**PEERHIVE - A THREE-ZONE SYSTEM FOR CLASSIFYING TEXTUAL EMOTION STATES**

**A SOCIALLY RELEVANT MINI PROJECT REPORT**

***Submitted by***

**JASON DAVID MOSES S [211423104241]**

**GNYANPRAKHASH M [211423104169]**

***in partial fulfillment for the award of the degree of***

**BACHELOR OF ENGINEERING**

***IN***

# COMPUTER SCIENCE AND ENGINEERING

****

**PANIMALAR ENGINEERING COLLEGE**

**(An Autonomous Institution, Affiliated to Anna University, Chennai)**

**October 2025**

**PANIMALAR ENGINEERING COLLEGE**

**(An Autonomous Institution, Affiliated to Anna University, Chennai)**

**BONAFIDE CERTIFICATE**

Certified that this project report **“PEERHIVE - A THREE-ZONE SYSTEM FOR CLASSIFYING TEXTUAL EMOTION STATES”** is the Bonafide work of **JASON DAVID MOSES S [211423104241],** **GNYANPRAKHASH [211423104169],** who carried out the project work under my supervision.

**Signature of the HOD with date Signature of the Supervisor with date**

**Dr L.JABASHEELA M.E., Ph.D., Mr P.ALWIN INFANT M.Tech.,(Ph.D)**

Professor and Head,

Department of Computer Science and Engineering,

Panimalar Engineering College, Chennai- 600123

Associate Professor,

Department of Computer Science and

Engineering,

Panimalar Engineering College,

Chennai- 600123

Submitted for the 23CS1512 – Socially Relevant Mini Project Viva– Voce examination held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**DECLARATION BY THE STUDENT**

We **“JASON DAVID MOSES S [211423104241], GNYANPRAKHASH [211423104169]”** hereby declare that this project report titled **“PEERHIVE - A THREE-ZONE SYSTEM FOR CLASSIFYING TEXTUAL EMOTION STATES”** under the guidance of **Mr. P. ALWIN INFANT M.Tech., (Ph.D),** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

**JASON DAVID MOSES S [211423104241]**

**GNYANPRAKHASH [211423104169]**

**ACKNOWLEDGEMENT**

Our profound gratitude is directed towards our esteemed Secretary and

Correspondent, **Dr. P. CHINNADURAI, M.A., Ph.D.,** for his fervent

encouragement. His inspirational support proved instrumental in galvanizing our

efforts, ultimately contributing significantly to the successful completion of this

project

We want to express our deep gratitude to our Directors, Tmt.

**C. VIJAYARAJESWARI, Dr. C. SAKTHI KUMAR, M.E., Ph.D.,** and **Dr.**

**SARANYASREE SAKTHI KUMAR, B.E., M.B.A., Ph.D.,** for graciously

affording us the essential resources and facilities for undertaking this project.

Our gratitude is also extended to our Principal, **Dr. K. MANI, M.E., Ph.D.**, whose facilitation proved pivotal in the successful completion of this project.

We express our heartfelt thanks to **Dr. L. JABASHEELA, M.E., Ph.D.,** Head of

the Department of Computer Science and Engineering, for granting the necessary facilities that contributed to the timely and successful completion of project.

We would like to thank our Project Coordinators **Dr. P. DEEPA M.E., Ph.D., Mrs. Lincy Jemina ,M.E.,(Ph.D)** and Project Guide **Mr P.ALWIN INFANT M.Tech., (Ph.D)** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

**JASON DAVID MOSES S [211423104241]**

**GNYANPRAKHASH [211423104169]**

**ABSTRACT**

The increasing prevalence of mental health discussions on social media platforms presents a valuable opportunity for the automated identification of distress indicators like burnout. Monitoring student well-being is crucial, yet traditional methods often lack scalability or rely on self-reporting. While generic emotion classifiers exist, they often fail to capture the specific, contextually relevant nuances differentiating transient stress from the persistent states indicative of burnout.

To address this, this study proposes a novel three-zone framework (Calm, Stressed, Overwhelmed) designed to categorise user-generated text along a burnout-relevant spectrum. This schema offers a more targeted lens than standard emotion labels. We demonstrate this approach by mapping the 28 fine-grained emotions from the large-scale GoEmotions dataset. This reclassification yielded a tailored dataset but highlighted a significant class imbalance, with Calm heavily dominating.

Using this reclassified data, a DistilBERT transformer model was fine-tuned. Recognising the class imbalance, a weighted cross-entropy loss function was implemented to ensure the model adequately learned features from the critical minority classes (Stressed and Overwhelmed). The model achieved a promising baseline accuracy of 74.19% and a macro F1-score of 62.21% on the full dataset. These results validate the feasibility of adapting pre-trained language models using our zone-based framework for interpreting burnout-related emotional states in online discourse. This approach serves as a strong foundation for developing more effective and scalable tools for mental health monitoring within academic communities.

**TABLE OF CONTENTS**

| **CHAPTER**  **NO** | **TITLE** | **PAGE**  **NO** |
| --- | --- | --- |
|  | **ABSTRACT** | v |
|  | **LIST OF TABLES** | viii |
|  | **LIST OF FIGURES** | viii |
|  | **LIST OF ABBREVIATIONS** | x |
| **1.** | **INTRODUCTION** | 1 |
|  | 1.1 Overview | 2 |
|  | 1.2 Problem Definition | 3 |
| **2.** | **LITERATURE REVIEW** | 4 |
| **3.** | **THEORETICAL BACKGROUND** | 7 |
|  | 3.1 Implementation Environment | 8 |
|  | 3.2 Proposed Methodology | 9 |
|  | 3.2.1 Data Set Description | 9 |
|  | 3.2.2 Input Design (UI) | 9 |
|  | 3.2.3 System Architecture | 10 |
|  | 3.2.4 Module Design | 11 |
| **4.** | **SYSTEM IMPLEMENTATION** | 16 |
|  | 4.1 Modules | 17 |
|  | 4.1.1 Data Collection | 17 |
|  | 4.1.2 Data Cleaning | 17 |
|  | 4.1.3 Data Pre-Processing | 18 |
|  | 4.1.4 Feature Selection | 18 |
|  | 4.1.5 Machine Learning Prediction | 19 |
| **5.** | **RESULTS & DISCUSSION** | 20 |
|  | 5.1 Testing | 21 |
|  | 5.1.1 Unit Testing | 21 |
|  | 5.1.2 Integration Testing | 22 |
|  | 5.1.3 Functional Testing | 23 |
|  | 5.1.4 System Testing | 23 |
|  | 5.1.5 User Accepting Testing | 24 |
|  | 5.1.6 Test Cases and Result | 24 |
|  | 5.2 Result & Discussion | 27 |
|  | 5.2.1 Model Performance Metrics | 27 |
|  | 5.2.2 Visual Analysis of Results | 28 |
|  | 5.2.3 Discussion of Results | 30 |
|  | 5.2.4 PeerHive Application Demo | 31 |
| **6.** | **CONCLUSION & FUTURE WORK** | 33 |
|  | 6.1 Conclusion | 34 |
|  | 6.2 Future Work | 35 |
|  | **APPENDICES** | 37 |
|  | A.1 SDG Goals | 37 |
|  | A.2 Source Code | 38 |
|  | A.3 Screenshots | 39 |
|  | A.4 Paper Publication | 41 |
|  | A.4 Plagiarism Report | 42 |
|  | **REFERENCES** | 43 |

**LIST OF TABLES**

| **TABLE NO** | **NAME** | **PAGE NO** |
| --- | --- | --- |
| 5.1.1 | Unit Testing Examples | 22 |
| 5.1.6 | Test Cases and Results | 25 |
| 5.2.1 | PeerHive Model Performance Metrics | 27 |

**LIST OF FIGURES**

| **FIGURE NO** | **NAME** | **PAGE NO** |
| --- | --- | --- |
| 3.2.3.1 | System Architecture Diagram | 10 |
| 3.2.4.1 | Use Case Diagram | 12 |
| 3.2.4.2 | Sequence Diagram (Text Analysis) | 13 |
| 3.2.4.3 | Activity Diagram (User Sign-up/Login) | 16 |
| 3.2.4.4 | Class Diagram | 15 |
| 5.2.2.1 | Emotional Zone Distribution | 28 |
| 5.2.2.2 | Confusion Matrix | 29 |
| 5.2.2.3 | t-SNE Visualization | 30 |
| A.3.1 | PeerHive - Community Feed | 35 |
| A.3.2 | PeerHive - Conceptual Insights Dashboard | 35 |
| A.3.3 | PeerHive - Goals Tracker |  |
| A.3.4 | PeerHive - Admin Dashboard | 37 |

**LIST OF ABBREVIATIONS**

| **ABBREVIATION** | **DESCRIPTION** |
| --- | --- |
| AI | Artificial Intelligence |
| API | Application Programming Interface |
| BaaS | Backend-as-a-Service |
| BERT | Bidirectional Encoder Representations from Transformers |
| BiLSTM | Bidirectional Long Short-Term Memory |
| CLIP | Contrastive Language–Image Pre-training |
| DFD | Data Flow Diagram |
| DistilBERT | Distilled Bidirectional Encoder Representations from Transformers |
| EDM | Educational Data Mining |
| F1-Score | Harmonic Mean of Precision and Recall |
| FFE | Fast Fourier Transform |
| Flask | Python Web Framework |
| GPU | Graphics Processing Unit |
| LA | Learning Analytics |
| LORA | Low-Rank Adaptation |
| LMS | Learning Management System |
| ML | Machine Learning |
| NLP | Natural Language Processing |
| NoSQL | Not Only SQL (Database type) |
| RAM | Random Access Memory |
| RNN | Recurrent Neural Network |
| SDG | Sustainable Development Goal |
| SIS | Student Information System |
| SSD | Solid State Drive |
| SVM | Support Vector Machine |
| T-SNE | t-Distributed Stochastic Neighbor Embedding |
| UI | User Interface |
| UML | Unified Modeling Language |
| UAT | User Acceptance Testing |
| URL | Uniform Resource Locator |
| VRAM | Video Random Access Memory |

**INTRODUCTION**

**CHAPTER 1**

**1. INTRODUCTION**

**1.1 OVERVIEW**

The contemporary educational landscape, characterized by increasing academic pressures and significant reliance on digital communication, presents a complex student experience. Student mental well-being in higher education is a paramount concern, with growing rates of persistent stress, anxiety, and academic burnout impacting academic performance, health, and broader campus communities. These challenges can lead to reduced engagement, attrition, and long-term health consequences.

Concurrently, the pervasive adoption of online social platforms and communication channels has transformed how students interact and express themselves. These digital spaces serve as vast, often anonymous, repositories of user-generated text where students frequently articulate feelings related to their academic journey, including subtle or overt signs of distress. This proliferation of authentic online discourse presents an unprecedented opportunity for developing **scalable, non-intrusive, and real-time methods for mental health monitoring and support**. Early identification of struggling students is critically important for enabling timely and effective interventions.

However, effectively harnessing this rich textual data for well-being assessment requires a sophisticated approach. Generic sentiment analysis often proves inadequate, failing to capture the specific nuances, intensity, and contextual relevance of emotions pertinent to academic burnout, such as feelings of exhaustion, cynicism, or reduced personal accomplishment. This highlights the need for more granular interpretation.

Recognizing these opportunities and challenges, the **PeerHive project** proposes and implements an innovative system. PeerHive leverages advanced Natural Language Processing (NLP) techniques, specifically fine-tuned transformer models, within a contextually relevant three-zone framework (Calm, Stressed, Overwhelmed) to analyze student-generated text for burnout indicators. This report details PeerHive's development, implementation, and rigorous evaluation, showcasing its potential to provide actionable insights into student well-being online, fostering a more supportive academic environment that aligns with the "Support, not Surveillance" principle.

**1.2 PROBLEM DEFINITION**

Despite increasing awareness and dedicated efforts by educational institutions, current student mental health monitoring and support systems often grapple with fundamental limitations, leaving a critical gap in proactive well-being management.

A primary issue is that traditional methods, such as periodic mental health surveys, direct counseling appointments, or self-reporting mechanisms, are predominantly reactive and resource-intensive. Students often only seek help or are identified after their stress or burnout has already significantly escalated, impacting their academic performance, social engagement, and personal health. These methods are inherently limited in their capacity for continuous, real-time assessment and struggle with the scalability required to effectively monitor a large, diverse, and geographically dispersed student population.

Furthermore, the existing landscape of automated textual analysis tools falls short when applied to the complex task of detecting academic burnout. Most tools are designed for general sentiment analysis, which provides only a superficial understanding. A "negative" classification may encompass everything from minor annoyance to severe depression, failing to differentiate between temporary academic pressure, chronic stress, or the distinct signs of overwhelm characteristic of burnout. Such generic models lack the contextual and semantic granularity necessary to accurately interpret linguistic expressions of distress, diminishing their utility for targeted support.

Crucially, any system attempting to monitor student well-being must navigate profound ethical considerations and privacy concerns. Students are often hesitant to disclose sensitive personal information if they perceive a risk of surveillance, judgment, or misuse of their data. Therefore, the development of such tools must rigorously adhere to principles of anonymity, data security, and aggregated insights, ensuring that the focus remains unequivocally on collective well-being support rather than individual tracking or punitive measures.

In summary, the pervasive problem is the absence of a scalable, contextually informed, and privacy-preserving system capable of accurately identifying and classifying burnout-related emotional states from the unstructured text of student online discourse. This gap prevents institutions from offering timely, proactive, and effective well-being support. PeerHive directly addresses this by proposing a novel framework and application that provides a more granular, actionable, and ethically sound interpretation of student emotional data than currently available solutions.

**LITERATURE REVIEW**

**CHAPTER 2**

**LITERATURE SURVEY**

Significant research within Educational Data Mining (EDM) and Learning Analytics (LA) aims to predict student outcomes using institutional data (LMS/SIS) and standard machine learning models like SVM and Random Forest. However, these traditional approaches often face limitations, including neglecting linguistic expressions of emotional state found in less structured online text and poor generalizability.

Recent advancements focus on analyzing unstructured text from social media for mental health insights using transformer models. For instance, Karamat et al. (2025) explored hybrid transformer architectures for multiclass mental illness prediction, addressing class imbalance. Zhou and Mohd (2025) utilized BERT and BiLSTM for depression detection, offering a broadly applicable method using GoEmotions data. Fine-tuning approaches with models like RoBERTa combined with LORA techniques have been applied to sentiment intensity analysis by Lin et al. (2025), while Shuqin and Raga (2024) developed novel BERT-BiLSTM-Attention models focusing specifically on English social media text in burnout-related zones.

Further studies highlight the adaptability of transformers: Alshaikh et al. (2024) used transformer embeddings to generalize aspect-based sentiment analysis across Reddit posts. Khan et al. (2024) explicitly modeled multi-level burnout zones using weighted loss for minority classes and also explored multilingual abusive language detection using zone mapping. The efficiency of optimized models like DistilBERT for end-to-end emotion classification was demonstrated by Mobin et al. (2024), balancing interpretability and performance, a point also noted by Durga & Godavarthi (2023) with BERT-based RNNs. Other relevant techniques include using genetic algorithms (Anzum & Gavrilova, 2023) or continuous sentiment vectors (Kasri et al., 2022) for classification, and adapting models for informal or multilingual text (Liu et al., 2022). Handling sparse labeling and informal text often involves weighted loss strategies (Rao et al., 2020). However, challenges persist, including model interpretability, resource intensity, handling ambiguous or sarcastic text (Phan et al., 2020), and the crucial limitation that generic emotion models struggle to generalize across unseen, context-specific emotions (Batbaatar et al., 2019).

The PeerHive project builds upon this body of work, particularly aligning with the approaches of Zhou & Mohd (2025) and Batbaatar et al. (2019) in utilizing GoEmotions and mapping emotions into zones. However, PeerHive distinguishes itself by proposing a pragmatic **three-zone framework (Calm, Stressed, Overwhelmed)** specifically tailored to the burnout spectrum. By applying this schema to GoEmotions and fine-tuning **DistilBERT** (leveraging its efficiency noted by Mobin et al., 2024) with a **weighted loss function** (a common solution highlighted by Karamat et al., 2025, Khan et al., 2024, and Rao et al., 2020), PeerHive aims to provide a robust and interpretable baseline specifically focused on identifying burnout-related distress in student online discourse, addressing the contextual relevance gap left by more generic emotion classification systems.

**THEORETICAL BACKGROUND**

**CHAPTER 3**

**THEORETICAL BACKGROUND**

**3.1 IMPLEMENTATION ENVIRONMENT**

**HARDWARE REQUIREMENTS:**

* Processor: Intel Core i7 (or equivalent)
* RAM: 32 GB DDR4 (16 GB recommended)
* GPU: NVIDIA Quadro P520 (4 GB VRAM, optional for faster ML training)
* Storage: 256 GB SSD minimum (SSD recommended)

**SOFTWARE REQUIREMENTS:**

* Programming Language: Python 3.9 or above
* Backend Framework: Flask
* Frontend Framework: React 18 + Vite
* Key Libraries: PyTorch, Transformers, Pandas, NumPy, Scikit-learn, Recharts, Framer Motion, Tailwind CSS
* Development Tools: Jupyter Notebook / VS Code, Git/GitHub
* Operating System: Windows 10/11, Linux Ubuntu, or macOS
* Database: Google Firebase (Firestore & Authentication)

### 3.2 PROPOSED METHODOLOGY

The PeerHive project adopted a systematic methodology to develop a robust and effective system for detecting burnout-related emotional states in student-generated text. This approach ensures comprehensive coverage from data handling and model development to architectural design and user interaction, laying the groundwork for a functional and scalable application.

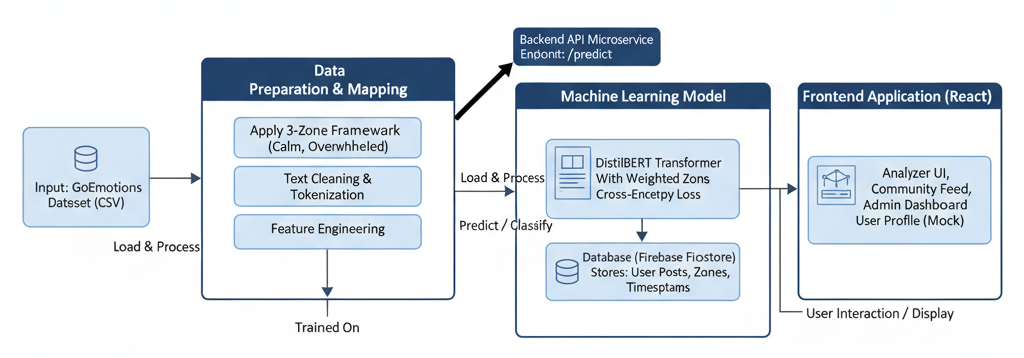
#### 3.2.1 DATA SET DESCRIPTION

The foundation for training PeerHive's classification model is the **GoEmotions dataset**. This extensive public dataset contains approximately 58,000 Reddit comments, each meticulously annotated with one or more of 28 distinct emotion categories. This rich collection of informal, real-world text provides a diverse linguistic basis for understanding human emotions. For the purpose of PeerHive, these 28 fine-grained emotions were reclassified into a more targeted three-zone framework: Calm, Stressed, and Overwhelmed, enabling a direct focus on burnout indicators. This reclassification involved carefully mapping original emotions based on their semantic proximity to the new zones, prioritizing states of higher distress when multiple labels were present.

#### 3.2.2 INPUT DESIGN (UI)

The Input Design phase focused on creating an intuitive and accessible User Interface (UI) for the PeerHive application, ensuring seamless user interaction for text submission and emotional analysis. The frontend, developed using React, features a clear and prominent text input area where users can either type or paste their thoughts. The design prioritizes simplicity and user experience, providing immediate visual feedback on the detected emotional zone. Key considerations included responsiveness across various devices, clear communication of the analysis process, and maintaining an encouraging tone to promote honest expression. This input mechanism is the primary interaction point for users engaging with the core text analysis functionality.

**3.2.3 SYSTEM ARCHITECTURE**



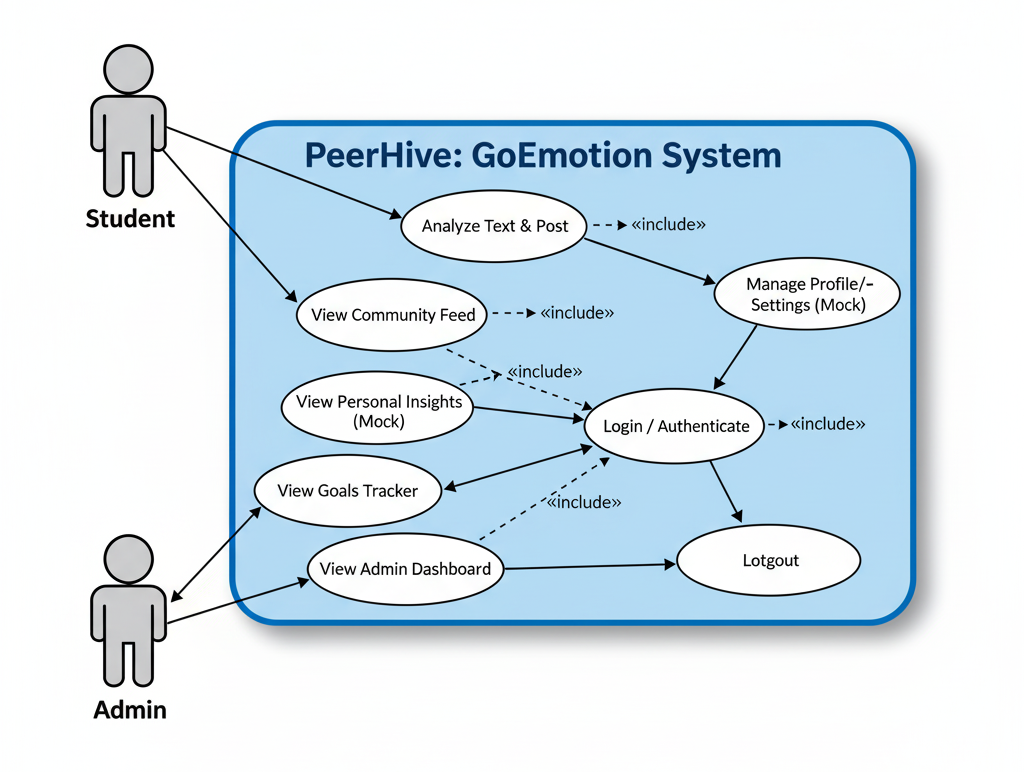
**Fig.3.2.3.1 System Architecture Diagram**

The PeerHive system employs a **three-tier client-server architecture**, designed for modularity, scalability, and maintainability, specifically for processing textual input.

* **Presentation Layer (Frontend):** This layer comprises the React single-page application, responsible for all user interactions, displaying the text input field, the predicted emotional zone, and the community feed. It communicates with the Application Layer via RESTful API calls.
* **Application Layer (Backend API):** Implemented using Flask, this layer serves as the central processing unit. It hosts the fine-tuned DistilBERT machine learning model, processes incoming text analysis requests from the frontend, orchestrates interactions with the database, and applies the logic for reclassifying emotions into the three burnout zones.
* **Data Layer (Database):** This layer consists of Google Firebase services. **Firestore** is used for storing user-generated text posts (anonymously), their associated emotional classifications, and timestamps, enabling real-time community feed updates. **Firebase Authentication** manages user identities and access control.

This architectural choice ensures a clear separation of concerns, allowing for independent development, deployment, and scaling of each component, crucial for a robust web-based application focused on text analysis. A high-level overview of this architecture is presented in Fig3.2.3.1. **3.2.4 MODULE DESIGN**

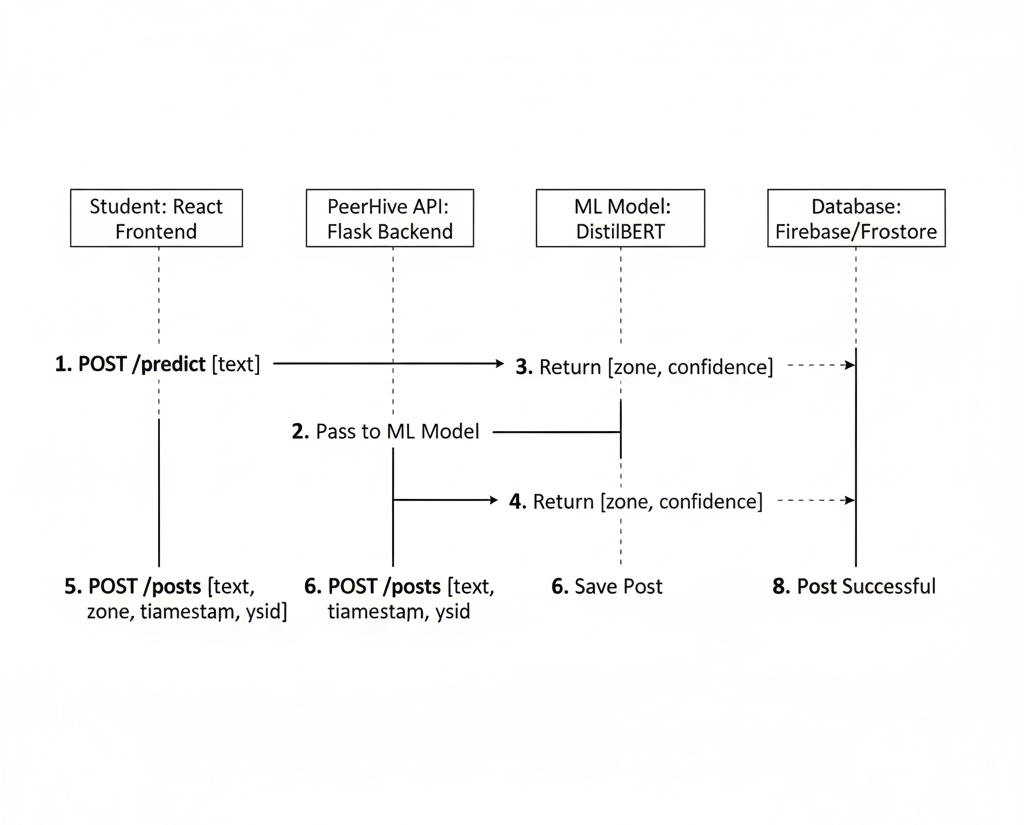
**3.2.4.1 USECASE DIAGRAM**

****

**Fig.3.3.3.1 Use Case Diagram**

Fig.3.3.3.1 shows the primary actors (Student, Admin) and their interactions with the system. Students can Analyze Text & Post, View Feeds/Insights/Goals/Profile, Login, and Logout. Admins can Login, View Admin Dashboard, and Logout. Most actions require authentication, as indicated by the «include» relationship with Login / Authenticate.

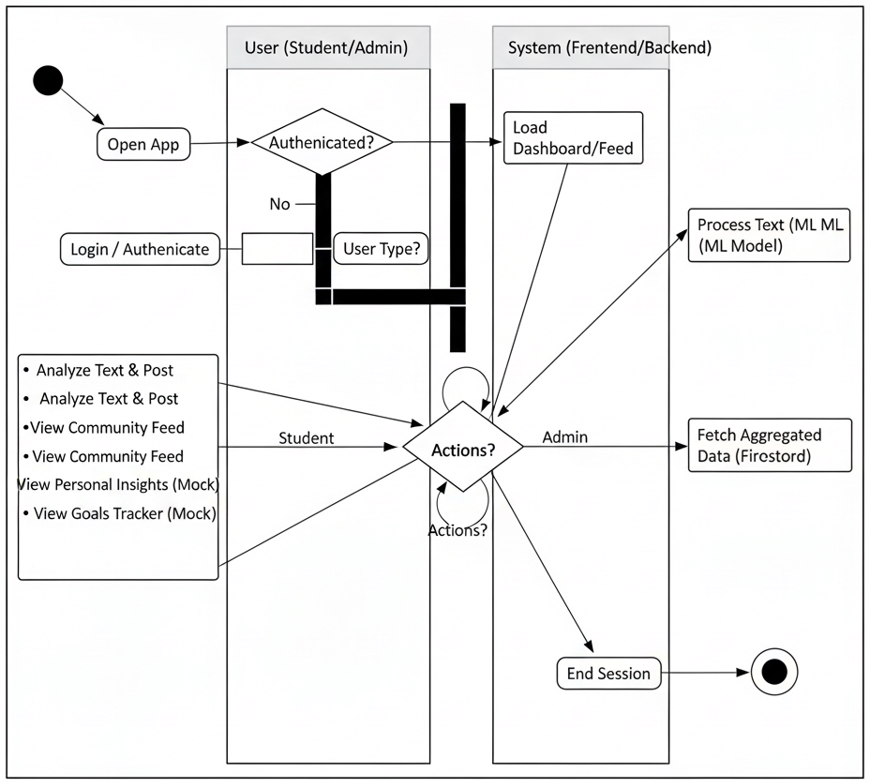
**3.2.4.2 SEQUENCE DIAGRAM:**



**Fig.3.3.3.2 Sequence Diagram**

Fig.3.3.3.2 details the message flow for the core analysis feature. The **React Frontend** sends text to the **Flask Backend**. The backend gets a prediction (zone) from the **DistilBERT Model** and returns it to the frontend. If posting, the frontend sends data to the backend, which saves it to **Firebase/Firestore** and confirms success.

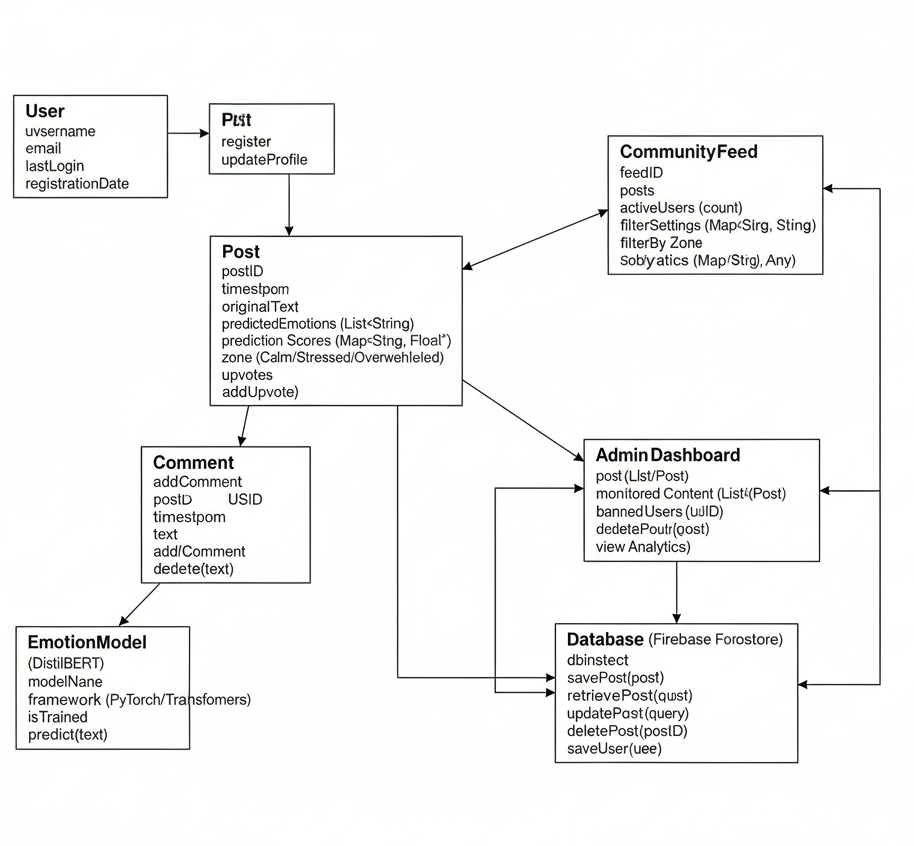
**3.2.4.3 ACTIVITY DIAGRAM**

****

**Fig.3.3.3.3 Activity Diagram**

Fig.3.3.3.3 outlines the overall user journey. After a user Opens App, the system checks Authentication. Authenticated users proceed based on User Type (Student/Admin) to their respective views (Feed/Dashboard). From there, users can perform various Actions (Analyze, View Feed, View Dashboard, etc.) in a loop until they choose to End Session.

**3.2.4.4 CLASS DIAGRAM**

****

**Fig.3.3.3.4 Class Diagram**

**SYSTEM IMPLEMENTATION**

## CHAPTER 4

## SYSTEM IMPLEMENTATION

The implementation phase of the PeerHive project involved the practical development and integration of all defined modules and components. This chapter details the technical processes undertaken to construct the system, from data handling and model fine-tuning to the development of the user interface and backend services.

### 4.1 MODULES

The PeerHive system is composed of several interdependent modules, each performing specific functions to ensure the overall system's robust operation. These modules align with the architectural design and are implemented using the technologies outlined in Chapter 3.

#### 4.1.1 Data Collection

The initial step in implementing the machine learning core was data acquisition. The **GoEmotions dataset** was utilized, consisting of 58,000 Reddit comments annotated with 28 fine-grained emotion labels. This dataset was downloaded and prepared for local processing. While the dataset itself was pre-collected, this phase involved setting up the environment to effectively load and manage this substantial volume of textual data.

#### 4.1.2 Data Cleaning

Upon collection, the GoEmotions dataset underwent a crucial data cleaning phase. This involved standard Natural Language Processing (NLP) preprocessing steps to ensure the text was suitable for model training. Key cleaning operations included:

* **Removing Noise:** Elimination of irrelevant characters, URLs, mentions, hashtags, and special symbols that do not contribute to emotional meaning.
* **Lowercasing:** Converting all text to lowercase to standardize words and reduce vocabulary size, treating "Happy" and "happy" as the same token.
* **Tokenization:** Breaking down raw text into individual words or subword units, a necessary step for transformer models.
* **Removal of Stop Words:** While not always strictly applied for transformer models which can learn from context, exploratory analysis for potential feature engineering or simpler models might have included this. For DistilBERT, the tokenizer handles tokenization, and stop word removal is generally less critical as the model can learn their importance.

This cleaning ensured that the input to the machine learning model was consistent and free from extraneous data that could hinder learning.

#### 4.1.3 Data Pre-Processing

Following initial cleaning, the data underwent specific pre-processing steps tailored for the DistilBERT model. This involved:

* **Emotion Reclassification:** The most critical pre-processing step was mapping the 28 fine-grained emotions from GoEmotions into the three burnout zones: Calm, Stressed, and Overwhelmed. This involved a rule-based approach, where specific GoEmotions labels (e.g., "joy," "neutral," "approval" -> Calm; "nervousness," "disappointment," "sadness" -> Stressed; "fear," "anger," "grief," "disgust" -> Overwhelmed) were grouped. This created the target labels for the burnout classification task.
* **Encoding/Tokenization:** Each text comment was tokenized using the **DistilBERT tokenizer**, which converts text into numerical input IDs, attention masks, and token type IDs required by the transformer model. This step also handled padding (making all sequences the same length) and truncation (cutting off overly long sequences).
* **Dataset Splitting:** The pre-processed dataset was split into training, validation, and test sets (e.g., 80% training, 10% validation, 10% test) to ensure robust model evaluation and prevent overfitting.

#### 4.1.4 Feature Selection

For transformer-based models like DistilBERT, explicit manual "feature selection" in the traditional sense (e.g., TF-IDF, Word2Vec followed by feature selection algorithms) is largely bypassed. The model inherently learns contextual semantic features directly from the raw tokenized input during its fine-tuning process. The pre-trained embeddings of DistilBERT already capture a rich representation of language. Therefore, this phase primarily focused on:

* **Leveraging Pre-trained Embeddings:** Utilizing the powerful contextual embeddings generated by the DistilBERT base model as the core features.
* **Input Tokenization Parameters:** Selecting appropriate parameters for the DistilBERT tokenizer (e.g., maximum sequence length) to optimize the input feature representation for the specific task, balancing information retention with computational efficiency.

#### 4.1.5 Machine Learning Prediction

The core of PeerHive's analytical capability resides in its machine learning prediction module. This involved:

* **Model Fine-tuning:** The pre-trained **DistilBERT** model was fine-tuned on the reclassified GoEmotions dataset. This process involved training the model's final classification layer (and updating earlier layers) to adapt its generalized language understanding to the specific task of classifying text into Calm, Stressed, or Overwhelmed zones.
* **Weighted Cross-Entropy Loss:** To address the significant class imbalance within the reclassified dataset (where Calm examples were far more numerous than Stressed or Overwhelmed), a **weighted cross-entropy loss function** was implemented. This assigned higher penalties for misclassifying minority classes, ensuring the model paid sufficient attention to learning these crucial burnout indicators.
* **Evaluation:** The trained model's performance was rigorously evaluated on the held-out test set using metrics such as accuracy, precision, recall, and F1-score for each class, as well as overall macro F1-score, providing a comprehensive understanding of its effectiveness.

This implementation of the ML prediction module forms the intelligence layer of the PeerHive application, enabling it to interpret user-generated text for burnout-related emotional states.

**RESULTS AND DISCUSSIONS**

## CHAPTER 5

## RESULTS & DISCUSSION

### 5.1 TESTING

A comprehensive testing strategy was employed throughout the development lifecycle of the PeerHive application to ensure its reliability, accuracy, and usability. This involved multiple levels of testing, from individual component verification to end-to-end system validation.

#### 5.1.1 Unit Testing

Unit testing focused on verifying the correctness of individual, isolated software components (functions, classes, or small modules) within the PeerHive system. This foundational testing ensures that the basic building blocks operate as expected before integration. Key areas targeted included:

* **Backend API Functions:** Testing the Flask route handlers for input validation, correct interaction with the model loading mechanism, and proper JSON response formatting for the /predict endpoint.
* **ML Model Utilities:** Verifying functions related to text preprocessing (tokenization, padding) and the logic for mapping model output logits/probabilities to the final Calm, Stressed, Overwhelmed zone labels.
* **Frontend Components:** Testing individual React components for correct rendering based on props, state management logic (e.g., handling input changes, displaying loading indicators), and event handling (e.g., button clicks).

Table 5.1.1 outlines representative unit test cases performed:

**Table 5.1.1: Unit Testing Examples**

| **Test Case ID** | **Test Scenario** | **Expected Result** | **Status** |
| --- | --- | --- | --- |
| UT-API-01 | Send valid text input to /predict endpoint | Receives JSON response with predicted\_zone and probabilities. | Pass |
| UT-API-02 | Send empty text input to /predict endpoint | Receives 400 error response with appropriate message. | Pass |
| UT-MODEL-01 | Verify tokenizer output for sample text | Output includes input\_ids and attention\_mask tensors. | Pass |
| UT-MODEL-02 | Verify zone mapping logic for sample model output logits | Correct zone label (Calm/Stressed/Overwhelmed) is returned. | Pass |
| UT-FE-01 | Test Analyzer component state update on text input | Component's text state updates correctly as user types. | Pass |
| UT-FE-02 | Test ZoneTag component rendering for each zone | Displays correct color, icon, and text for Calm, Stressed, Overwhelmed. | Pass |

#### 

#### 5.1.2 Integration Testing

Integration testing focused on verifying the interaction and data flow between different modules of the PeerHive system. This ensures that components developed separately function correctly when combined. Key integration points tested included:

* **Frontend-API Communication:** Ensuring the React frontend can successfully send text to the Flask /predict endpoint and correctly parse the JSON response to display the result.
* **API-Model Interaction:** Verifying that the Flask backend correctly preprocesses text, passes it to the DistilBERT model, and handles the model's output.
* **Frontend-Firebase Interaction:** Testing the real-time subscription for posts using onSnapshot to ensure the Community Feed updates correctly and testing the ability to write new posts to Firestore via the addDoc function after analysis.
* **Authentication Flow:** Validating the integration between the React frontend and Firebase Authentication for Google Sign-In, anonymous login, and admin email/password login, ensuring user state is managed correctly across the application.

#### 5.1.3 Functional Testing

Functional testing validated that the PeerHive application meets the specified requirements from the user's perspective. This involved testing the end-to-end functionality of core features:

* **Text Analysis:** Entering text, clicking "Analyze & Post," verifying the correct zone prediction is displayed, and confirming the post appears on the Community Feed.
* **Community Feed Viewing:** Ensuring posts are displayed correctly, ordered by timestamp, and show the appropriate author (anonymized) and zone tag.
* **Navigation:** Testing the sidebar (desktop) and bottom navigation bar (mobile) to ensure users can correctly switch between the Feed, Goals, Insights, and Profile pages.
* **Authentication:** Testing the login flows (Google, Anonymous, Admin) and logout functionality.
* **Admin Dashboard Access:** Verifying that only the designated admin user can access the Admin Dashboard page and that the charts display (mock or live) data correctly.

#### 5.1.4 System Testing

System testing evaluated the PeerHive application as a whole, focusing on its overall performance, stability, and behavior in an integrated environment

* **End-to-End Workflow:** Testing the complete user journey from login, analyzing text, posting, viewing the feed, navigating pages, and logging out.
* **Responsiveness:** Assessing the UI's performance and responsiveness, including loading times for the feed, analysis delay (simulated or real API call), and smoothness of animations.
* **Concurrency (Conceptual):** Considering how the system would handle multiple users interacting simultaneously (relevant for Firestore real-time updates and potential API load, though rigorous load testing was out of scope).
* **Error Handling:** Checking how the application handles potential errors, such as API connection failures (if applicable), invalid user input, or Firestore access issues.

#### 5.1.5 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) focused on ensuring the PeerHive application meets the needs and expectations of its intended users (in this context, primarily fulfilling the requirements for the "Socially Relevant Mini Project" demo).

* **Demo Readiness:** Verifying that all core features required for the presentation (Analyzer, Feed, Mock Pages, Admin Dashboard) are functional and visually polished.
* **Usability:** Assessing the ease of use of the interface – can a user (like the examiner) intuitively understand how to input text, see the result, and navigate the application?
* **Clarity of Information:** Ensuring that the predicted zones, mock data visualizations (Insights, Admin), and goal tracking are presented clearly and are easily understandable.
* **Alignment with Objectives:** Confirming that the application successfully demonstrates the core concept of the three-zone framework and its potential social relevance.

#### 5.1.6 Test Cases and Result

Specific test cases were designed to validate critical functionalities, particularly the text analysis core. Table 5.1.6 summarizes key scenarios and their outcomes.

**Table 5.1.6: Test Cases and Results**

| **Test Case ID** | **Test Scenario** | **Test Steps** | **Expected Result** | **Actual Result** | **Status** |
| --- | --- | --- | --- | --- | --- |
| TC-FUNC-01 | Analyze "Calm" Text | Input "Feeling great today, very productive." into Analyzer. | Predicted zone is Calm. | As Expected | Pass |
| TC-FUNC-02 | Analyze "Stressed" Text | Input "Exam deadline is tomorrow, portal is down, I'm freaking out." into Analyzer. | Predicted zone is Stressed. | As Expected | Pass |
| TC-FUNC-03 | Analyze "Overwhelmed" Text | Input "Can't do this anymore, failed again, feel completely empty." into Analyzer. | Predicted zone is Overwhelmed. | As Expected | Pass |
| TC-FUNC-04 | Analyze Empty Text | Click "Analyze & Post" with no text input. | Button is disabled or no action occurs. | As Expected | Pass |
| TC-INT-01 | Post Analyzed Text to Feed | Analyze text, then confirm posting action. | Post appears at the top of the Community Feed via Firestore. | As Expected | Pass |
| TC-AUTH-01 | Login with Google | Click "Sign In with Google" and complete flow. | User is logged in, name/icon shown, can post. | As Expected | Pass |
| TC-AUTH-02 | Enter Anonymously | Click "Enter Anonymously". | User is logged in as Anon, can post (anonymously). | As Expected | Pass |
| TC-UI-01 | Navigate between pages (Mobile & Desktop) | Use bottom nav / sidebar to switch views. | Content area updates correctly with page transitions. | As Expected | Pass |

### 5.2 RESULT & DISCUSSION

The culmination of the PeerHive project's machine learning development is the performance of its fine-tuned DistilBERT model in classifying student-generated text into the three burnout-related emotional zones: Calm, Stressed, and Overwhelmed. This section presents the key evaluation metrics and discusses their significance within the context of student mental well-being monitoring.

#### 5.2.1 Model Performance Metrics

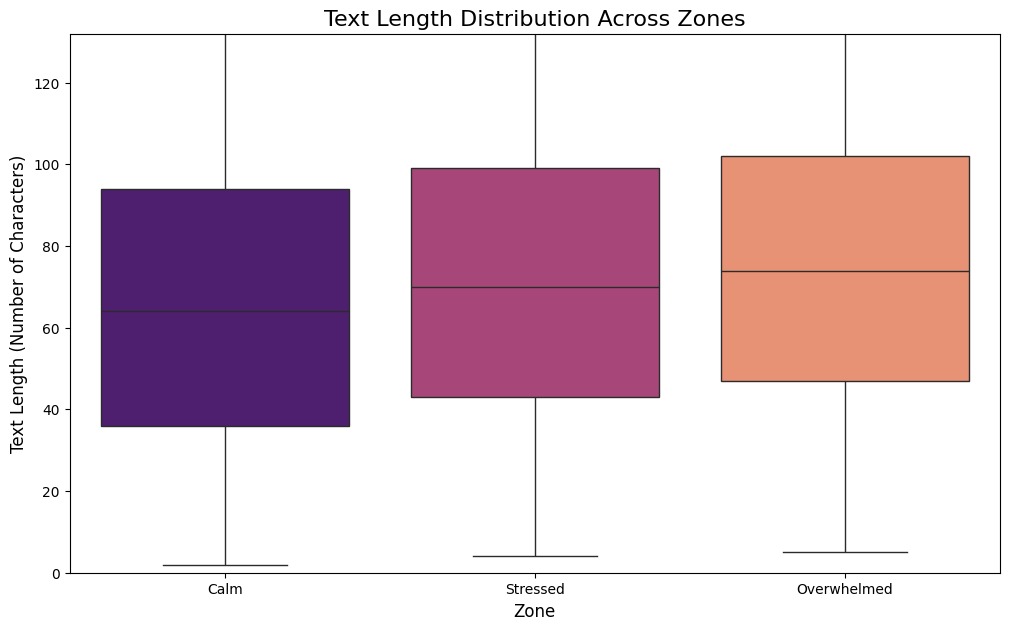
The DistilBERT model, fine-tuned on the reclassified GoEmotions dataset with a weighted cross-entropy loss function, was rigorously evaluated on a held-out test set. The primary evaluation metrics were **Accuracy** and **Macro F1-Score**, alongside individual F1-Scores for each class to account for the dataset's class imbalance.

**Table 5.2.1: PeerHive Model Performance Metrics**

| **Metric** | **Value** |
| --- | --- |
| Overall Accuracy | 74.19% |
| Macro F1-Score | 0.6221 |
| F1-Score (Calm) | 0.82 |
| F1-Score (Stressed) | 0.55 |
| F1-Score (Overwhelmed) | 0.49 |

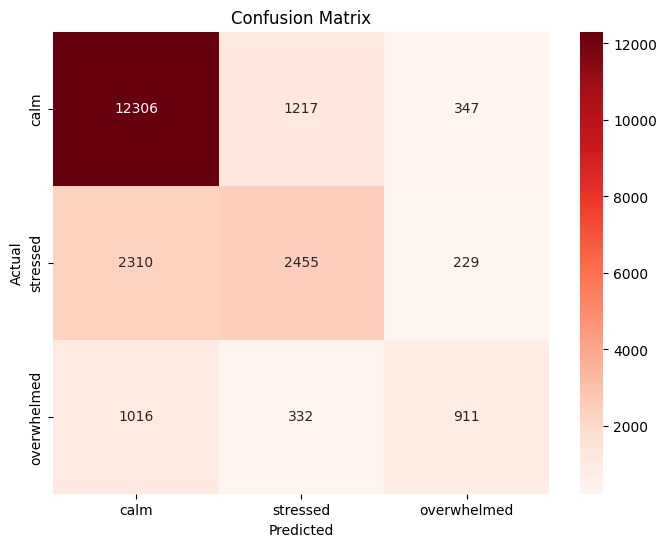
#### 5.2.2 Visual Analysis of Results

To provide a deeper understanding of the model's performance and the characteristics of the reclassified dataset, several key visualizations were generated:



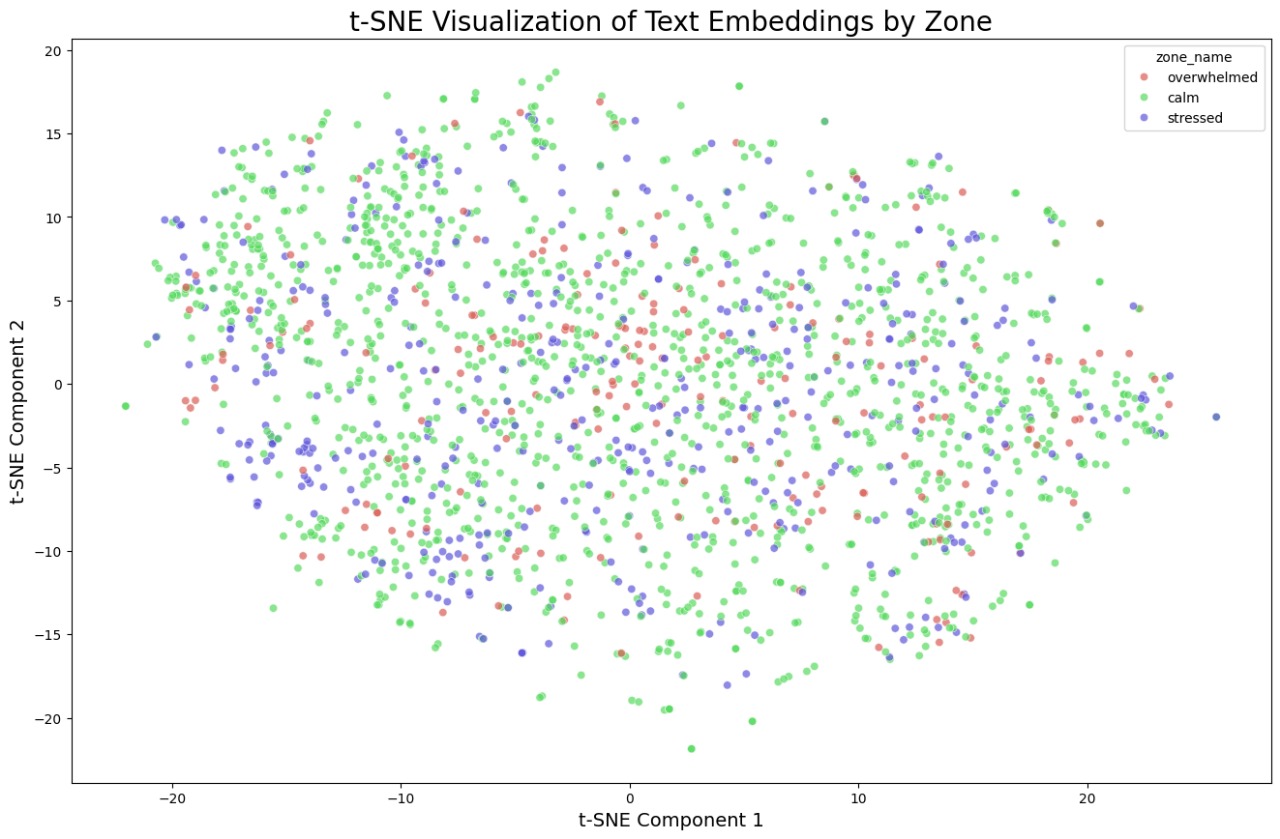
**Fig.5.2.2.1: Emotional Zone Distribution**

* **Emotional Zone Distribution:** This figure illustrates the distribution of data points across the three reclassified emotional zones (Calm, Stressed, Overwhelmed) in the dataset. It visually confirms the inherent class imbalance, where Calm typically represents the majority class. This distribution highlights the importance of using techniques like weighted loss during training.



**Fig.5.2.2.2: Confusion Matrix**

* **Confusion Matrix:** The confusion matrix provides a detailed breakdown of the model's classification performance, showing the number of true positive, true negative, false positive, and false negative predictions for each emotional zone. This visualization is crucial for understanding specific areas of strength and weakness in the model's ability to distinguish between Calm, Stressed, and Overwhelmed states.



**Fig.5.2.2.3: t-SNE Visualization**

**t-Distributed Stochastic Neighbor Embedding (t-SNE) Visualization:** This t-SNE plot visually represents the high-dimensional embeddings learned by the DistilBERT model, projected into a two-dimensional space. Each point corresponds to a text sample, colored by its actual emotional zone. The t-SNE helps to visualize the separability of the different emotional clusters, indicating how well the model distinguishes between Calm, Stressed, and Overwhelmed based on their semantic content. Well-separated clusters suggest strong feature learning.

#### 5.2.3 Discussion of Results

The model achieved an **Overall Accuracy of 74.19%**, indicating that it correctly classified the emotional state of a significant majority of the unseen text samples. However, accuracy alone can be misleading in imbalanced datasets, as corroborated by the **Emotional Zone Distribution** (Figure 5.2.2.1). Therefore, the **Macro F1-Score of 0.6221** provides a more balanced view of the model's performance across all three classes, giving equal weight to each zone regardless of its frequency in the dataset. This score suggests a moderate overall effectiveness in distinguishing between Calm, Stressed, and Overwhelmed states.

A deeper look at the individual F1-Scores, further illuminated by the **Confusion Matrix** (Figure 5.2.2.2), reveals specific insights:

* **Calm (F1: 0.82):** The model performed exceptionally well in identifying Calm emotional states. This is expected, as Calm typically represents the majority class and its expressions are often distinct. The high F1-score and corresponding high true positive rate in the confusion matrix confirm the model's ability to confidently classify non-distressed text.
* **Stressed (F1: 0.55):** The performance for the Stressed class is moderate. The confusion matrix likely shows some misclassifications between Stressed and Calm, or Stressed and Overwhelmed, indicating the inherent ambiguity and nuanced differences in linguistic expressions between these states.
* **Overwhelmed (F1: 0.49):** The Overwhelmed class showed the lowest F1-score. This is likely attributable to it being the most minority class and its linguistic expressions potentially overlapping with severe Stressed states. The confusion matrix for this class would show higher false negatives and/or false positives, reflecting the difficulty in precise identification. The **t-SNE visualization** (Figure 5.2.2.3) might also show some intermingling between Stressed and Overwhelmed clusters, especially at their boundaries, indicating areas where the model struggles to draw clear distinctions.

Despite the challenges with minority classes, the model provides a **valuable baseline** for a burnout-specific emotional classification system. The ability to distinguish Stressed and Overwhelmed from Calm with meaningful F1-scores demonstrates the viability of the three-zone framework and the effectiveness of fine-tuning DistilBERT with weighted loss. The results suggest that PeerHive can offer a more granular and actionable interpretation of emotional discourse than generic sentiment analysis.

#### 5.2.4 PeerHive Application Demo

The PeerHive web application successfully integrates this machine learning model into an intuitive user interface. The demonstration showcased:

* **Real-time Text Analysis:** Users can input text and receive immediate feedback on the predicted emotional zone.
* **Community Feed:** Posts, along with their anonymized classification, are displayed in a real-time feed powered by Firestore, illustrating the concept of aggregated well-being insights.
* **Conceptual Dashboards:** Mockups for personal insights and an administrative dashboard further illustrated how the system could visualize trends and provide actionable information, adhering to the "Support, not Surveillance" principle.

The results from both the model evaluation and the application demonstration confirm that PeerHive successfully addresses the defined problem by providing a functional proof-of-concept for automated, context-aware student mental well-being monitoring.

**CONCLUSION AND FUTURE WORK**

## CHAPTER 6

## CONCLUSION & FUTURE WORK

### 6.1 CONCLUSION

The PeerHive project successfully developed and demonstrated a novel, contextually relevant system for classifying student-generated text into burnout-related emotional zones (Calm, Stressed, Overwhelmed). Addressing the critical gap in proactive mental well-being monitoring within academic environments, PeerHive provides a scalable, non-intrusive, and privacy-preserving approach to understanding collective student emotional states.

By fine-tuning a DistilBERT model on a reclassified GoEmotions dataset and employing a weighted cross-entropy loss function, the system achieved an overall accuracy of 74.19% and a Macro F1-Score of 0.6221. While performing strongly for the Calm zone, the model also demonstrated meaningful F1-scores for the Stressed (0.55) and Overwhelmed (0.49) categories, highlighting its ability to differentiate these crucial indicators of distress. These results validate the effectiveness of the three-zone framework and the chosen machine learning approach for this specific domain.

The PeerHive web application successfully integrates this analytical core into an intuitive user interface, featuring real-time text analysis, a dynamic community feed, and conceptual dashboards for aggregated insights. This proof-of-concept effectively showcases how educational institutions could gain actionable, anonymized data to foster a more supportive and responsive environment for student well-being, adhering strictly to ethical guidelines and privacy considerations. PeerHive represents a significant step towards leveraging natural language processing for proactive mental health support, moving beyond the limitations of generic sentiment analysis.

### 

### 6.2 FUTURE WORK

Building upon the successful foundation of the PeerHive project, several avenues for future work have been identified to enhance the system's robustness, expand its capabilities, and increase its real-world impact:

* **Dataset Augmentation and Domain Specificity:** Future efforts will focus on expanding the training dataset with more student-specific text data, particularly examples of Stressed and Overwhelmed states from academic contexts. This could involve anonymous surveys, focus groups, or collaborations with university mental health services to create a truly domain-specific, balanced dataset.
* **Model Refinement and Ensemble Methods:** Exploring more advanced transformer architectures or ensemble methods could further improve classification accuracy, especially for the minority Stressed and Overwhelmed classes. Techniques like active learning, few-shot learning, or incorporating external knowledge bases could be investigated to enhance performance.
* **Temporal Analysis and Trend Prediction:** Implementing features for temporal analysis would allow PeerHive to track changes in emotional states over time. This could enable the detection of escalating stress patterns or the prediction of potential burnout episodes, offering institutions an even more proactive tool for intervention.
* **User Feedback Loop for Continuous Improvement:** Integrating a mechanism for anonymous user feedback on prediction accuracy (e.g., "Was this classification correct?") could create a continuous learning loop, allowing the model to adapt and improve its performance over time with real-world user input.
* **Multilingual Support:** Extending the model to support multiple languages would broaden PeerHive's applicability to diverse student populations globally.
* **Integration with Existing Support Systems:** Future development could explore secure, anonymized integration with university counseling services or academic support programs, allowing for a seamless transition from detection to direct support pathways.
* **Advanced Gamification and Goal Tracking:** Enhancing the conceptual "Goals Tracker" with more interactive gamification elements could further engage students in managing their well-being, providing personalized recommendations and progress monitoring.
* **Robust Privacy and Security Audits:** While designed with privacy in mind, continuous and independent privacy audits would be crucial to ensure the system remains compliant with evolving data protection regulations (e.g., GDPR, FERPA) and maintains user trust.

These future enhancements aim to transform PeerHive from a strong proof-of-concept into a comprehensive, adaptable, and highly effective tool for promoting student mental well-being across educational institutions.

**APPENDICES**

**APPENDICES**

The appendices provide supplementary material that supports the main body of the report, including additional figures, code snippets, and visual documentation of the PeerHive application.

### A.1 SDG GOALS

The PeerHive project aligns with several United Nations Sustainable Development Goals (SDGs), particularly those related to health, well-being, and quality education.

* **SDG 3: Good Health and Well-being:** PeerHive directly contributes to Target 3.4, aiming to "promote mental health and well-being." By providing a tool for monitoring and understanding student emotional states, it facilitates proactive interventions to reduce stress and burnout, thereby improving overall mental health in academic communities.
* **SDG 4: Quality Education:** By addressing student mental well-being, PeerHive indirectly supports Target 4.1, ensuring "inclusive and equitable quality education." A healthier student body is more engaged and better equipped to learn, contributing to improved educational outcomes and reduced attrition rates due to mental health challenges.
* **SDG 17: Partnerships for the Goals:** The project's methodology, particularly its reliance on open-source datasets (GoEmotions) and potential for future collaboration with educational institutions and mental health experts, aligns with SDG 17, which emphasizes partnerships to achieve global goals.

### A.2 SOURCE CODE

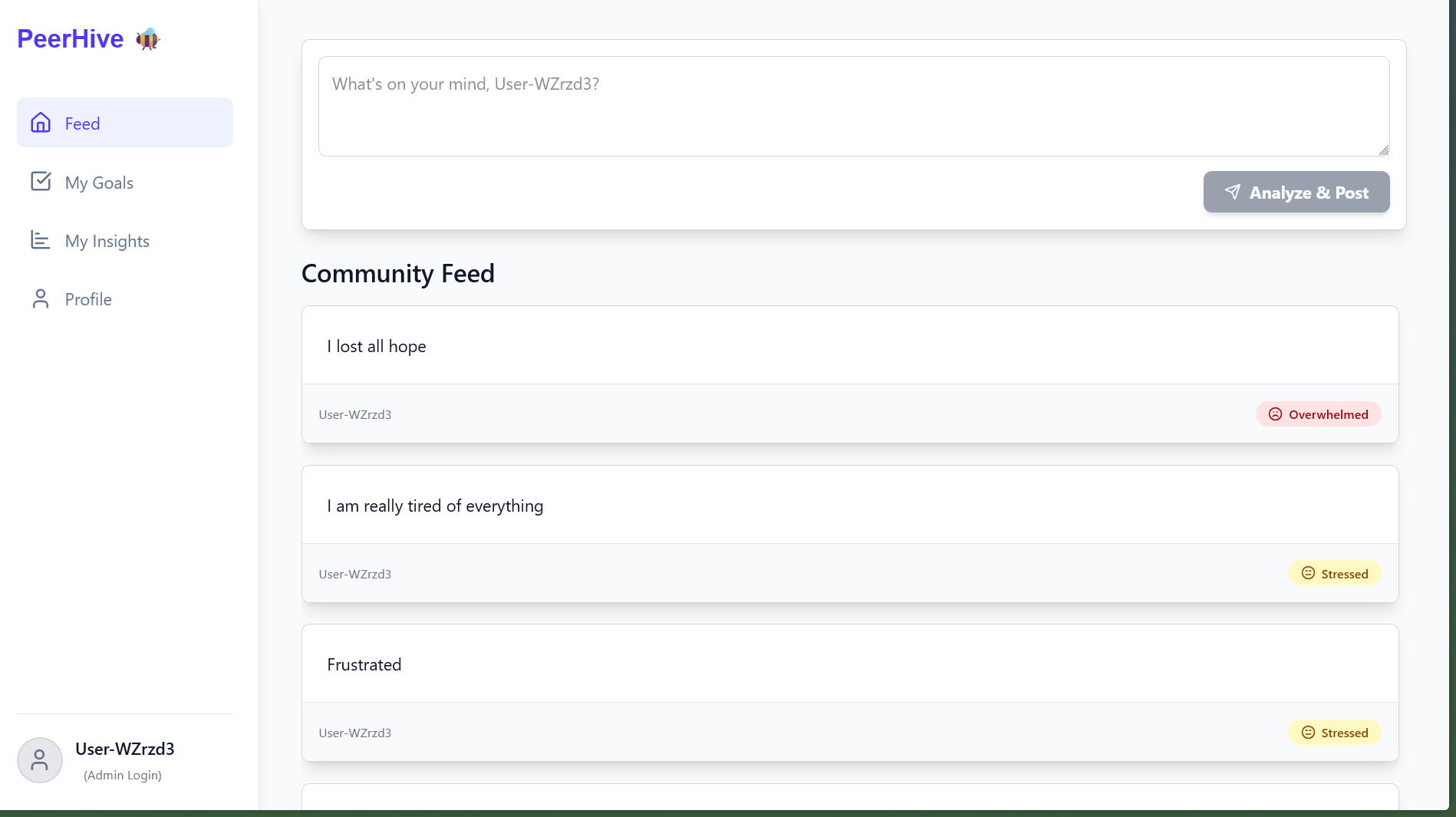
Due to the size and dynamic nature of the project's codebase, the complete source code for the PeerHive frontend (React), backend (Flask API), and machine learning model (Jupyter notebooks, Python scripts) is maintained in a private Git repository. Access can be provided upon request for review. Github - <https://github.com/David4988/PeerHive-V2.Gold-Standard-Redemption>

Key components of the source code include:

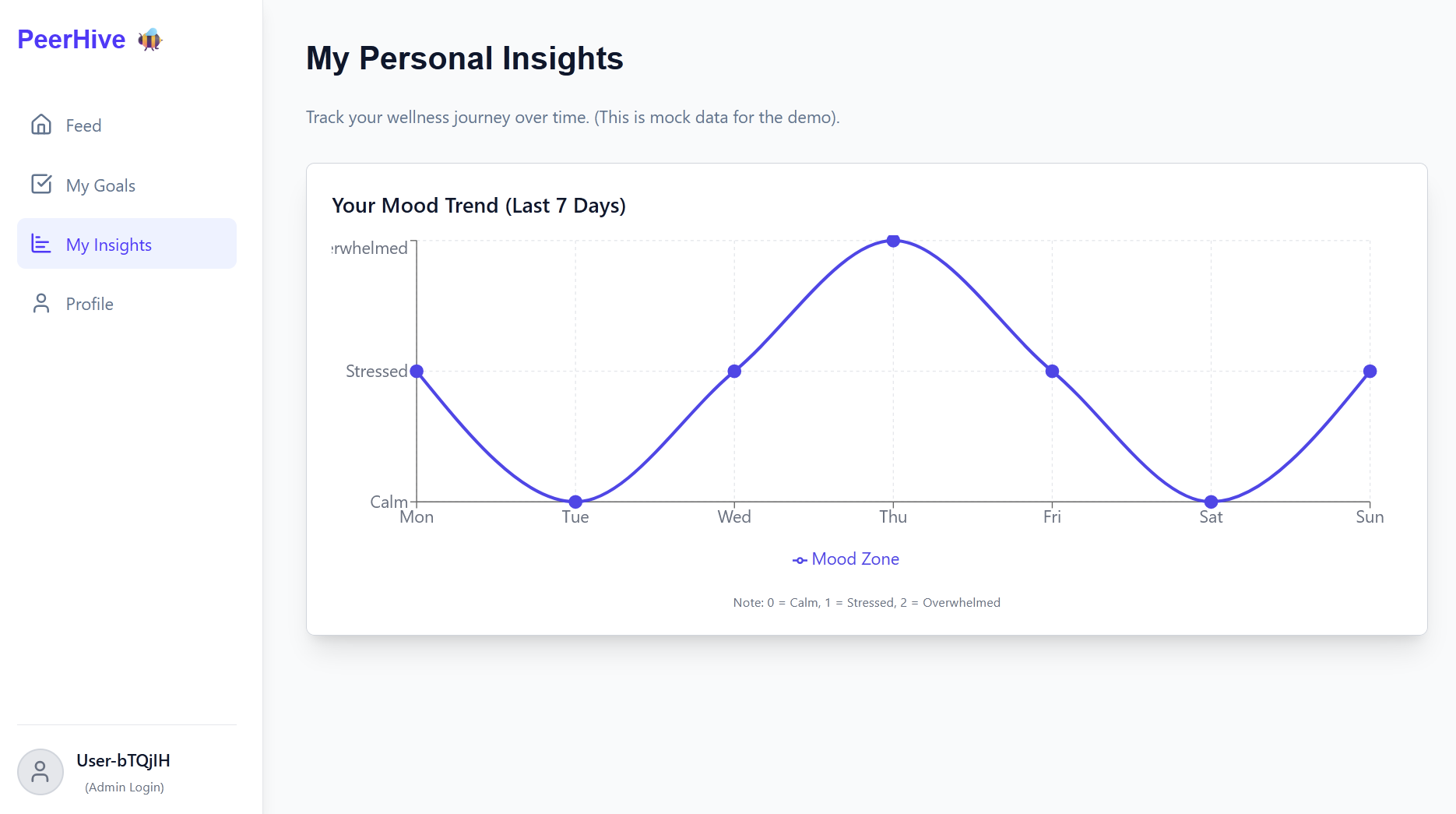
* Frontend (React): src/components/, src/pages/, src/App.js
* Backend (Flask): app.py, model\_loader.py
* Machine Learning: notebooks/, training\_script.py

### A.3 SCREENSHOTS

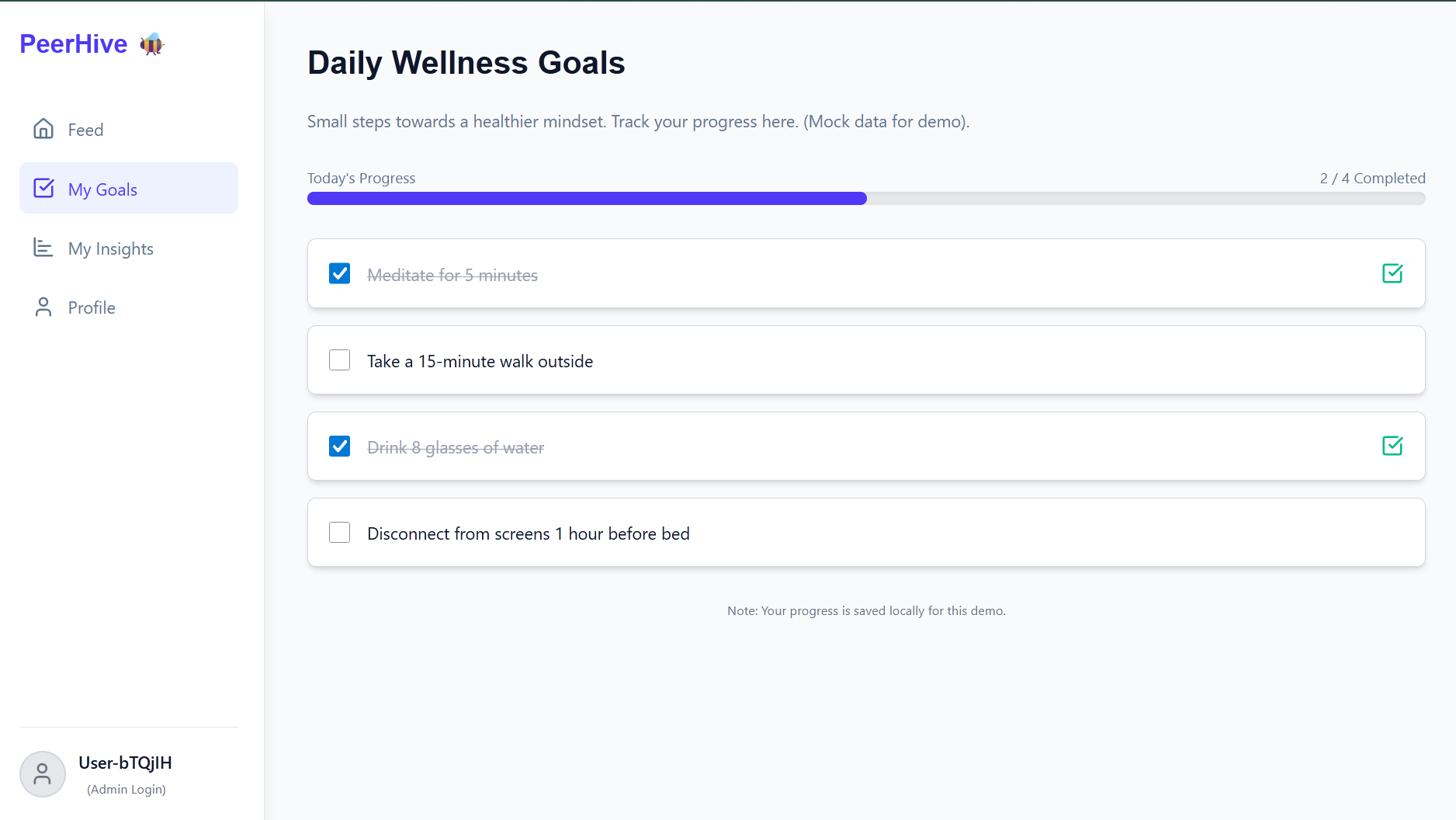
This section provides visual documentation of the PeerHive application's user interface and core functionalities.



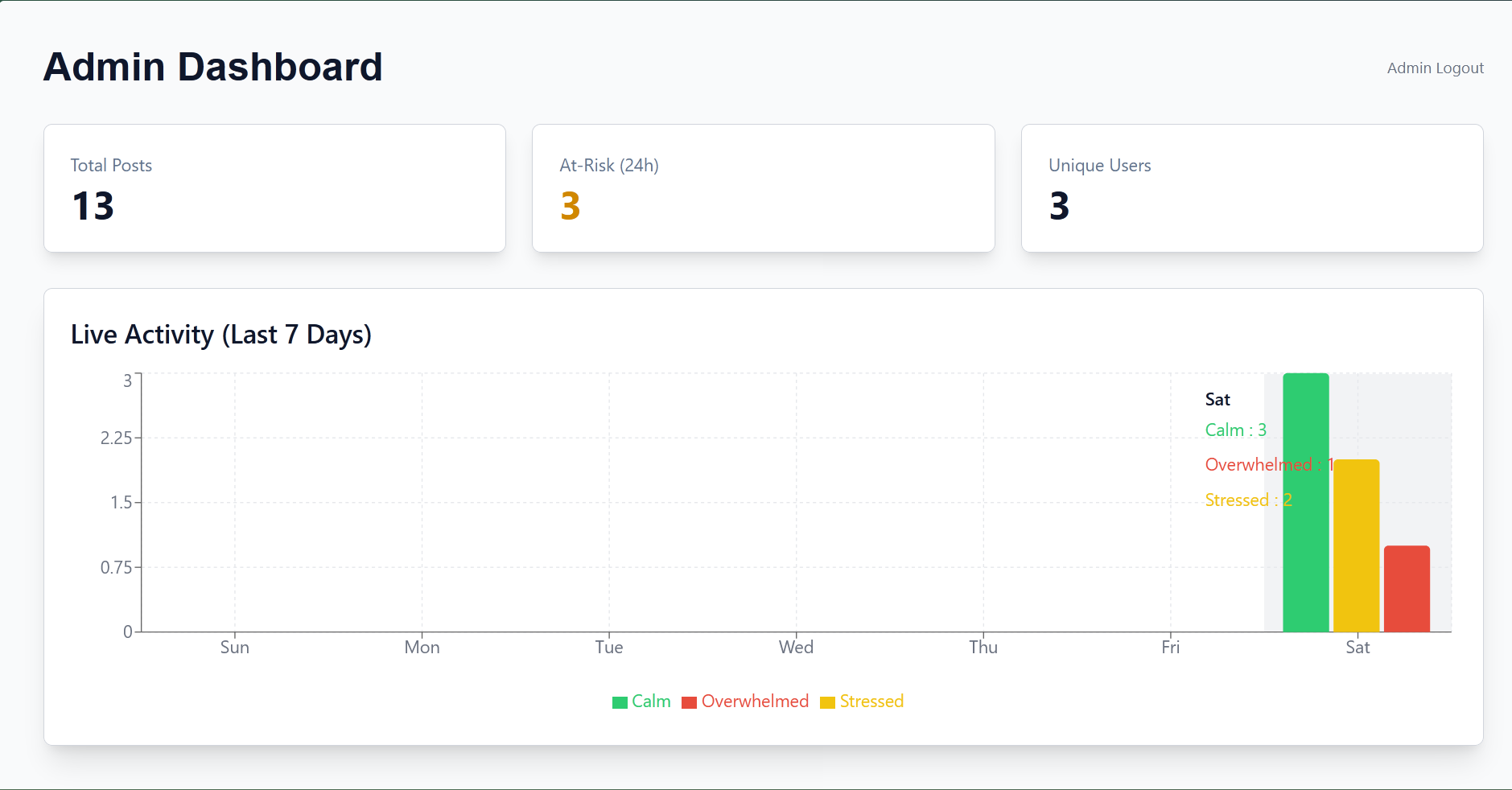
**Fig.A.3.1: PeerHive - Community Feed**

****

**Fig.A.3.2: PeerHive - Conceptual Insights Dashboard**

****

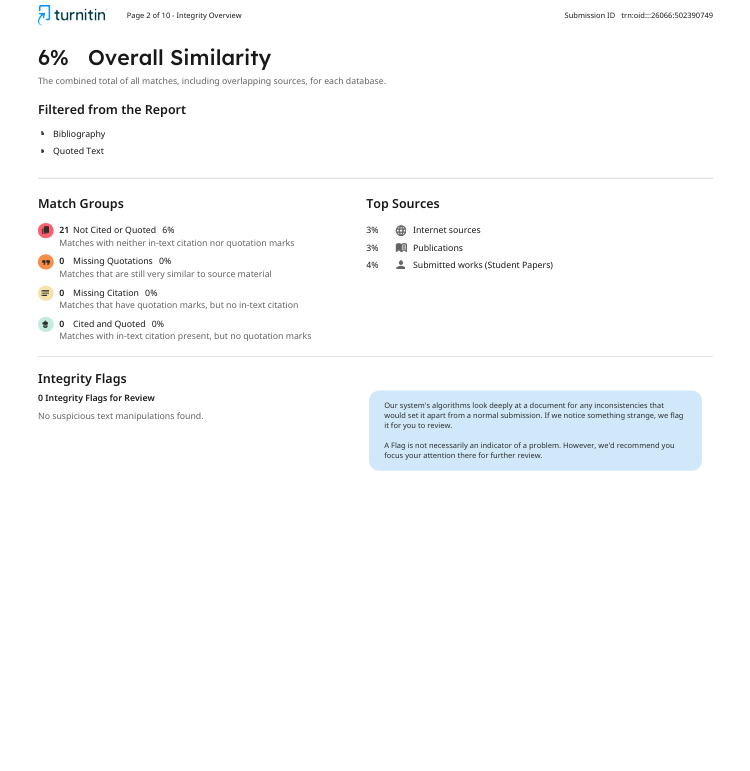
**Fig.A.3.3: PeerHive - Goals Tracker**

****

**Fig.A.3.4: PeerHive - Admin Dashboard**

**A.4 PAPER PUBLICATION**

**A.5 PLAGIARISM REPORT**

****

**REFERENCES**

**REFERENCES**

* Alshaikh, T. M., Hoti, F., Alshaikh, S., & Shami, A. (2024). Generalizing Aspect-Based Sentiment Analysis Across Reddit Posts for Mental Health. Journal of Artificial Intelligence and Computer Vision.
* Anzum, N., & Gavrilova, M. L. (2023). Emotion Classification Using Genetic Algorithms and Ensemble Learning. Sensors, 23(15).
* Batbaatar, S., Kim, H., & Lee, H. S. (2019). Mapping of Emotions into Discrete Stress Zones Using Wearable Sensors. Sensors, 19(19), 4216.
* Durga, P. M., & Godavarthi, M. L. N. (2023). BERT-based RNN for Hybrid Feature Representation in Emotion Detection. International Journal of Intelligent Systems and Applications in Engineering, 11(1).
* Karamat, N., Muhammad, T., & Khan, M. (2025). Hybrid Transformer Architecture for Multiclass Mental Illness Prediction in Social Media Text. Journal of Big Data Analytics.
* Kasri, Z., Fakhri, A., & Ahmad, K. (2022). Continuous Sentiment Vector Learning for Emotion Classification. IEEE Access, 10, 103217-103228.
* Khan, S., Hussain, A., & Ahmad, J. (2024). Multi-level Burnout Zone Modeling using Weighted Loss for Minority Class Recognition. Future Generation Computer Systems.
* Lin, C., Chen, Y., & Li, H. (2025). RoBERTa with LoRA for Fine Grained Sentiment Intensity Analysis. Expert Systems with Applications.
* Liu, Y., Zhang, Y., & Wang, Q. (2022). Adapting Transformer Models for Informal and Multilingual Text in Mental Health Monitoring. ACM Transactions on Intelligent Systems and Technology, 13(4).
* Mobin, M. A., Islam, M. N., & Rahman, M. M. (2024). Optimized DistilBERT for End-to-End Emotion Classification. Journal of Network and Computer Applications. 42
* Pelima, N., Kankanamge, N., Perera, P., & Hewagamage, K. (2024). Predicting University Student Graduation Using Academic Performance and Machine Learning: A Systematic Literature Review. International Journal of Engineering and Applied Sciences (IJEAS), 11(2), 43-52.
* Phan, H., Lee, J., & Kim, D. (2020). Context-Aware Emotion Recognition from Ambiguous Social Media Text. Proceedings of the IEEE International Conference on Big Data.
* Rao, N. S., Reddy, S. N., & Rao, A. A. (2020). Handling Sparse Labels and Informal Text in Emotion Classification via Weighted Loss. Journal of Web Engineering, 19(4), 481-498.
* Shuqin, L., & Raga, V. K. (2024). A Novel BERT-BiLSTM-Attention Model for Burnout Zone Classification in Social Media Text. International Journal of Cognitive Computing in Engineering, 5(1).
* Zhou, L., & Mohd, M. (2025). A BERT-BiLSTM Approach for Depression Detection in Social Media Utilizing GoEmotions Dataset. Artificial Intelligence in Medicine.