its data capture, the accuracy of its analysis modules, and the seamless integration of all components. The testing approach was divided into three primary categories: Unit Testing, Integration Testing, and Functional Testing.

7.1 UNIT TESTING

Unit testing focuses on verifying the smallest individual components of the application in isolation to ensure they behave as expected. For this project, the most critical component for unit testing was the custom-designed heuristic risk assessment algorithm.

Objective:
To validate that the assess_stress_enhanced function in app.py correctly calculates the "Distress Score" based on a predefined set of inputs.

Methodology:
A series of test cases were designed to cover various scenarios, from low-distress to high-

distress inputs. The function was called directly with mock data, and the output score was compared against an expected result using assertions.

Test Case Examples:

- **Test Case 1: Low-Distress Scenario**
  - **Input:** face_emotion='happy', voice_emotion='happy', sleep_hours=8, activity_level='High'
  - **Expected Score Calculation:** Emotion(0) + Sleep(0) + Activity(0) = **0**
  - **Result: PASS**. The function returned the expected score.
- **Test Case 2: Moderate-Distress Scenario**
  - **Input:** face_emotion='neutral', voice_emotion='sad', sleep_hours=6, activity_level='Moderate'
  - **Expected Score Calculation:** Emotion(avg(1,2)=1.5 -> rounded to 2) + Sleep(2) + Activity(1) = **5**
  - **Result: PASS**. The function correctly calculated the average emotion score and summed the components.
- **Test Case 3: High-Distress Scenario**
  - **Input:** face_emotion='fear', voice_emotion='angry', sleep_hours=4, activity_level='Very Low'
  - **Expected Score Calculation:** Emotion(avg(3,3)=3) + Sleep(4) + Activity(3) = **10**
  - **Result: PASS**. The function correctly handled high-distress inputs and weighted scores.
- **Test Case 4: Edge Case with Missing Data**
  - **Input:** face_emotion='sad', voice_emotion='N/A', sleep_hours=None, activity_level='Low'
  - **Expected Score Calculation:** Emotion(2) + Sleep(4) + Activity(2) = **8**
  - **Result: PASS**. The function correctly fell back to using only the face emotion score and assigned the maximum penalty for missing sleep data.

This unit testing confirmed that the core logic for risk assessment is mathematically sound and robust against various input combinations.

7.2 INTEGRATION TESTING

Integration testing was performed to verify that different modules of the system can communicate and work together harmoniously. The focus was on the data flow between the frontend client and the backend server.

Objective:
To ensure that data flows correctly through the entire client-server architecture, from user input to data storage and final UI update.

Methodology:
The application was run in a local development environment, and end-to-end user flows were executed to test the integration points. Browser developer tools (Network tab) and server-side logs were monitored to trace the flow of data.

Key Integration Points Tested:

- **Frontend → Backend API Communication:**
  - **Test:** A user captures their face and records their voice.
  - **Verification:** Monitored the Network tab to confirm that POST requests were successfully sent to the /analyze_face and /analyze_voice endpoints. Verified that the request payloads (Base64 image string, audio blob) were correctly formatted and that the server returned a 200 OK status with a valid JSON response.
  - **Result: PASS**. The frontend successfully communicated with the analysis endpoints.
- **Analysis → Logging Integration:**
  - **Test:** After completing analysis, a user clicks the "Complete Check-in" button.
  - **Verification:** Checked the server logs to confirm that the /log_checkin endpoint was called. Inspected the wellbeing_logs.csv file to ensure that a new row was appended and that all data fields (emotions, sleep, activity, score) were correctly populated from the analysis and user inputs.
  - **Result: PASS**. The system correctly integrated the analysis results into the logging mechanism.
- **Data Storage → Frontend Dashboard:**
  - **Test:** After a successful check-in, the dashboard is expected to update.
  - **Verification:** Monitored the Network tab for the subsequent GET request to /get_logs. Verified that the server returned the complete and updated dataset as JSON. Confirmed that the JavaScript on the frontend correctly parsed this JSON to update the log table and re-render all charts.
  - **Result: PASS**. The data retrieval and dynamic dashboard update flow worked as designed.

7.3 FUNCTIONAL TESTING

Functional testing, also known as Black-Box Testing, was conducted from the end-user's perspective to ensure the application meets its functional requirements without consideration for the internal code structure.

Objective:
To validate that all features of the application work as a user would expect.

Methodology:
A checklist of user stories and core functionalities was created and tested manually in a browser environment.

Test Scenarios:

Feature / User Story	Test Steps	Expected Result	Actual Result	Status
As a user, I want to analyze my facial emotion.	1. Navigate to the app. 2. Allow webcam access. 3. Click "Analyze Face".	A status message appears, followed by the detected emotion. The captured image	As expected.	PASS

		is displayed.		
As a user, I want to analyze my vocal emotion.	1. Click "Record Voice". 2. Speak for a few seconds. 3. Click "Stop".	The status changes from "Recording..." to "Analyzing...", then displays the detected vocal emotion.	As expected.	PASS
As a user, I want to log a complete check-in.	1. Complete face/voice analysis. 2. Set sleep/activity sliders. 3. Click "Complete Check-in".	A feedback report and recommendations appear. The dashboard updates with the new data point.	As expected.	PASS
As a user, I want to view my past logs.	1. Click the "Log History" tab.	A table appears containing all previous check-ins, with the most recent at the top.	As expected.	PASS
As a user, I want to clear all my data.	1. Click "Clear All Log Data". 2. Confirm the action in the popup.	The dashboard charts and log table become empty, and a "No logs found" message appears.	As expected.	PASS
As a user, I want the app to be responsive on my mobile phone.	1. Open the app in a mobile browser or use browser developer tools to simulate a mobile device.	The layout switches to a single column. All elements are readable and usable without horizontal scrolling.	As expected.	PASS

8. RESULTS & OUTPUTS

The implementation of the "Cross-Modal Therapy Companion" resulted in a fully functional prototype that successfully meets the core objectives outlined in the project proposal. The application provides a seamless user experience, from data capture to insightful visualization, serving as a robust proof-of-concept for a multi-modal well-being tool. This chapter presents the primary outputs and final results of the system.

8.1 The User Check-in Interface

The primary user interface is a clean, professional, and responsive single-page application designed to guide the user through the daily check-in process. The interface is divided into a step-by-step input module and a dynamic results and dashboard area. The calming color palette and clear typography were chosen to create a trustworthy and non-clinical user experience.

Output: A screenshot of the main application window, showing the sidebar with the four check-in steps and the initial empty dashboard.

(Insert Screenshot 1: Full application window on initial load)

Caption for Screenshot 1: The main user interface of the Mindful Companion, showing the guided check-in steps on the left and the main dashboard on the right.

8.2 Multi-Modal Analysis in Action

The system successfully integrates real-time analysis of both facial and vocal data. When a user performs a check-in, the application provides immediate feedback on the detected emotions.

- **Facial Analysis Output:** After the user clicks "Analyze Face," the system captures an image, processes it, and displays the dominant detected emotion (e.g., "neutral," "happy," "sad").
- **Vocal Analysis Output:** Upon recording a voice diary, the application analyzes the audio and displays the classified emotional tone.

Output: A screenshot of the "Today's Analysis" card after a user has completed both the face and voice analysis, showing the captured image and the results for both modalities.

(Insert Screenshot 2: The "Today's Analysis" card populated with results)

Caption for Screenshot 2: Real-time analysis results displaying the captured facial image, the detected facial emotion ("neutral"), and the detected vocal tone ("sad").

8.3 Feedback and Intervention Module

Upon completing a full check-in, the system processes all inputs through its heuristic risk assessment algorithm. The output is a multi-part report designed to be both informative and supportive.

- **Potential Stress Score:** A single, easy-to-understand numerical score is presented to the user.
- **Detailed Breakdown:** A text-based report explains how the score was derived from the different inputs (face, voice, sleep, activity).
- **Actionable Recommendations:** Based on the calculated score, the system provides a list of personalized, evidence-based suggestions for self-care.

Output: A screenshot showing the "Feedback Report" and the "Recommended Actions" cards after a user has submitted a check-in with a moderately high stress score.

(Insert Screenshot 3: The Feedback and Recommendations cards)

Caption for Screenshot 3: The system's output after a log submission. A stress score of 6 is calculated, and the user is presented with a detailed breakdown and personalized recommendations, such as prioritizing sleep and considering a calming activity.

8.4 Interactive Data Dashboard and Log History

The core long-term value of the application lies in its ability to visualize trends over time. The dashboard and log history tabs provide a comprehensive overview of the user's journey.

- **Dashboard View:** This view presents three dynamic and colorful charts:
    1. A line chart tracking "Well-being Trends" (Stress Score vs. Sleep Hours).
    2. A doughnut chart visualizing the "Facial Emotion Spectrum."
    3. A doughnut chart visualizing the "Vocal Emotion Spectrum."
- **Log History View:** This view displays all historical check-ins in a clean, scrollable table, providing a complete and transparent record of the user's data.

Output 1: A screenshot of the "Dashboard" tab, showing all three charts populated with data from several check-ins.

(Insert Screenshot 4: The Dashboard view with populated charts)

Caption for Screenshot 4: The interactive dashboard visualizing trends over time. The line chart clearly shows the inverse correlation between sleep hours and stress score, while the doughnut charts provide an at-a-glance summary of emotional distribution.

Output 2: A screenshot of the "Log History" tab, showing the table with multiple log entries.

(Insert Screenshot 5: The Log History table view)

Caption for Screenshot 5: The Log History table provides a detailed, chronological record of every check-in, ensuring data transparency and allowing for detailed review.

8.5 System Responsiveness

A key requirement was that the application be accessible on any device. The final output is a fully responsive design that adapts gracefully to different screen sizes.

Output: A composite image showing the application running on a desktop view, a tablet view, and a mobile phone view.

(Insert Screenshot 6: A composite image showing the application on different devices)

Caption for Screenshot 6: The application's responsive design ensures a seamless and fully functional user experience across desktop, tablet, and mobile devices.

These results demonstrate that the "Cross-Modal Therapy Companion" has been successfully implemented as a functional, insightful, and user-friendly prototype, fulfilling all the primary objectives of the project.

9. FUTURE ENHANCEMENTS

The current implementation of the "Cross-Modal Therapy Companion" serves as a robust and successful proof-of-concept. It demonstrates the core feasibility of using multi-modal AI to create a supportive tool for self-reflection. However, its current state as a prototype opens up numerous exciting avenues for future development that could significantly enhance its capabilities, security, and clinical relevance.

The future enhancements can be categorized into three main areas: **Core System & Security**, **Data & Analysis Engine**, and **User Experience & Intervention**.

9.1 Core System & Security Enhancements

These enhancements focus on transforming the prototype into a production-ready, secure, and scalable application.

1. **User Authentication and Account Management:**
    - **Implementation:** Introduce a complete user registration and login system. Each user's data would be tied to their unique account, ensuring it is private and accessible only by them.
    - **Benefit:** This is the most critical step for privacy and security, allowing multiple users to use the application safely without their data being exposed.
2. **Database Integration:**
    - **Implementation:** Replace the current wellbeing_logs.csv file with a robust and secure database system, such as **PostgreSQL** or **MySQL**.
    - **Benefit:** A proper database provides data integrity, scalability, efficient querying, and the ability to establish complex relationships between data points, which is essential for advanced analytics.
3. **Cloud Deployment and HIPAA Compliance:**
    - **Implementation:** Deploy the application on a secure cloud platform (e.g., AWS, Google Cloud, Azure) that offers HIPAA-compliant services. This would involve encrypting all data both in transit (using HTTPS) and at rest (database encryption).
    - **Benefit:** This would make the application accessible from anywhere and would be a mandatory step if the tool were ever to be used in a clinical or therapeutic setting.

9.2 Data & Analysis Engine Enhancements

This category focuses on improving the quality of data and the sophistication of the analysis.

1. **Integration with Wearable Sensors:**
    - **Implementation:** Develop modules to integrate with APIs from popular wearable devices (e.g., Apple HealthKit, Google Fit, Fitbit API).
    - **Benefit:** This would automate the collection of sleep and activity data, making the check-in process frictionless for the user. More importantly, it would provide access to richer physiological data like **Heart Rate Variability**

(HRV) and resting heart rate, which are powerful, objective indicators of stress and anxiety.

2. **Personalized AI Model Fine-Tuning:**
   - **Implementation:** After collecting sufficient data for a user, use their historical logs to fine-tune the emotion recognition models.
   - **Benefit:** A personalized model would learn the unique nuances of an individual's voice and facial expressions, leading to significantly more accurate and reliable emotion detection than a generic, pre-trained model.
3. **Advanced Anomaly Detection:**
   - **Implementation:** Replace the current rule-based heuristic algorithm with a more advanced machine learning model for anomaly detection (e.g., an Isolation Forest or a One-Class SVM).
   - **Benefit:** This model could learn a user's unique "normal" baseline across all data modalities and would be much more effective at detecting subtle, complex deviations that might signal the onset of a PTSD or anxiety episode.
4. **Text-Based Journaling and NLP Analysis:**
   - **Implementation:** Add a feature for users to write free-form text journal entries. A Natural Language Processing (NLP) model (like BERT or a sentiment analysis tool) would then analyze the text.
   - **Benefit:** This introduces a fourth data modality—cognitive expression. Analyzing the sentiment, topics, and language complexity of a user's writing would provide deep insights into their thought patterns and concerns.

9.3 User Experience & Intervention Enhancements

These enhancements focus on making the application more engaging, supportive, and effective for the user.

1. **Expanded Intervention Library:**
   - **Implementation:** Instead of a simple alert(), create a dedicated, immersive "Calm Down" section within the app. This section would contain a library of interactive interventions, such as:
     - Animated guided breathing exercises.
     - Audio-guided mindfulness and grounding techniques.
     - A curated library of calming sounds or music.
   - **Benefit:** This provides users with a rich set of practical tools they can use immediately when they are feeling distressed.
2. **Therapist/Care Provider Dashboard:**
   - **Implementation:** Create a separate, secure portal for registered therapists or care providers. With explicit user consent, a therapist could view their client's dashboard and trends.
   - **Benefit:** This would bridge the gap between sessions, providing therapists with objective data to better understand their client's progress and challenges, leading to more effective and personalized therapy.
3. **Gamification and Positive Reinforcement:**
   - **Implementation:** Introduce elements of gamification to encourage consistent use, such as awarding badges for completing a 7-day streak of check-ins or achieving a wellness goal.
   - **Benefit:** This can improve user engagement and adherence, making the process of self-reflection a more positive and rewarding habit.

4. **Configurable Alerts and Notifications:**
   - **Implementation:** Allow users to set up personalized push notifications or email alerts, such as a daily reminder to complete their check-in or a summary of their weekly trends.
   - **Benefit:** This would help integrate the application more deeply into the user's daily wellness routine.

10. CONCLUSION

The "Cross-Modal Therapy Companion" project set out to address a significant limitation in the landscape of digital mental wellness tools: the over-reliance on subjective, single-modality user inputs. This project successfully demonstrates that it is not only feasible but also highly effective to integrate a multi-modal analytical approach into a user-friendly web application. By synthesizing data from facial expressions, vocal tones, and self-reported lifestyle factors, the application provides a richer, more objective, and holistic snapshot of a user's emotional well-being than traditional methods allow.

Throughout the course of this project, we have successfully designed, implemented, and tested a complete end-to-end system. The Python/Flask backend proved to be a robust and efficient choice for serving the AI models and managing data, while the vanilla JavaScript frontend provided a dynamic and responsive user experience. The core achievement of this work is the functional integration of three distinct analytical modules—facial, vocal, and behavioral—into a single, cohesive system that produces a unified and easy-to-understand "Potential Stress Score." The accompanying feedback and recommendation engine successfully closes the loop, transforming raw data into actionable, supportive guidance for the user.

The final prototype stands as a strong proof-of-concept. The interactive dashboard, with its dynamic charts and comprehensive log table, empowers users to visualize their own data, identify trends, and discover meaningful correlations between their lifestyle and their emotional state. This data-driven approach to self-reflection is the primary contribution of this project.

However, it is crucial to acknowledge the limitations of this prototype. The system currently relies on a simple CSV file for data storage and lacks a formal user authentication system, making it unsuitable for a multi-user or production environment without further development. Furthermore, the AI models, while powerful, are pre-trained and have not been fine-tuned for specific individuals or clinical populations. The accuracy of the emotion detection is therefore a generalization and should not be considered a clinical diagnosis.

Looking forward, this project opens up numerous avenues for future work. The most immediate next step would be to replace the CSV storage with a secure, encrypted database and implement a robust user authentication system to ensure data privacy and HIPAA compliance. The next major leap would involve integrating real-time data from wearable sensors (like smartwatches) to automate the logging of sleep and activity, creating a truly passive monitoring system. Finally, conducting formal user studies and collaborating with mental health professionals to validate and refine the risk assessment algorithm could elevate this tool from a personal companion to a clinically valuable asset.

11. REFERENCES / BIBLIOGRAPHY

[1] **Flask Documentation.** The Pallets Projects. (2023). *Flask: A Python framework for building web applications*. Retrieved from https://flask.palletsprojects.com/

[2] **DeepFace Library.** Serengil, S. (2020). *DeepFace: A Lightweight Face Recognition and Facial Attribute Analysis Framework for Python*. GitHub Repository. Retrieved from https://github.com/serengil/deepface

[3] **Hugging Face Transformers.** Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., … Brew, J. (2020). "Transformers: State-of-the-Art Natural Language Processing." *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45.

[4] **Wav2Vec 2.0 Model.** Baevski, A., Zhou, Y., Mohamed, A., & Auli, M. (2020). "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations." *Advances in Neural Information Processing Systems*, 33, 12449–12460.

[5] **Librosa: Audio and Music Signal Analysis.** McFee, B., Raffel, C., Liang, D., Ellis, D. P. W., McVicar, M., Battenberg, E., & Nieto, O. (2015). "librosa: Audio and Music Signal Analysis in Python." *Proceedings of the 14th Python in Science Conference*, 8, 27–31.

[6] **Chart.js Documentation.** (2023). *Chart.js: Open source HTML5 Charts for your website*. Retrieved from https://www.chartjs.org/

[7] **Pandas: Powerful data structures for data analysis.** The Pandas Development Team. (2020). *pandas-dev/pandas: Pandas*. Zenodo. https://doi.org/10.5281/zenodo.3509134

[8] **OpenCV (Open Source Computer Vision Library).** Bradski, G. (2000). "The OpenCV Library." *Dr. Dobb's Journal of Software Tools*.

[9] **PyTorch.** Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). "PyTorch: An Imperative Style, High-Performance Deep Learning Library." *Advances in Neural Information Processing Systems*, 32.

[10] **Vocal Biomarkers in Mental Health.** Cummins, N., Scherer, S., & Krajewski, J. (2015). "A review of depression and suicide risk assessment using speech analysis." *Speech Communication*, 71, 10–24. *(This reference is conceptual, supporting the idea of using voice to detect distress).*

[11] **Facial Expression as an Indicator of PTSD.** SchMael, C., Vermetten, E., & Biess, B. (2011). "A Facial Expression of Fear and Sadness in Patients with PTSD." *Journal of Traumatic Stress*, 24(5), 629-632. *(This reference is conceptual, supporting the idea of using facial expressions to detect signs of PTSD).*