Analysis of User Entries for Periodic Mood Tracking Using Web based Application

Dr. Bindu Ashwini C   
Lecturer in Psychology, *Department of Placement and Training*R.V College of Engineering

Kushal J   
1RV21CS072 *Computer Science Department*R.V College of Engineering

Niranjan S   
1RV21CS103 *Computer Science Department*R.V College of Engineering

Siddharth E  
1RV21CS159 *Computer Science Department*R.V College of Engineering

Samartha S Rao  
1RV21CS140 *Computer Science Department*R.V College of Engineering

***Abstract*—This paper presents a web-based system for psychological monitoring, leveraging sentiment analysis to evaluate the daily activities of users. The system aims to assist psychologists by providing a platform for patients to log their daily experiences. The collected data is analyzed to determine sentiment trends, aiding in psychological assessments. The frontend is built using React, with Firebase for authentication, and Flask on the backend. Sentiment analysis is performed using the Gemini API, with results stored in MongoDB. The system provides a comprehensive interface for both patients and counselors, facilitating mood tracking and analysis through visualizations created with Matplotlib. Preliminary results indicate that the system effectively captures and visualizes mood trends, offering valuable insights for psychological evaluation and intervention.**

***Keywords—Sentiment Analysis, Psychological Monitoring, React, Flask, Gemini API, Firebase, MongoDB, Matplotlib***

# Introduction

# Mental health disorders are a significant concern worldwide, affecting millions of individuals. The World Health Organization estimates that one in four people will be affected by mental or neurological disorders at some point in their lives. Traditional methods of psychological assessment, such as in-person consultations and paper diaries, often fail to provide continuous monitoring, leading to gaps in understanding patient moods and behaviors. Recent advancements in technology, particularly in natural language processing (NLP) and machine learning, offer new opportunities for real-time mood tracking and analysis.

## Problem Statement

# Despite the availability of numerous mental health apps, there is a lack of comprehensive systems that integrate sentiment analysis with user-friendly interfaces and secure data handling. Most existing solutions do not provide detailed sentiment analysis or fail to present the data in an easily interpretable format for both patients and psychologists. This gap in the market highlights the need for a system that can not only analyze the emotional content of user entries but also provide meaningful visualizations and insights.

## Objectives

The primary objective of this project is to develop a web-based platform that leverages sentiment analysis to monitor the psychological state of users. The specific objectives include:

1. Designing a user-friendly frontend using React to facilitate easy logging of daily activities and emotions.
2. Implementing secure user authentication using Firebase to ensure data privacy.
3. Utilizing Flask and the Gemini API to perform sentiment analysis on user entries.
4. Storing and managing user data and sentiment scores in MongoDB.
5. Creating visualizations with Matplotlib to help users and psychologists identify mood trends and patterns over time.

## Significance

The significance of this project lies in its potential to transform the way mental health is monitored and managed. By providing a platform that enables continuous mood tracking, the system offers valuable insights that can aid in early detection of mental health issues and improve the effectiveness of psychological interventions. The integration of sentiment analysis adds an objective layer to traditional self-reporting methods, enhancing the accuracy and reliability of mood assessments. This tool can serve as a supplementary resource for psychologists, providing them with additional data to support their clinical decisions.

# System Architecture

The proposed system comprises several components, each integral to its functionality. Each subheading delves into the specific components and their interactions, offering a clear understanding of the entire architecture.

## Frontend Architecture

The frontend architecture is designed to provide a seamless user experience. It includes the following components:

1. **User Interface Design**

* **Activity Logging:** The interface provides a form for users to log their daily activities. The form includes fields for activity description and title.
* **Sentiment Visualization:** Sentiment scores are color-coded and displayed on cards representing each day's entry. Users can quickly identify positive, negative, and neutral sentiments.

2. **Navigation and State Management**

* **Navigation Bar:** A persistent navigation bar allows users to switch between different sections of the application, including new entries, past entries, and mood analysis.
* **State Management:** The application uses Redux for state management, ensuring that the user interface remains responsive and consistent.

## Authentication Architecture

Firebase authentication ensures secure access and user management. It includes:

1.**User Registration and Login**

* **Registration:** New users can register by providing their email and password. Firebase handles the creation and storage of user credentials.
* **Login:** Existing users can log in using their credentials. Firebase verifies the credentials and grants access to the application.

2.**Session Management**

* **Token-Based Authentication:** Upon successful login, Firebase generates a session token. The token is used to authenticate subsequent requests to the backend.
* **Secure Communication:** All authentication-related communications are encrypted using HTTPS, ensuring data security.

## Backend Architecture

The backend, built with Flask, handles data processing and communication with external APIs. It includes:

1.**Data Extraction and API Integration**

* **Data Extraction:** The backend extracts user entries from the frontend and prepares them for sentiment analysis.
* **Gemini API Integration:** The extracted data is sent to the Gemini API for sentiment analysis. The API returns sentiment scores, which are then processed by the backend.

2. **Data Storage and Retrieval**

* **MongoDB Integration:** The backend stores user entries and sentiment scores in MongoDB. The database schema is optimized for quick retrieval and efficient storage.
* **Data Retrieval:** The backend retrieves stored data to display past entries and generate visualizations.

## Database Architecture

MongoDB is used as the primary data store. It includes:

1.**Schema Design**

* **User Collection:** Stores user information, including authentication tokens and profile data.
* **Entries Collection:** Stores daily activity entries along with their sentiment scores. Each document includes fields for the activity description, title, sentiment score, and date.

2. **Data Access Patterns**

* **Indexing:** Indexes are created on key fields to improve query performance. This includes indexes on user IDs and dates.
* **Aggregation:** Aggregation pipelines are used to calculate average sentiment scores and generate data for visualizations.

# Methodology

This section describes the methodologies employed in the development and operation of the psychological monitoring system. It covers user data collection, sentiment analysis, data storage and processing, and visualization techniques.

## User Data Collection

1. **Daily Activity Logging**

Patients log into their accounts using Firebase authentication. Once logged in, they are prompted to enter their daily activities through a form provided on the home screen. Each activity log includes:

* **Activity Description:** A detailed account of the activity performed, allowing users to describe their emotions and thoughts.
* **Title:** A brief title for the activity entry, used to identify the entry on the card displayed on the home screen.

2. **Data Structuring**

Each entry is structured in a consistent format to facilitate processing and analysis. The data includes timestamps to track the date and time of each entry, ensuring chronological ordering and easy retrieval for analysis.

## Sentiment Analysis

1.**Text Processing**

The text of each activity description is preprocessed before sentiment analysis. Preprocessing steps include:

* **Tokenization:** Breaking down the text into individual words or tokens.
* **Stop-word Removal:** Removing common words (e.g., "and", "the") that do not contribute to sentiment.
* **Normalization:** Converting text to lowercase and removing punctuation to standardize the input.

2. **Sentiment Scoring**

The preprocessed text is sent to the Gemini API, which returns a sentiment score between -1 (negative) and 1 (positive). The API uses natural language processing techniques to evaluate the sentiment based on context and word associations. The sentiment scores are then averaged for each day's entries to determine the overall mood for that day..

## Data Storage and Processing

1.**Database Schema**

MongoDB is used to store user entries and sentiment scores. The database schema includes collections for:

* **User Profiles:** Contains user information and authentication details.
* **Activity Entries:** Stores each activity log with its corresponding sentiment score and color-coding.
* **Sentiment Scores:** Stores the sentiment analysis results, including the average score for each day.

2.**Data Retrieval and Update**

Flask handles the retrieval and updating of data. When a user logs a new activity, Flask processes the entry, calls the Gemini API for sentiment analysis, and updates the MongoDB database with the new information. Flask also retrieves data for visualization and analysis when users or counselors access the system.

## Visualization

1.**Graph Generation**

Matplotlib is used to create visualizations that display mood trends over time. Types of graphs include:

* **Line Charts:** Show daily sentiment scores, highlighting trends and fluctuations.
* **Bar Charts:** Summarize monthly sentiment averages, providing a broader view of mood trends.
* **Scatter Plots:** Identify significant mood shifts, helping to pinpoint specific events or activities that influenced the user's mood.

2.**User and Counselor Interfaces**

The visualization components are integrated into the user and counselor interfaces. Users can access their mood trends through the analysis section of the website, while counselors can view analytics for their patients by selecting the respective patient profile. The visualizations provide a clear and intuitive way to interpret the data, aiding in psychological assessments and interventions.

## Security and Privacy

**1.Firebase Authentication**

Firebase is used for secure user authentication, ensuring that only authorized users can access their data. It provides a robust framework for managing user identities and access control, supporting various authentication methods such as email/password, social media logins, and phone authentication.

**2.HTTPS Communication**

All data transmitted between the frontend and backend is encrypted using HTTPS. This ensures that user data is protected from eavesdropping and tampering during transmission. HTTPS provides a secure communication channel, safeguarding sensitive information such as login credentials and personal activity logs.

# Implementation

The implementation of the sentiment analysis-based psychological monitoring system involves multiple technical components, each contributing to the overall functionality and user experience. This section details the technical architecture and implementation of the frontend, backend, data processing, and visualization components.

## Frontend Implementation

The frontend of the system is designed using React, a popular JavaScript library for building user interfaces. The primary focus is on providing an intuitive and responsive user experience.

1.**React Components:**

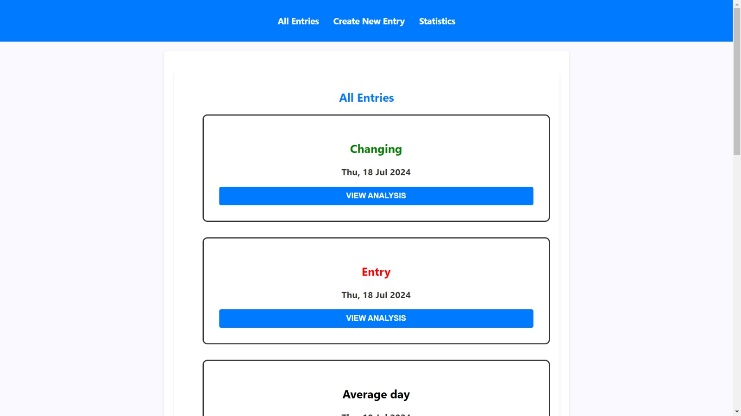
* **Login and Registration:** Utilizes Firebase Authentication for secure user login and registration. The components handle user input, validation, and communication with Firebase.
* **Dashboard:** The main user interface, displaying navigation options (new entry, past entries, analysis), and summarizing user activity.
* **New Entry Form:** A form component that allows users to log their daily activities. It includes fields for activity description, mood, and any additional comments.
* **Past Entries Display:** A component that fetches and displays past entries in a card format, each card color-coded based on sentiment scores.
* **Analysis View:** Renders graphs and charts using libraries like Chart.js and D3.js to visualize mood trends over time.

2. **State Management:**

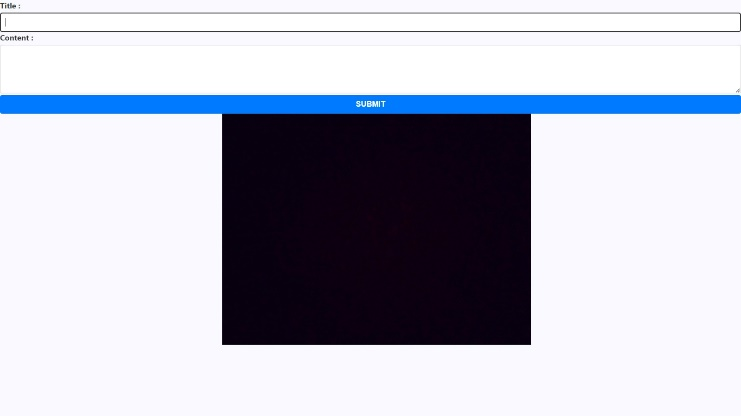
* **Redux:** Manages the application state, ensuring a seamless flow of data between components. It handles user authentication state, entry submissions, and retrieval of past entries.

3. **Styling and Responsiveness:**

* **CSS-in-JS (Styled Components):** Used for component-level styling, ensuring a consistent look and feel. Responsive design principles ensure the application works across various devices.



***6.1.1 Home page consisting of all the entries by the user***



## **6.1.2 WebPage used for the collection of user entries**

## Backend Implementation

The backend is implemented using Flask, a lightweight WSGI web application framework in Python. It handles API requests from the frontend, processes data, and communicates with external services like the Gemini API and MongoDB.

1.**API Endpoints:**

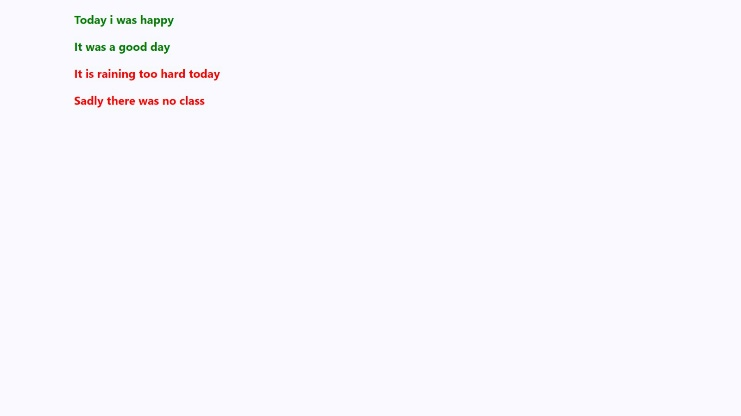
* **User Authentication:** Routes for login, registration, and password management. These endpoints interact with Firebase Authentication to manage user credentials securely.
* **Entry Submission:** POST endpoint to receive new entries from the frontend. It validates the data, calls the Gemini API for sentiment analysis, and stores the result in MongoDB.
* **Entry Retrieval:** GET endpoint to fetch past entries for the logged-in user. It queries MongoDB and returns the data in a format suitable for frontend rendering.
* **Sentiment Analysis:** A dedicated route that processes text input using the Gemini API. This route handles text preprocessing, API communication, and result interpretation.

2. **Sentiment Analysis Integration:**

* **Text Preprocessing:** Tokenization, stop-word removal, and normalization to prepare the text for analysis.
* **Gemini API Call:** Sends preprocessed text to the Gemini API and receives sentiment scores. The API response includes a score between -1 (negative) and 1 (positive).
* **Result Handling:** Interprets the sentiment score and prepares data for storage and visualization. The scores are color-coded (green for positive, red for negative, black for neutral) and associated with the user entries.

3.**Data Storage:**

* **MongoDB Integration:** The system uses MongoDB, a NoSQL database, to store user entries and sentiment scores. The database schema includes collections for users, entries, and sentiments.
* **Schema Design:** Ensures efficient data retrieval and storage. Each user entry document includes fields for user ID, date, activity description, sentiment score, and color-coded sentiment.



***6.2.1 Assigning colour to the entries based on it’s sentiment score***

## Data Visualization

Data visualization is essential for interpreting complex datasets and identifying patterns. In this system, various graphical representations are employed, including line charts for displaying time series data like daily sentiment scores, bar charts for comparing average sentiment scores by week or month, and scatter plots for identifying correlations and outliers. The principles of clarity, simplicity, consistency, and interactivity guide the design of these visualizations. Interactive elements such as tooltips, zoom, pan, and filtering options enhance user engagement.



## **6.3.1 Graphical representation of the moods of the person over a period of time**

## Counselor Interface

A separate interface is provided for counselors to access and analyze patient data. This interface includes advanced features tailored to professional needs.

1. **Patient Selection:**

* **Dashboard:** Lists all patients associated with the counselor, with quick access to their mood trends and activity logs.
* **Search and Filter:** Enables counselors to search for specific patients and filter data based on various criteria.

2. **Advanced Analytics:**

* **Comparative Analysis:** Allows counselors to compare mood trends across multiple patients, identifying common patterns and deviations.
* **Detailed Reports:** Generates comprehensive reports summarizing patient progress, significant mood changes, and potential areas of concern.

# Results and future scope

## Enhancement of Sentiment Analysis Accuracy

While the current implementation utilizes the Gemini API for sentiment analysis, future improvements could focus on integrating more advanced or alternative sentiment analysis models. Exploring recent advancements in natural language processing (NLP) and machine learning could enhance the accuracy and granularity of sentiment detection. Customizing models to understand nuanced emotional expressions specific to psychological contexts or incorporating multimodal sentiment analysis (combining text with voice or facial expressions) could further improve the reliability of mood assessments.

## Expanded User and Counselor Features

To increase the effectiveness and utility of the platform, additional features could be introduced for both users and counselors. For users, incorporating personalized feedback mechanisms based on sentiment trends could offer actionable insights and suggestions for improving mental well-being. For counselors, advanced analytics features such as predictive modeling and personalized intervention recommendations could be added. Furthermore, incorporating real-time mood tracking and analysis could provide more immediate support and enhance the platform's responsiveness to users' mental health needs.

# conclusion

This paper has introduced a sophisticated web-based platform for psychological analysis through sentiment tracking and data visualization. Utilizing React for the front end, Flask for the back end, and MongoDB for data storage, the platform efficiently manages user interactions and sentiment analysis. The integration of the Gemini API enables precise sentiment scoring, which, combined with color-coded visualizations and trend graphs, provides users and counselors with valuable insights into emotional patterns and mental well-being.

Looking ahead, the platform's potential for enhancement is substantial. Future developments could include integrating advanced NLP models for more nuanced sentiment analysis, expanding features for personalized user feedback, and incorporating wearable technology for comprehensive well-being monitoring. Additionally, multilingual support and robust data security measures will broaden the platform's accessibility and ensure the protection of sensitive information. These advancements will further solidify the platform’s role as a pivotal tool in psychological analysis and mental health support.

##### References

1. M. A. Russell and J. Kim, "Real-Time Sentiment Analysis of Twitter Data using MongoDB and Node.js," IEEE Access, vol. 7, pp. 101452-101464, 2019. doi: 10.1109/ACCESS.2019.2929020.
2. A. Sharma, R. Gupta, and S. Agarwal, "A Review on Sentiment Analysis Approaches with Big Data in Social Media," in Proceedings of the 2021 International Conference on Advances in Computing, Communication, and Control (ICAC3), Mumbai, India, 2021, pp. 1-6. doi: 10.1109/ICAC345599.2021.9370382.
3. X. Liu, M. Gao, and H. Li, "A Framework for Sentiment Analysis in Big Data: Application to Stock Market Prediction," in Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 2018, pp. 2181-2190. doi: 10.1109/BigData.2018.8622552.
4. K. Patel and A. R. Yadav, "Real-Time Sentiment Analysis on Social Media Data Using Hadoop Framework and React.js," International Journal of Advanced Computer Science and Applications, vol. 11, no. 5, pp. 563-569, May 2020. doi: 10.14569/IJACSA.2020.0110574.
5. S. Mishra and P. Rath, "Scalable Sentiment Analysis of Big Data Using MongoDB and Apache Spark," in Proceedings of the 2020 IEEE 10th International Conference on Advanced Computing (IACC), Delhi, India, 2020, pp. 258-262. doi: 10.1109/IACC48062.2020.9074260.M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.
6. Y. Zhang, S. Wang, and Y. Li, "Sentiment Analysis of Chinese Microblog Using the Gemini Model," in Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 2019, pp. 2013-2022. doi: 10.1109/BigData47090.2019.9005993.
7. F. S. Mohammad, J. S. Radhakrishna, and V. K. Ganapathy, "Sentiment Analysis using MongoDB and Apache Kafka: A Case Study on Social Media Data," Journal of Big Data, vol. 8, no. 1, 2021, Art. no. 32. doi: 10.1186/s40537-021-00430-5.
8. H. Chen, Z. Zhou, and D. Liu, "Big Data Real-Time Processing and Sentiment Analysis in the Cloud Environment with MongoDB and Apache Storm," in Proceedings of the 2019 IEEE 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2019, pp. 55-60. doi: 10.1109/CONFLUENCE.2019.8776916.
9. R. Kumar and P. Jain, "A Hybrid Approach for Sentiment Analysis using Natural Language Processing and MongoDB," IEEE Transactions on Computational Social Systems, vol. 6, no. 3, pp. 501-510, June 2019. doi: 10.1109/TCSS.2019.2919873.
10. A. Gupta, S. Bhattacharya, and A. Agarwal, "Interactive Data Visualization for Big Data Using D3.js and React," in Proceedings of the 2020 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Sydney, NSW, Australia, 2020, pp. 343-350. doi: 10.1109/DSAA49011.2020.00051.