VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belgaum-590018

A PROJECT REPORT **(21CSP76)** ON

"WellMind : An Intelligent Platform for Adolescent Mental Health Support"

Submitted in Partial fulfillment of the Requirements for the Degree of

Bachelor of Engineering in Computer Science & Engineering

By

Deepti Shridhar Hegde (1CR21CS059)

Saikiran Kadwad (1CR21CS159)

Shivani TR (1CR21CS169)

Vaishnavi M (1CR21CS205)

Under the Guidance of,

Smitha N

Assistant Professor, Dept. of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI, BANGALORE-560037

CMR INSTITUTE OF TECHNOLOGY

#132, AECS LAYOUT, IT PARK ROAD, KUNDALAHALLI, BANGALORE-560037

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work entitled **"WellMind : An Intelligent Platform for Adolescent Mental Health Support"** carried out by **Ms**. **Deepti Shridhar Hegde**, USN **1CR21CS059**, **Mr**. **Saikiran Kadwad**, USN **1CR21CS159**,**Ms**. **Shivani TR**, USN **1CR21CS169, Ms**. **Vaishnavi M**, USN **1CR21CS205,** bonafide students of CMR Institute of Technology, in partial fulfillment for the award of **Bachelor of Engineering** in Computer Science and Engineering of the Visveswaraiah Technological University, Belgaum during the year 2024-2025. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree**.**

.

(Smitha N)	**Dr.R Kesavamoorthy**	**Dr. Sanjay Jain**
(Assistant professor)	**Professor & Head**	**Principal**
Dept. of CSE, CMRIT	**Dept. of CSE, CMRIT**	**CMRIT**

External Viva

Name of the Examiners — Signature with Date

1. ____________________________ ________________________

2. ____________________________ ________________________

DECLARATION

We, the students of Computer Science and Engineering, CMR Institute of Technology, Bangalore declare that the work entitled " **WellMind : An Intelligent Platform for Adolescent Mental Health Support** " has been successfully completed under the guidance of Prof.Smitha N, Computer Science and Engineering Department, CMR Institute of Technology, Bangalore. This dissertation work is submitted in partial fulfillment of the requirements for the award of Degree of Bachelor of Engineering in Computer Science and Engineering during the academic year 2024 - 2025. Further, the matter embodied in the project report has not been submitted previously by anybody for the award of any degree or diploma to any university.

Place:

Date:

Team members:	**Signature**
Deepti Shridhar Hegde (1CR21CS059)	___________________
Saikiran Kadwad (1CR21CS159)	___________________
Shivani TR (1CR21CS169)	___________________
Vaishnavi M (1CR21CS205)	___________________

ABSTRACT

The WellMind Project is a unique tool designed to assess and monitor stress levels across three phases: Pre-activity, During activity, and Post-activity. In the Pre-activity phase, users provide responses to determine their baseline stress levels. The During activity phase involves engaging users with scenario-based questions, simple math problems, and recommended YouTube videos to observe stress changes in real-time. The Post-activity phase evaluates the impact of the activity through follow-up questions, allowing a comparison with the baseline. Using a Random Forest Classifier, the system predicts stress patterns, visualizes stress changes, and calculates the percentage of stress reduction or increase. By offering real-time monitoring, insightful analysis, and interactive tasks, WellMind helps users gain better self-awareness and manage stress effectively.

ACKNOWLEDGEMENT

I take this opportunity to express my sincere gratitude and respect to **CMR Institute of Technology, Bengaluru** for providing me a platform to pursue my studies and carry out my final year project

I have a great pleasure in expressing my deep sense of gratitude to **Dr. Sanjay Jain,** Principal, CMRIT, Bangalore, for his constant encouragement.

I would like to thank **Dr.R Kesavamoorthy,** Professor and Head, Department of Computer Science and Engineering, CMRIT, Bangalore, who has been a constant support and encouragement throughout the course of this project.

I consider it a privilege and honor to express my sincere gratitude to my guide **Smitha N, Assistant Professor,** Department of Computer Science and Engineering, for the valuable guidance throughout the tenure of this review.

I also extend my thanks to all the faculty of Computer Science and Engineering who directly or indirectly encouraged me.

Finally, I would like to thank my parents and friends for all their moral support they have given me during the completion of this work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ML	**Machine Learning**
RF	**Random Forest**
API	**Application Programming Interface**
UI	**User Interface**
DB	**Database**

CHAPTER 1

INTRODUCTION

Stress is an inevitable part of life, and learning how to manage it effectively is essential for both mental and physical well-being. The WellMind Project is an innovative program designed to monitor and analyze how stress levels change before, during, and after engaging in specific activities. Whether it involves mindfulness exercises, problem-solving tasks, or relaxation techniques, the project helps identify the most effective methods for reducing stress and enhancing overall well-being.

The project follows a structured yet straightforward process. It begins by assessing participants' stress levels through a set of questions before they start an activity, establishing a baseline for comparison. During the activity, participants engage in various tasks, such as watching calming videos, solving puzzles, or reflecting on different scenarios. After completing the activity, participants provide additional feedback to determine how their stress levels have shifted.

What sets the WellMind Project apart is its use of advanced data analysis and machine learning to interpret participants' responses. These responses are categorized into stress levels—low, medium, or high—at each phase. By comparing the results, the system calculates the overall effectiveness of the activity in reducing stress. This approach not only highlights the impact of each intervention but also identifies patterns that might otherwise go unnoticed.

To ensure clarity and accessibility, the project presents its findings using visual tools such as bar charts. These visuals simplify complex data, making it easy for individuals and researchers alike to understand the changes in stress levels. This makes the project not only a scientific tool but also a practical resource for anyone seeking to monitor and improve their stress management practices.

1.1 Relevance of the Project

The WellMind project is highly relevant in today's educational environment, where stress and mental health issues are becoming increasingly prevalent among students. By utilizing machine learning to assess and predict stress levels in real-time, the platform offers a proactive solution to monitor and manage mental health. This can greatly benefit educational institutions by providing data-driven insights that help in identifying students in need of support, fostering a healthier learning environment, and promoting overall well-being. Additionally, it empowers students with the tools to track their stress levels and make informed decisions to improve their mental health.

1.2 Problem Statement

Mental health issues, particularly stress, have become a significant concern among students, adversely affecting their academic performance and overall well-being. With the growing pressures of academic expectations, personal challenges, and social dynamics, students often struggle to manage their mental health effectively. While some support systems like counseling services exist, they are not always accessible, timely, or tailored to individual needs. Furthermore, the lack of real-time tools to monitor and assess stress levels leaves students without immediate support, potentially worsening their condition.

To address this gap, there is a need for an innovative solution that enables students to assess and manage their stress proactively. WellMind is a project aimed at developing a web-based platform that leverages machine learning to analyze stress levels through user input. By providing personalized stress assessments and actionable insights, the platform empowers students to monitor their mental health continuously. In doing so, it offers a scalable, data-driven solution that can be utilized by both individuals and educational institutions to enhance student well-being and support timely interventions.

1.3 Objectives

The primary objective of the WellMind project is to create a user-friendly platform that helps individuals better understand and manage their stress levels. By leveraging interactive tasks, data-driven insights, and machine learning, the project aims to provide users with actionable information about their mental well-being. Through seamless integration of engaging activities and advanced analytics, the platform aspires to:

- Evaluate stress levels across different phases (before, during, and after an activity).

- Identify patterns in stress responses to various stimuli, helping users gain deeper awareness of their mental health.
- Offer visual feedback through easy-to-understand data visualizations, empowering users to make informed decisions for stress management.

1.4 Scope of the project

The WellMind Project has the potential to make a meaningful impact across various sectors by offering valuable insights into stress management and evaluating the effectiveness of different interventions. In mental health, it can assess the impact of techniques like mindfulness and meditation by measuring how stress levels change before and after these activities. In education, the project tracks student stress levels before and after exams or project deadlines, helping educators understand how stress-relief strategies influence students' well-being. In the corporate world, it can measure the effectiveness of workplace interventions such as relaxation exercises or team-building activities, enhancing employee wellness. Additionally, in clinical settings, it helps track stress levels in patients undergoing treatment while providing individuals with tools to monitor and manage their personal stress.

Beyond its immediate applications, the WellMind Project also offers potential for broader research in stress management. However, certain challenges remain, such as the accuracy of stress predictions, which depends on the quality of the survey responses, and the need for the model to generalize across diverse user experiences. In the future, the project could integrate real-time data from wearable devices, like heart rate or skin conductivity, and provide personalized recommendations based on an individual's stress patterns. Despite these challenges, the WellMind Project's data-driven insights offer valuable information on how stress fluctuates during and after interventions, making it a powerful tool for enhancing stress management strategies in various fields, including mental health, education, corporate wellness, and clinical care.

1.5 Software Engineering Methodology

The WellMind Project adopts an Agile approach, focusing on iterative development and incorporating regular feedback. The work is divided into smaller phases, with each phase addressing specific functionalities, such as data collection, real-time analysis, and visualization. Frequent testing ensures that the system aligns with user needs and operates efficiently. Collaboration between team members, including developers and machine learning experts, ensures smooth integration of the frontend, backend, and analytical components. This methodology allows flexibility, quick adjustments to requirements, and the delivery of a reliable mental health assessment system.

Fig 1.1 Agile

The above fig 1.1 shows a description of the Agile methodology used in the project which includes design, development, testing and deployment which is discussed in the later chapters of this report.

1.6 Tools and Technologies

The WellMind platform integrates various modern tools and technologies to provide a seamless, user-friendly, and reliable solution for mental health assessment and stress detection. Below is a breakdown of the key tools and technologies utilized:

HTML/CSS/JavaScript: These essential technologies were used to design the structure and styling of the user interface, creating a clean, interactive, and responsive web experience.

React.js: React.js, a widely-used JavaScript library, was employed for building the frontend, ensuring the platform's interface is dynamic, user-friendly, and responsive, enhancing the overall user experience.

Python (Flask): Flask, a lightweight Python framework, was used to handle the server-side logic, manage user requests, and integrate machine learning models for predicting stress levels. This framework supports fast development and makes API interactions smooth and efficient.

MongoDB: MongoDB, a NoSQL database, was chosen for its flexibility in managing and storing unstructured data. It offers scalability and efficiency, making it ideal for storing user data, stress assessment results, and related information.

Scikit-learn: This Python library was used to implement machine learning models, particularly the Random Forest Classifier, which predicts stress levels based on data input from users. It is highly effective for classification tasks and model training.

Data Processing Libraries: For data manipulation and preparation, libraries such as Pandas and Numpy were utilized to efficiently handle large datasets and perform necessary transformations, ensuring data is in the right format for machine learning algorithms.

Visualization Tools: To present the findings in an understandable format, Matplotlib and Seaborn were employed to generate visualizations such as graphs and charts, allowing for easy comparison of stress levels and other metrics over time.

Jupyter Notebook: Jupyter Notebook provided an interactive environment for experimenting with data, testing machine learning models, and analyzing results before final implementation in the platform.

VS Code: Visual Studio Code was the main IDE used for coding, debugging, and managing the entire project, offering a comprehensive and efficient development environment.

Postman: Postman was employed to test and debug the APIs, ensuring seamless communication between the frontend and backend components of the platform.

1.7 Chapter Wise Summary

- Introduction: Introduces WellMind, a project aimed at monitoring and managing stress levels through user activities and machine learning.

- Literature Review: Examines current mental health technologies, identifying gaps that WellMind addresses with innovative solutions.
- Proposed Model: Outlines the platform’s design, featuring a React.js interface, Flask backend, MongoDB database, and a Random Forest-based prediction model.
- Implementation: Describes the modular development process, utilizing modern tools for user interaction, data handling, and stress visualization.
- Results: Highlights stress reduction after activities and demonstrates the accuracy of the machine learning model through evaluation metrics.
- Testing: Focuses on activities to explore stress contributors like anxiety and depression, refining the system based on user feedback.
- Conclusion: Summarizes the project’s success and proposes future advancements like real-time tracking and personalized interventions.

CHAPTER 2

LITERATURE SURVEY

This literature study aims to give an in-depth analysis of current studies, technologies. It is crucial to comprehend the current research in these areas because the project's goal is to improve mental health through the use of assistive technologies. With the aid of this review, significant advancements, new trends may be found that can direct the technical conception, moral considerations, and operational aspects of ”Mind-Aid.” It will also draw attention to shortcomings in the standards and technologies that are in use today, laying the groundwork for the innovation and progress that ”Mind-Aid” seeks to accomplish.

2.1 A Review of Gamification Related to Mental Health Treatment: Mental health conditions in Indonesia affect around 20 percent of the population, yet the country faces a shortage of professional psychiatrists and limited healthcare resources, making access to treatment difficult. Gamified treatment methods, such as relaxation applications, have emerged as cost-effective alternatives to traditional therapies. This paper reviews 32 research studies and a survey of 28 participants, finding that gamification can positively impact mental health by improving mood, reducing stress, and alleviating anxiety. The growing accessibility of gamified mental health 9 treatments, especially post-COVID-19, points to its potential for wider future adoption.

2.2 The Need for an Adaptive Sociotechnical Model for Managing Mental Health in a Pandemic: The COVID-19 pandemic has significantly increased mental health distress due to isolation, prompting research into technological solutions. This study reviews existing technologies for managing mental health challenges and proposes a sociotechnical model to evaluate their effectiveness. Despite the availability of various tools, no comprehensive, affordable mobile application exists to address anxiety and stress specifically during the pandemic. The proposed model aims to create a user-friendly mobile app that assesses user needs through a brief questionnaire, facilitating quicker and more effective coping strategies. Future research will expand this work to include community-based interventions for mutual support during such crises.

2.3 Preliminary Outcomes of a Culturally Tailored Mindfulness Mobile App for Mental Health within Underserved African American Communities During COVID-19: This poster presents preliminary findings from a study in progress, which indicate that a speciallydesigned mobile app can serve as a viable delivery system for a mindfulness-based stress reduction (MBSR)intervention for historically underserved and minority communities experiencing mental health effects of worry and stress during the COVID-19 pandemic. Our early data suggest that the app is wellreceived and usable for the target population and serves as an effective delivery platform for MBSR interventions for underserved and minority communities.

2.4 Mental Health Indices as Biomarkers for Assistive Mental Healthcare in University Students : This paper addresses the challenges of evaluating common mental health disorders in remote healthcare systems, where symptoms are difficult to quantify. We propose a methodology for assessing the correlation between three mental health indices—MHI-5, WHO-5, and PHQ-9 using ecological momentary assessments from 235 university students. Our study quantifies correlations, evaluates student perceptions of their mental health, and compares index scores with official diagnoses. We advocate for continuous mental health monitoring to aid campus counselors and psychologists in tracking student well-being and addressing mental distress. Future work aims to validate our findings through a bipartite control group study and enhance the integration of assessment and intervention strategies on campus

2.5 Application of Computer Technology in College Students' Mental Health Education : Mental health education is a vital aspect of college education, and the integration of computer 12technology has become increasingly common in addressing students' mental health challenges. This paper introduces two key systems: a mental file management system and a mental health auxiliary teaching platform. The file management system uses JAVA and MySQL for mental health assessments, enabling mobile terminal detection and data export. The auxiliary teaching platform applies SQL Server2008 and ASP.NET MVC for improved functionality. These technologies enhance the efficiency and personalization of mental health education, offering valuable insights for future innovation in mental health support

2.6 WIP: Faculty Perceptions of Graduate Student Mental Health: A Productivity Framing : Research highlights an increasing mental health crisis in graduate education, affecting productivity, retention, and overall well-being. This study investigates faculty perceptions regarding graduate student mental health and how these perceptions influence their actions when mental health challenges emerge. Through phenomenological inquiry, we examined faculty attitudes (n = 3) and their impact on programmatic and individual support decisions. Thematic analysis of interviews revealed diverse faculty views on stress and mental health, with work productivity often driving these conversations. Findings suggest the need for improving faculty comfort in discussing mental health and establishing clear institutional policies to support student well-bein

Reference	Technique	Objective/ Features	Existing Algorithms	Advantages	Demerits
[1]	Gamification, mobile apps	Explores the potential of gamified treatments like relaxation apps to improve mental health, especially in Indonesia with limited psychiatric resources.	None explicitly mentioned	Cost-effective, accessible, and engaging way to treat mental health issue	May lack clinical validation, user engagement can vary.
[2]	The Need for an Adaptive Sociotechnical Model for Managing Mental Health in a Pandemic	Proposes a user-friendly app to assess mental he -ath needs and provide coping strategies during pandemics.	None explicitly mentioned	Tailored solutions for pandemic-driven mental health, user-centric model.	No existing comprehensive mobile solution, requires further research.

[3]	Mobile app, mindfulness-based stress reduction (MBSR)	To deliver a culturally tailored mindfulness intervention to underserved communities for stress reduction during COVID-19.	Mindfulness-based stress reduction (MBSR)	Effective for underserved populations, easy to use and culturally relevant	Needs more data for broader application, scalability challenges.
[4]	Ecological momentary assessments (EMA), mental health indices	Evaluates the correlation between mental health indices (MHI-5, WHO 5, PHQ-9) and perceptions of mental health in university students.	MHI-5, WHO-5, PHQ-9	Continuous monitoring of student well-being, aids in campus mental health interventions.	Requires validation with a control group, limited generalizability.
[5]	JAVA, MySQL, SQL Server2008, ASP.NET MVC	Discusses the integration of computer systems for mental health management, including file management and teaching platforms.	JAVA, MySQL, SQL Server2008, ASP.NET MVC	Personalized, efficient mental health support in educational settings.	May require customization for different contexts, potential integration issues.
[6]	Phenomenological Inquiry	Investigate faculty perceptions of graduate student mental health, and their impact on support decisions.	None (Qualitative research)	Provides insights into faculty attitudes towards mental health	Limited sample size (n=3); subjective data.

Table 2.1 Comparison of various algorithms

CHAPTER 3

PROPOSED MODEL

The WellMind Project utilizes a Random Forest Classifier to analyze stress levels during Pre-activity, During activity, and Post-activity phases. By leveraging user responses and combining multiple decision trees, the model provides precise stress predictions. This approach helps identify stress patterns, visualize insights, and promote effective mental well-being strategies.

3.1 System Architecture

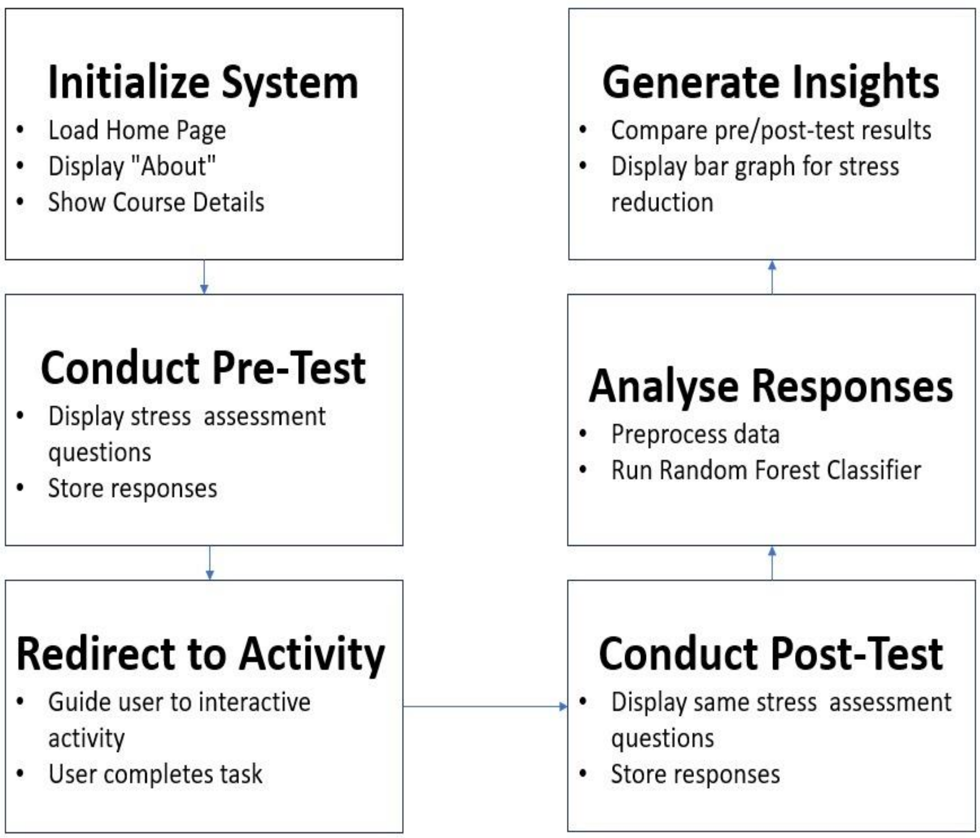

Fig 3.1 System Architecture

The system architecture of the project provides a comprehensive framework that defines the interaction between different components. It is designed to ensure seamless functionality and efficient data flow throughout the platform.

1. Frontend Layer: The frontend, developed using React.js along with HTML, CSS, and JavaScript, provides an interactive and user-friendly interface. It allows users to register or log in, complete stress assessments, and view results in a visually engaging manner. The interface ensures a smooth user experience by handling real-time interactions and requests.

2. Backend Layer: The backend is implemented using Flask, a lightweight Python web framework. This layer processes user requests, handles business logic, and facilitates communication between the frontend and the database. It also integrates the machine learning model to predict stress levels based on the user's inputs.

3. Database Layer: The system uses MongoDB, a NoSQL database, to store and manage user information, assessment responses, and analysis results. Its flexible schema enables efficient storage of structured and unstructured data.

4. Machine Learning Model: The trained Random Forest Classifier processes the data collected from users to predict stress levels. Tools like Pandas and NumPy handle data preprocessing, while Scikit-learn is used to build and evaluate the machine learning model.

5. Visualization: Data visualization tools like Matplotli and Seaborn generate charts and graphs, providing users with insights into their stress patterns across different phases.

6. API Communication: The frontend communicates with the backend using Axios, making HTTP requests to Flask API endpoints. Flask-CORS ensures secure cross-origin requests, enabling smooth interaction between the frontend and backend.

This architecture ensures modularity, scalability, and robust functionality, making it easy to analyze user stress patterns and deliver actionable insights.

3.2 Flow Chart

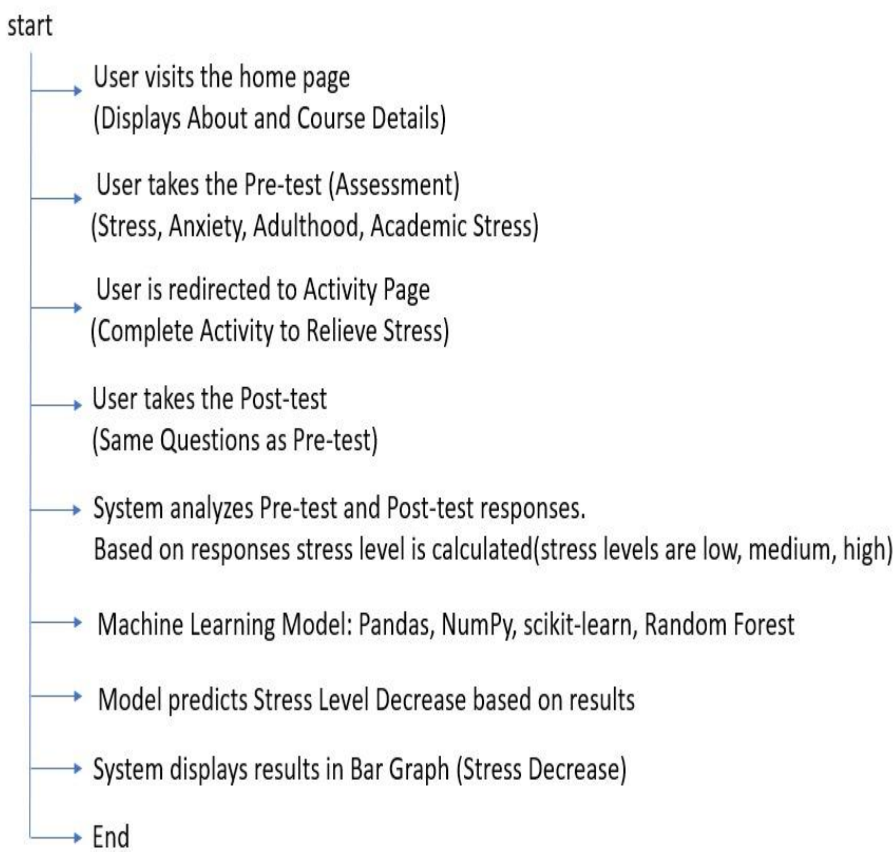

Fig 3.2 Flowchart for the Project

The flowchart represents the systematic workflow of the project, outlining the sequence of steps involved in assessing and analyzing user stress levels. It begins with the user login or registration process, where users provide their credentials to access the platform. After successful authentication, thepre-activity stress assessment phase is initiated. Here, users answer a set of questions designed to evaluate their baseline stress levels.

Next, the system transitions to the activity phase, where users engage with tasks such as scenario-based questions, mathematical problems, or video interactions. The real-time responses and interactions are analyzed to observe stress level fluctuations during the activity.

Following the activity, users proceed to the post-activity stress assessment, where they respond to follow-up questions to determine changes in stress levels. The collected data from all three phases is then processed by the backend, utilizing a trained Random Forest Classifier to predict stress levels.

Finally, the results are compiled and visualized using charts to provide a comparative analysis of stress levels across the phases. The flowchart ensures a clear depiction of the project's logical steps, helping to understand the interaction between users and the system.

3.3 Random Forest Architecture

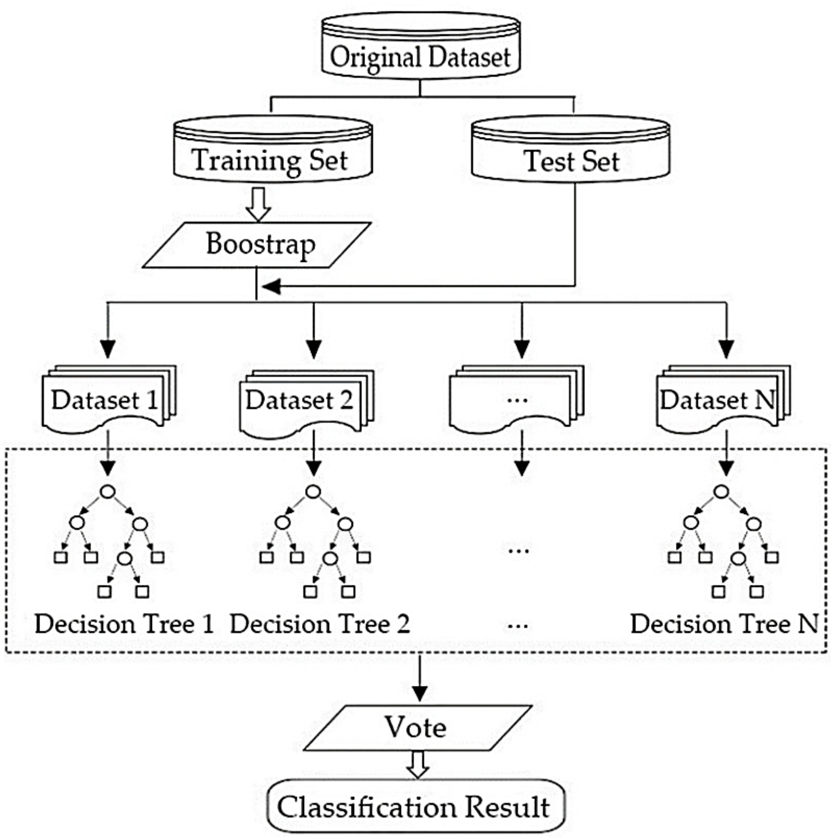

Fig 3.3 Random Forest Architecture

1. Input Collection :-Collect user responses across the three phases: Pre-activity, During activity, and Post-activity. Inputs include textual answers, numerical responses, and interaction data.

2. Data Preparation:- Transform categorical inputs (like phases and responses) into numerical values using encoding methods. Ensure the dataset is cleaned and standardized for further processing.

3. Feature Selection:- Identify critical features, such as the activity phase, question types, and user responses, to accurately predict stress levels.

4. Model Training :- Create multiple decision trees by sampling different portions of the dataset. Train each tree separately to ensure diverse learning and minimize overfitting.

5. Prediction of Stress Levels:- Pass user inputs through all trained decision trees. Combine their outputs through majority voting to generate the final stress level prediction.

6. Result Analysis :- Predict stress levels for each phase: Pre-activity, During activity, and Post-activity. Compute average stress levels and evaluate variations between phases.

7. Visualization and Insights:- Display stress levels using visual tools like graphs to provide users with clear insights into their mental well-being and progress.

This refined approach ensures an effective and accurate stress level prediction using Random Forest Classifier.

CHAPTER 4

IMPLEMENTATION

The implementation of the WellMind project combines various advanced technologies and follows a modular approach to deliver a seamless, efficient, and user-centric mental health platform. It is structured into four key components: the frontend, backend, database, and machine learning module, ensuring a robust system architecture.

The frontend is developed using HTML, CSS, and JavaScript, complemented by React.js for creating a responsive and dynamic user interface. Features include secure user authentication, stress assessment questionnaires, and real-time data visualization through graphs and charts. These elements work together to ensure an intuitive and engaging experience for users.

The backend is built with Python and the Flask framework, responsible for processing user requests, implementing business logic, and integrating machine learning predictions. Communication between the frontend and backend is secured with Flask-CORS, ensuring seamless data flow and protection against cross-origin vulnerabilities.

MongoDB serves as the database, chosen for its scalability and flexibility in managing structured and semi-structured data. It stores critical information such as user profiles, stress assessment responses, and analysis results. The database design supports efficient retrieval and storage of data, enabling smooth system operation.

The machine learning module, powered by Scikit-learn, features a Random Forest Classifier trained on labeled datasets to predict stress levels accurately. Preprocessing of user data is performed using Pandas and NumPy, ensuring the model's compatibility and efficiency. Once trained, the model is integrated into the backend for real-time stress analysis.

To provide insights, Matplotlib and Seaborn are utilized for visualizing stress trends and comparisons, offering users actionable feedback on their mental health patterns. Development and testing tools like VS Code, Postman, and Jupyter Notebook facilitate streamlined coding, API validation, and machine learning experimentation.

This implementation ensures that WellMind operates as a reliable and effective platform for assessing and managing mental health in a user-friendly manner.

4.1 Modular and Procedural Approach

The development of the WellMind platform is based on a modular and procedural methodology to ensure scalability, maintainability, and clarity. This approach organizes the project into distinct modules, each with a specific function, and follows a systematic sequence of steps to achieve the desired results.

The project comprises the following essential modules:

1. Frontend Module: Manages user interactions and displays user-friendly interfaces. Built using HTML, CSS, and React.js to ensure responsive and reusablecomponents.

2. Backend Module: Handles data flow between the frontend andatabase.Implements logic and API endpoints using Flask.

3. Database Module: Stores user details, stress responses, and analysis outcomes. Utilizes MongoDB for efficient and adaptable data storage.

4. Machine Learning Module:Processes user input to predict stress levels. Developed using Scikit-learn and incorporates a Random Forest Classifier for accurate predictions.

5.Visualization Module:Creates graphical displays of results. Uses libraries like Matplolib and Seaborn for clear and interactive data visualization.

The procedural method includes the following steps:

1. Requirement Gathering: Identify the system's needs and define objectives.

2. Design and Planning: Develop wireframes, flowcharts, and database designs to visualize system structure.

3. Frontend Development: Create a user-friendly interface with seamless navigation.

4. Backend Integration: Develop APIs and connect the machine learning model to the Flask backend for real-time predictions.

5. Database Configuration: Set up MongoDB for efficient data storage and retrieval.

6. Model Development: Train the Random Forest Classifier on preprocessed datasets for precise stress level predictions.

7. Testing and Validation: Conduct detailed testing to ensure seamless integration and reliable data flow.

8. Deployment: Launch the platform, ensuring compatibility across devices and accessibility for end users.

4.2 Comparative Study

1. Data Gathering: Collect the stress level data from users during the Pre-activity and Post-activity phases.

2. Data Preprocessing: Clean, standardize, and arrange the data to ensure it is suitable for analysis.

3. Compute Averages: Calculate the average stress level for both the Pre-activity and Post-activity phases.

4. Analyze Differences: Find the difference between the average stress levels of the Pre-activity and Post-activity phases.

5. Percentage Change Calculation: Calculate the percentage change in stress using the formula:

$$\text{Percentage Change} = \frac{\text{Pre-activity Level} - \text{Post-activity Level}}{\text{Pre-activity Level}} \times 100$$

$\longrightarrow$ Equation (1)

- The above equation is used to calculate the percentage change between two levels, typically used to compare a pre-activity and post-activity.

6. Statistical Significance Testing: Determine if the change in stress levels is statistically significant.

7. Categorize Outcomes: Classify the result as either a reduction, increase, or no significant change in stress levels based on the percentage change.

8. Visualization: Create a graph or chart to visually represent the comparison of stress levels before and after the activity.

9. Generate Feedback: Provide users with feedback on how the activity influenced their stress levels.

4.3 Random Forest Algorithm

1. Data Collection: Gather user responses from the pre-activity, during-activity, and post-activity phases.

2. Data Preprocessing: Prepare the data by cleaning, normalizing, and encoding it using tools like Pandas and NumPy.

3. Dataset Splitting: Divide the dataset into training and testing sets, such as 80% for training and 20% for testing.

4. Model Training: Use the Random Forest Classifier to train on subsets of the data with multiple decision trees.

5. Feature Analysis: Determine the most significant factors influencing stress levels using feature importance.

6. Model Evaluation: Assess the model's accuracy and performance with evaluation metrics like precision and F1-score.

7. Integration: Embed the trained model into the Flask backend to enable real-time stress level predictions.

8. Prediction: Generate predictions for stress levels (e.g., Low, Moderate, High) based on user inputs for each phase.

9. Visualization: Create visual representations of stress level trends and comparisons using Matplotlib or Seaborn.

4.4 Machine Learning Code

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier

# Load the dataset
df = pd.read_csv("responses_with_stress_level.csv")

# Preprocessing: Encoding categorical columns into numeric values
encoder = LabelEncoder()

# Encode 'Phase', 'Question', and 'Response' columns
df['Phase'] = encoder.fit_transform(df['Phase'])
df['Question'] = encoder.fit_transform(df['Question'])
df['Response'] = encoder.fit_transform(df['Response'])

# Split the data into features (X) and target (y)
X = df[['Question', 'Response', 'Phase']]  # Features
y = df['Stress_Level']  # Target

# Split the data into training (80%) and testing (20%) sets
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Initialize and train the Random Forest Classifier
clf = RandomForestClassifier(n_estimators=100, random_state=42)
clf.fit(X_train, y_train)
```

```
# Function to calculate the average stress level for a given phase
def calculate_average_stress_level(phase, model, data):
    # Filter the data by the given phase (Pretest/Posttest)
    phase_data = data[data['Phase'] == phase]

    # Extract features (Question, Response, Phase) for prediction
    features = phase_data[['Question', 'Response', 'Phase']]

    # Predict stress levels using the trained model
    predicted_stress = model.predict(features)

    # Calculate and return the average predicted stress level
    average_stress = predicted_stress.mean()
    return average_stress
```

4.5 Code

FrontEnd Code:-

```
import React, { createRef, useEffect, useState } from "react";
import { isDataArray, typeOfData } from "../../utils/Tools";
import Layout from "../layout/Layout";
import ModuleHeader from "../module-main/modules-header/ModuleHeader";
import Accordion from '@mui/material/Accordion';
import AccordionSummary from '@mui/material/AccordionSummary';
import AccordionDetails from '@mui/material/AccordionDetails';
import ExpandMoreIcon from '@mui/icons-material/ExpandMore';
import "./ActivityResults.css";
import { useParams } from 'react-router-dom';
import * as imports from "../../Imports";
```

```
import toastr from "toastr";
import { apiGetRequest, apiPostRequest } from "../../utils/Network";
import Dialog from "../../components/dialogue/Dialog";
import { all } from "axios";
import Loader from "../../components/loader/Loader";
const modules = [
  { modulename: "Module1", num: 20 },
  { modulename: "Module2", num: 30 },
  { modulename: "Module3", num: 40 },
  { modulename: "Module4", num: 50 },
];

export default function ActivityResults() {
  const [resultData, setResultData] = useState(null);
  const [showPopup, setPopup] = useState(false);
  const [open, setOpen] = useState(false);
  const [feedback, setFeedback] = useState(false);
  const [certificate, setCertificate] = useState(false);
  const [mlResult, setMlResult] = useState(false);
  const [btnDisable, setBtnDisable] = useState(true);
  const [loader, setLoader] = React.useState(false);
 const handleClose = () => {
    setOpen(false);
    setPopup(false);
    setCertificate(false);
    setFeedback(false);
  };
```

BackEnd Code:-

```
from dependencies import *
from flask import Flask
app = Flask(_name_, static_folder='templates/static',static_url_path='/static')
cors = CORS(app, resources={r"": {"origins": ""}}, supports_credentials=True)

SECRET_KEY = "mindaid"
app.config['SECRET_KEY'] = '123marutidocksuzuki321'
app.config['MONGO_DBNAME'] = 'MINDTEST'
app.config['MONGO_URI'] = 'mongodb://localhost:27017/MINDTEST'
app.config['JWT_SECRET_KEY'] = 'super-secret'
app.config['JWT_TOKEN_LOCATION'] = ['headers']
app.config['JWT_HEADER_NAME'] = 'Authorization'

save_dir = os.path.join(os.getcwd(), "../../mind-aid-development/src/images/")
save_path = os.path.join(save_dir, "edited_certificate.jpg")
```

4.6 Performance Metrics

The following metrics are calculated from the confusion matrix to assess model performance:

1. Accuracy: The ratio of correct predictions (both True Positives and True Negatives) to total predictions.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

→Equation (2)

- In the above equation Accuracy is calculated by dividing the sum of true positives(TP) and true negatives(TN) by the total number of observations.

2. Precision: Measures how many of the predicted positive stress levels were actually correct.

$$Precision = \frac{TP}{TP + FP} \longrightarrow \text{Equation (3)}$$

- In the above equation Precision is calculated by dividing the numbr of true positives(TP) by the sum of true positives(TP) and false Positives(FP).

3. Recall (Sensitivity): Shows how well the model identifies all true positive stress levels.

$$Recall = \frac{TP}{TP + FN} \longrightarrow \text{Equation (4)}$$

- In the above equation Recall is calculated by dividing the number of true poitives(TP) by the sum of true positives(TP) and False negative(FN).

4. F1 Score: The harmonic mean of precision and recall, providing a balance between the two when they are in conflict.

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \longrightarrow \text{Equation (5)}$$

- In the above equation F1 is calculated by taking twice the product of Precision and Recall and dividing it by their sum.

5. Specificity: Measures how well the model identifies negative stress levels (i.e., not stressed).

$$\text{Specificity} \Longrightarrow \frac{\text{TN}}{\text{TN + FP}} \longrightarrow \text{Equation (6)}$$

- In the above equation Specificity is calculated by dividing the number of true negatives(TN) by the sum of true negatives(TN) and false positives(FP).

4.6.1 Confusion Matrix

A confusion matrix is a tool used to assess the performance of a classification model by comparing its predictions to the actual values. In the case of your stress level prediction, it helps identify how accurately the model classifies different stress levels (e.g., low, medium, high).For a binary classification, the confusion matrix could be represented as:

	Predicted Low Stress	**Predicted High Stress**
Actual Low Stress	True Negative(TN)	False Positive(FP)
Actual High Stress	False Negative(FN)	True Positive(TP)

Table 4.1 Confusion Matrix

CHAPTER 5

RESULTS AND DISCUSSION

The findings from the WellMind project show that the Random Forest Classifier is proficient in predicting stress levels before and after an activity. The evaluation metrics, such as accuracy, precision, recall, and F1 score, indicate that the model performs well in classifying stress levels with minimal errors. Most users showed a noticeable decrease in stress levels post-activity, as reflected in the percentage change calculation. The confusion matrix confirms that the model accurately distinguishes between low and high stress, though a few misclassifications were observed. While the model's predictions of stress fluctuations are effective, additional tuning and a larger dataset could enhance its accuracy and adaptability across different users.

Fig.5.1 Home page

In the above Fig. 5.1, once a user accesses this website URL (Mind-Aid), the landing page will automatically load. Since the user has not logged in, they will only receive a brief overview of the website.

Fig.5.2 About page

In the above Fig. 5.2, An overview of the Mind-Aid website can be found in the About Section. The purpose of this part is to familiarize users with the main goals and features of the platform.

Fig.5.3 Courses Section

In the above Fig. 5.3, when users select the courses button, they can see all the 9 modules details.

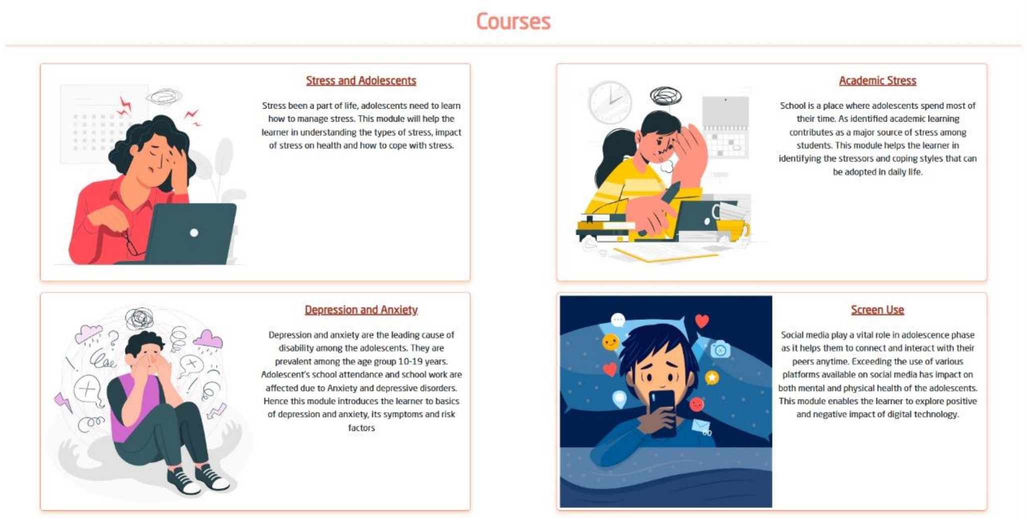

Fig.5.4 Course description

In the above Fig. 5.4, when users clicks on courses button of the fig. 6.3, this is the modules details page which we get.

Fig.5.5 Available courses

In the above Fig. 5.5, There are four different stages of progress for each module: Pending, Not Started, In Progress, and Completed. As learners continue through the modules, they may track their progress, ensuring that they are continually working toward their objective of mental well-being.

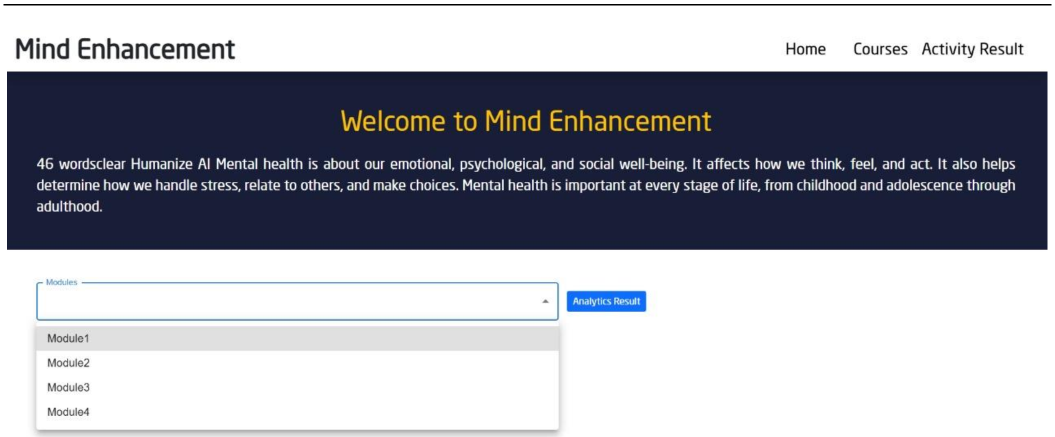

Fig.5.6 Modules Completion

In the above Fig. 5.6, Students can also see the results of four modules on the results screen. From there, they can select a particular module and see the questions and answers they have already answered.

Fig.5.7 Result

In the above Fig. 5.7 It is showing the stress level comparision before activity and after activity

CHAPTER 6

TESTING

In this section, we have highlighted some testcase that we used to check our results in the application developed.it consists of 3 phases pre-test, activity and pro-test, each phases includes questions that the user needs to answer to complete the testing process.

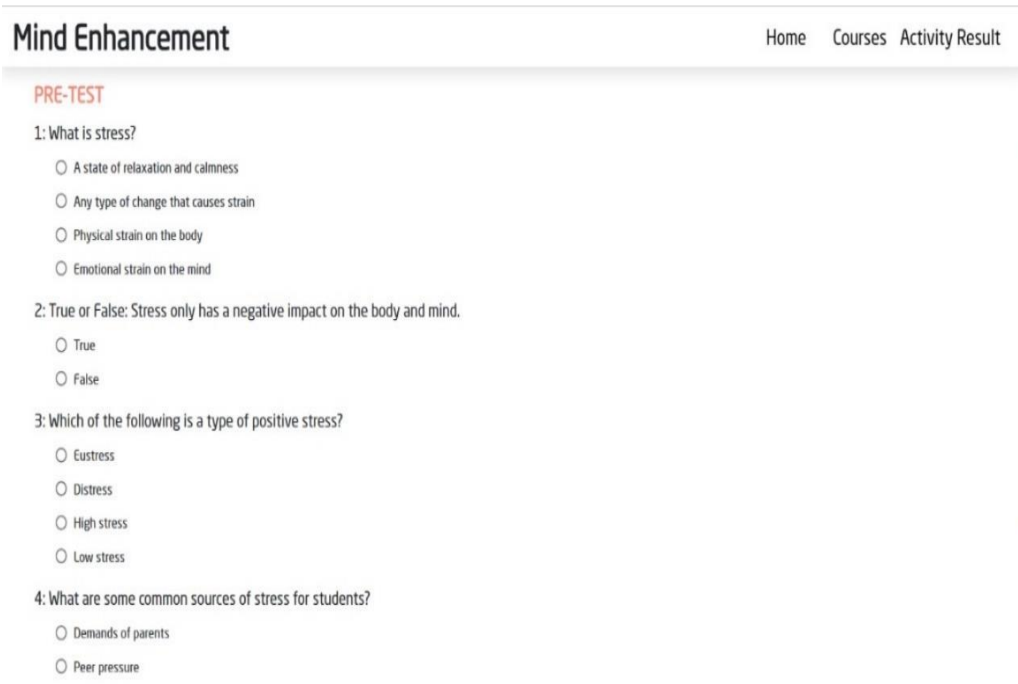

Fig.6.1 Pre-test

In the above Fig. 6.1 The pre-test is an initial assessment where users answer questions related to stress, anxiety, adulthood, and academic stress. This helps to establish a baseline for the user's mental health status before engaging in any activity.

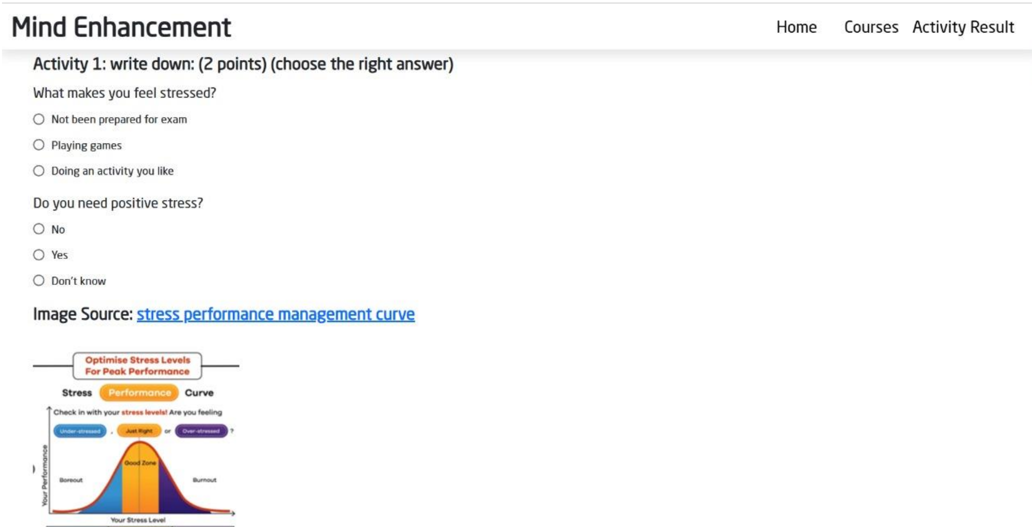

Fig.6.2 Activity module

In the above Fig. 6.2 After completing the pre-test, users engage in scenario-based questions and case studies designed to provoke thoughtful responses. These activities aim to help users reflect on their stressors and promote mental well-being.

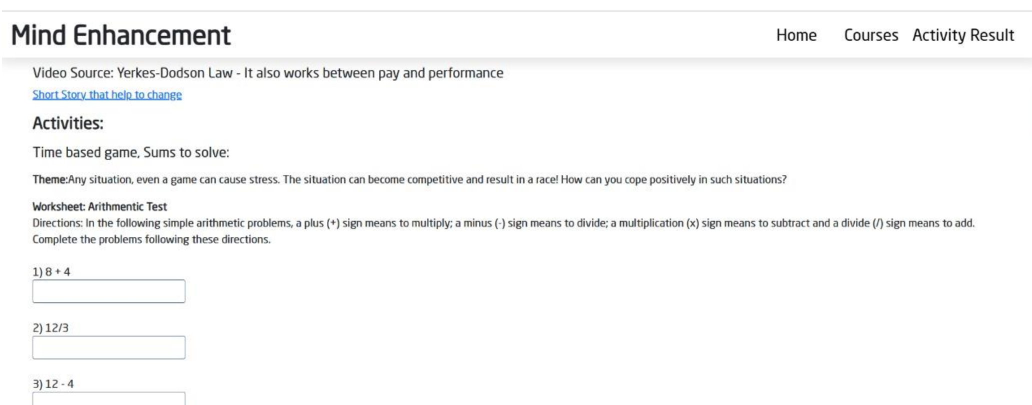

Fig 6.3 Arithmetic Test

In the above Fig. 6.3 there are questions related to arithmetic to identify the mental state of the user.

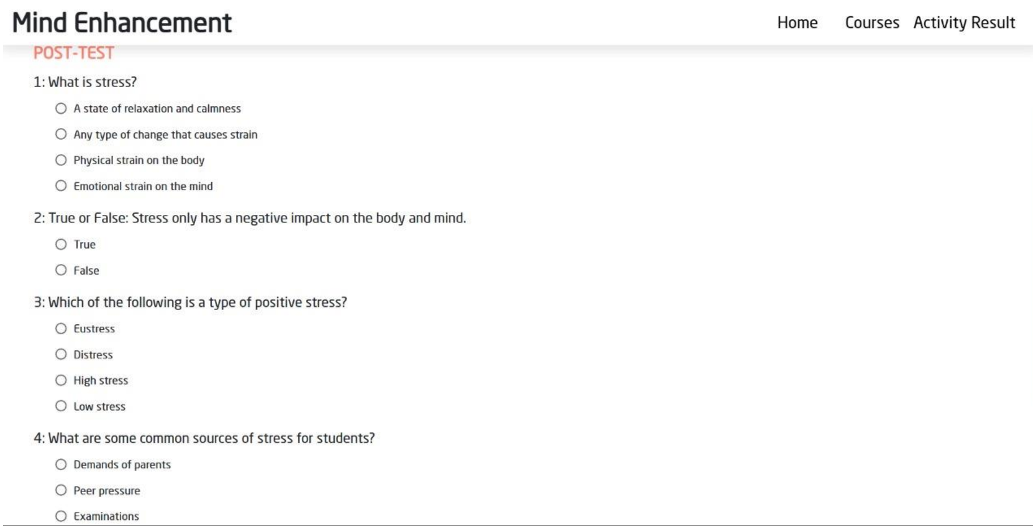

Fig.6.4 Post-test

In the above Fig. 6.4 The post-test, identical to the pre-test, allows users to reassess their stress levels after completing the activity. By comparing the results, the system can determine the impact of the activity on the user's mental health.

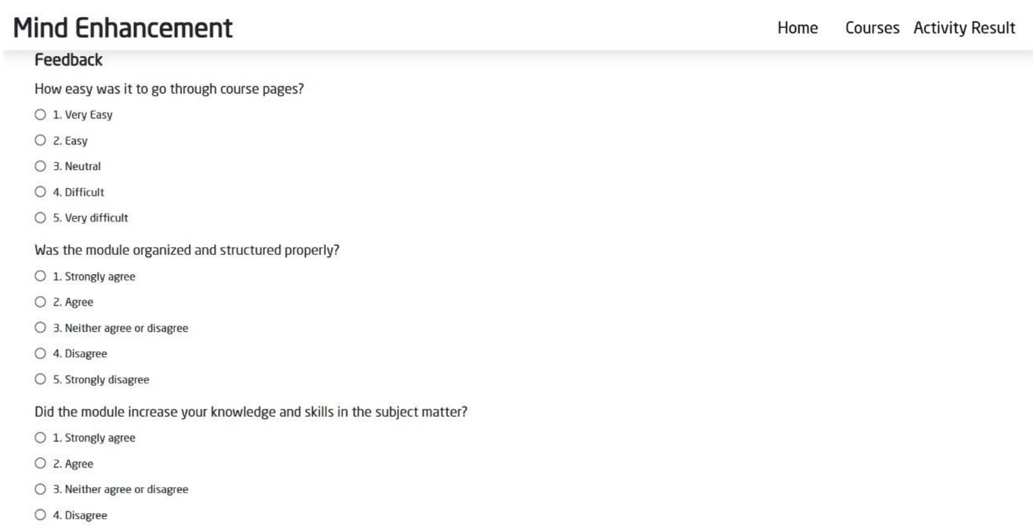

Fig.6.5 Feedback

In the above Fig.6.6 The feedback section collects users' responses regarding their experience with the pre-test and activity It helps assess the effectiveness of the intervention and provides insights for future improvements.

```
How easy was it to go through course pages? : "Very Easy"
Was the module organized and structured properly? : "Strongly agree"
Did the module increased your knowledge and skills in the subject matter? : "Neither agree or disagree"
How helpful did you find the module? : "Helpful"
Rate your overall experience of the Module : "Very Good "
```

```
How easy was it to go through course pages? : "Easy"
Was the module organized and structured properly? : "Strongly agree"
Did the module increased your knowledge and skills in the subject matter? : "Strongly agree"
How helpful did you find the module? : "Very helpful"
Rate your overall experience of the Module : "Very Good "
```

```
How easy was it to go through course pages? : "Very Easy"
Was the module organized and structured properly? : "Agree"
Did the module increased your knowledge and skills in the subject matter? : "Disagree"
How helpful did you find the module? : "Undecided"
Rate your overall experience of the Module : "Acceptable"
```

```
How easy was it to go through course pages? : "Easy"
Was the module organized and structured properly? : "Strongly agree"
Did the module increased your knowledge and skills in the subject matter? : "Strongly agree"
How helpful did you find the module? : "Very helpful"
Rate your overall experience of the Module : "Very Good "
```

```
How easy was it to go through course pages? : "Very Easy"
Was the module organized and structured properly? : "Strongly agree"
Did the module increased your knowledge and skills in the subject matter? : "Agree"
How helpful did you find the module? : "Helpful"
Rate your overall experience of the Module : ""
```

Fig 6.6 Feedback images

In the above Fig.6.6 shows the Feedback from the users on a mental health module.

CHAPTER 7

CONCLUSION

7.1 Conclusion

The "WellMind" project successfully demonstrates the integration of technology and mental health assessment to create a platform that aids in detecting stress levels among individuals. By leveraging a robust tech stack, including React.js for an interactive frontend, Flask for a scalable backend, and MongoDB for efficient data management, the project ensures seamless functionality and user experience. The incorporation of machine learning techniques, particularly the Random Forest Classifier, enhances the accuracy of stress level predictions, providing valuable insights to users.

This project emphasizes the importance of mental health in educational and professional environments, offering a practical solution to identify and address stress-related concerns early. The modular architecture, user-friendly interface, and data-driven approach underline the potential of technology in tackling real-world challenges. Ultimately, "WellMind" is a step towards fostering mental well-being through innovative and accessible tools.

7.2 Future Scope

The "WellMind" project has significant potential for future enhancements and expansions. Some possible directions include:

1. Advanced Machine Learning Models: Incorporating more sophisticated machine learning algorithms like deep learning models can improve the accuracy of stress detection and allow for more nuanced analysis of user responses.

2.Real-Time Monitoring: Expanding the platform to include real-time stress monitoring using wearable devices, such as smartwatches or fitness trackers, to collect physiological data like heart rate and skin temperature.

3. Multi-Language Support: Adding multi-language support to make the platform accessible to a diverse user base, particularly in regions where mental health resources are limited.

4. Personalized Interventions: Developing personalized stress management strategies based on user history and preferences, including relaxation exercises, mindfulness activities, or cognitive-behavioral therapy techniques. 5. Integration with Other Platforms: Integrating the system with existing educational or corporate wellness programs to provide a comprehensive mental health solution for students, employees, and organizations.

6. Data Privacy and Security: Enhancing data privacy and security measures to build trust among users and comply with global data protection regulations like GDPR or HIPAA.

By focusing on these areas, "WellMind" can evolve into a more advanced and holistic platform, offering valuable support for mental health management and promoting well-being across various sectors.

REFERENCES

[1] C. Arihta, D. Fujiono, Veronica, J. V. Moniaga and I. Dwi Rianto, *"A Review of Gamification Related to Mental Health Treatment,"* 2022 3rd International Conference on Electrical Engineering and Informatics (ICon EEI), Pekanbaru, Indonesia, 2022, pp. 172-176, doi: 10.1109/IConEEI55709.2022.9972282

[2] Y. -P. Chang et al., *"Poster: Preliminary Outcomes of a Culturally Tailored Mindfulness Mobile App for Mental Health within Underserved African American Communities During COVID-19,"* 2022 IEEE/ACM Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE), Arlington, VA, USA, 2022, pp. 176-177.

[3] R. Majethia, V. P. Sharma and R. Dwaraghanath, *"Mental Health Indices as Biomarkers for Assistive Mental Healthcare in University Students,"* 2022 10th International Conference on Affective Computing and Intelligent Interaction (ACII), Nara, Japan, 2022, pp. 1-8, doi: 10.1109/ACII55700.2022.9953847.

[4] F. Liu and Y. Mai, *"Application of Computer Technology in College Students' Mental Health Education,"* 2022 3rd International Conference on Education, Knowledge and Information Management (ICEKIM), Harbin, China, 2022, pp. 1019-1024, doi: .1109/ICEKIM55072.2022.00222.

[5] D. Feil-Seifer, A. Kirn, K. L. Stienhorst and M. C. Parker, *"WIP: Faculty Perceptions of Graduate Student Mental Health: A Productivity Framing,"* 2023 IEEE Frontiers in Education Conference (FIE), College Station, TX, USA, 2023, pp. 01-05, doi: 10.1109/FIE58773.2023.10343185.

[6] W. Haibin, C. Lin, W. Liyan and S. Yongping, *"The Visualization Analysis of Network Mental Health Education Literature in China Based on Citespace,"* 2020 5th International Conference on Information Science, Computer Technology and Transportation (ISCTT), Shenyang, China, 2020, pp. 126-129, doi: 10.1109/ISCTT51595.2020.00030.

[7] S. Li and Y. Lai, "Research on Measures of Deep Integration of Information Technology and Mental Health Education," 2021 International Conference on

Education, Information Management and Service Science (EIMSS), Xi'an, China, 2021, pp. 71-75, doi: 10.1109/EIMSS53851.2021.00024.

[8] Y. Shen, *"Analysis on Mental Health Education Strategy of Higher Vocational College Students Based on Information Technology,"* 2021 2nd International Conference on Education,Knowledge and Information Management (ICEKIM), Xiamen, China, 2021, pp. 666-669, doi: 10.1109/ICEKIM52309.2021.00151.

[9] Y. Zhang, *"Research on Innovative Ways of Mental Health Education for Vocational Students under the Internet Background,"* 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE), Harbin, China, 2020, pp. 1696-1700, doi: 10.1109/ICMCCE51767.2020.00372.

[10] R. Aggarwal and A. Goyal, *"Anxiety and Depression Detection using Machine Learning,"* 2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON), Faridabad, India, 2022, pp. 141-149, doi: 10.1109/COM-IT-CON54601.2022.9850532.