List of Tables

List of Figures

Contents

1 INTRODUCTION

Mental health has become an increasingly important topic in the global dialogue, as mental health conditions have been rising across various age groups and demographics. In an era marked by rapid technological advancements, societal shifts, and heightened stress levels, individuals face an ever-growing need for reliable mental health support. Yet, a significant number of people still struggle with seeking help due to the stigma surrounding mental health, lack of accessible resources, or a lack of awareness about available support systems.This project aims to bridge this gap by providing a digital mental health assistant to assess, guide, and provide relevant resources for users.

The Mental Health Support System is an innovative Chatbot Integrated web application designed to provide a comprehensive, effective mental health solution for users. This system offers a unique approach by integrating a chatbot powered by OpenAI's advanced language model, which is capable of offering personalized responses and guidance to users. With features include a mental health quiz for initial assessments, a chatbot that provides tailored support, self-care activity suggestions, reminders, a progress tracker, and recommendations for professional help if necessary , the system aims to empower users by giving them the tools to assess and manage their mental well-being effectively. Additionally, the platform allows users to access a library of helpful exercises and YouTube videos to improve their mental well-being.

The backend of the system is developed using Flask, with a user-friendly frontend interface. The chatbot's integration utilizes OpenAI's powerful model to provide real-time responses to user queries. The system also supports personalized suggestions based on individual user profiles.

Through this project, the goal is to provide users with a scalable, user-friendly, and accessible platform that can serve as a first step toward improving their mental health. By promoting mental wellness through personalized suggestions, constant engagement, and access to professional help, the Mental Health Support System aims to make mental health care more accessible to a larger audience, thereby contributing to a society that values and prioritizes emotional well-being.

2 SUPPORTING LITERATURE

2.1 Literature Review

Paper 1: Kang, Boyoung, and Munpyo Hong. "Development and Evaluation of a Mental Health Chatbot UsingChatGPT 4.0: Mixed Methods User Experience Study With Korean Users." JMIRMedicalInformatics 13 (2025): e63538.doi: 10.2196/63538

The paper "Development and Evaluation of a Mental Health Chatbot Using ChatGPT 4.0" by Boyoung Kang and Munpyo Hong presents the design and evaluation of HoMemeTown Dr. CareSam, a chatbot aimed at providing mental health support to Korean young adults. The study addresses the increasing prevalence of stress, anxiety, and depression, especially in the post-pandemic era, where traditional mental health services often face issues such as stigma, long wait times, and limited accessibility. To overcome these challenges, the authors developed a culturally adaptive and personalized chatbot using OpenAI's ChatGPT 4.0 API, with features like emotion recognition, gratitude journaling, and AI-based risk detection. The chatbot supports both Korean and English, improving its accessibility.

A pilot study involving 20 Korean participants aged 18–27 was conducted to assess the chatbot's usability and effectiveness. Participants, familiar with other platforms like Woebot and Happify, evaluated Dr. CareSam based on positivity, empathy, active listening, personalization, and professionalism. The chatbot received high ratings in positivity (9.0), empathy (8.7), and active listening (8.0), while scores for personalization and professionalism were slightly lower (7.4 and 7.0 respectively). Compared to Woebot's scripted replies and Happify's structured CBT-based activities, Dr. CareSam offered more dynamic and engaging conversations. However, challenges such as slower response times, occasional language inconsistencies, and lack of deep contextual understanding during longer interactions were noted.

The study highlights the potential of AI chatbots for delivering accessible and engaging mental health support, but it also points out limitations such as the small sample size, limited generalizability, and ethical concerns regarding data privacy and the role of AI in therapy. The authors recommend enhancing natural language processing, improving response clarity, and integrating real-time crisis support features. In conclusion, while Dr. CareSam shows promise as a supplementary tool for mental health care, further improvements and integration with professional services are essential for broader and safer application.

Paper 2: Oghenekaro, Linda Uchenna, and Christopher Obinna Okoro. "Artificial intelligence-basedchatbotforstudent mental health support." Open Access Library Journal 11.5 (2024): 1-14.DOI: 10.4236/oalib.1111511

The paper *"Artificial Intelligence-Based Chatbot for Student Mental Health Support"* by Linda Uchenna Oghenekaro and Christopher Obinna Okoro presents an AI-driven chatbot aimed at supporting students struggling with mental health issues such as stress, anxiety, and depression. The study acknowledges the limitations of traditional counseling services, especially due to the stigma surrounding therapy, which often discourages students from seeking help. To address this gap, the authors propose a chatbot that provides accessible, stigma-free, and personalized mental health assistance through artificial intelligence.

The chatbot utilizes AI-based Cognitive Behavioral Therapy (CBT) techniques to help students manage emotions, identify negative thought patterns, and access timely self-help strategies. It serves as a scalable and cost-effective alternative to human therapists, offering a conversational interface that enhances emotional well-being while protecting user privacy. The research follows a mixed-methods design, incorporating both quantitative surveys and qualitative feedback to evaluate the system's usability and effectiveness.

Technically, the chatbot was built using ReactJS, Vite, SCSS, and the Dido training framework, with data sourced from Kaggle and GitHub. It incorporates Natural Language Processing (NLP) algorithms for intelligent, real-time response generation. Additional features like sentiment analysis and conversation tracking allow the chatbot to personalize interactions based on students' emotional states.

The study reports increased student engagement and improved mental health outcomes, with users expressing high satisfaction compared to traditional mental health applications. Students appreciated the ease of use and anonymity, which encouraged open emotional expression. The chatbot was particularly effective in delivering CBT-based strategies like cognitive restructuring and guided self-reflection. However, concerns about data confidentiality and occasional inaccuracies in responses were noted.

In comparison to other chatbots like Woebot and Ellie, which rely on scripted dialogues, this chatbot's AI-generated responses made conversations more dynamic and adaptive. The paper also discusses privacy and ethical concerns, emphasizing the importance of safeguarding user data and recommends incorporating human oversight to build trust and credibility in AI-based mental health tools. While the chatbot shows strong potential, the authors highlight limitations such as ethical concerns, user skepticism, and the inability to manage high-risk cases requiring professional intervention.

In conclusion, the study underscores the transformative potential of AI in student mental health support. It emphasizes the importance of future research focused on integrating such chatbot systems within university mental health services and developing hybrid models that combine AI and human support for more comprehensive and reliable mental health care.

Paper 3: Kang, Boyoung, and Munpyo Hong. "Development and Evaluation of a Mental Health Chatbot UsingChatGPT 4.0: Mixed Methods User Experience Study With Korean Users." JMIRMedicalInformatics 13 (2025): e63538.DOI 10.3389/fdgth.2023.1034724

The paper titled *"Supporting Mental Health Self-Care Discovery Through a Chatbot"* by Joonas Moilanen, Niels van Berkel, Aku Visuri, Ujwal Gadiraju, Willem van der Maden, and Simo Hosio explores the use of conversational agents in mental health self-care discovery. Recognizing that mental health self-care is essential for well-being, the study investigates how AI-driven chatbots can assist users in discovering and adopting self-care strategies. The primary objective is to analyze user trust, system security perception, and the impact of conversational interactions on mental health support.

The study employed a between-subjects experimental design, involving 80 higher education students. One group used a standalone web-based Decision Support System (DSS) to explore self-care techniques, while the other engaged with an AI-powered chatbot. The chatbot guided users through self-care recommendations, tailoring suggestions based on user responses. Trust in the system was evaluated using the Trust in Automation Scale, measuring system integrity, reliability, security perception, and user confidence.

Results indicate higher engagement and interaction rates among chatbot users, though they reported lower trust in system security and integrity compared to DSS users. Privacy concerns and uncertainties about AI-driven responses contributed to this skepticism. While the chatbot increased accessibility and engagement, users found the DSS's structured approach more predictable and reliable. Some users appreciated the chatbot's conversational flow, while others expressed discomfort with its lack of transparency regarding response generation and data usage.

Limitations include perceived security concerns, ethical issues related to data privacy, and the chatbot's lack of real-time professional oversight. Future research should focus on enhancing chatbot transparency, integrating human oversight, and refining AI-driven self-care personalization. Improved security measures and ethical AI practices are necessary for chatbot-assisted mental health self-care discovery.

In conclusion, the chatbot-based approach demonstrated higher engagement and accessibility, but trust-related concerns remain a barrier to adoption. The study contributes to digital mental health interventions, emphasizing the need for trust-building mechanisms and ethical AI practices in chatbot-assisted self-care. Refining chatbot design and improving user confidence will be critical in leveraging conversational agents for mental health support and self-care empowerment.

2.2 Literature Review Summary

TITLE	YEAR	JOURNAL NAME/PUBLISHER	SUMMARY
Development and Evaluation of a Mental Health Chatbot Using ChatGPT 4.0	2025	JMIR MEDICAL INFORMATICS / ©2025	This study explores the creation of a chatbot-based intervention for mental health support among Korean young adults. Built using OpenAI's ChatGPT 4.0, it features emotion recognition, gratitude journaling, and risk detection. Evaluation shows high user satisfaction but highlights challenges in personalization, response clarity, and ethical concerns.
Artificial Intelligence-Based Chatbot for Student Mental Health Support	2024	Open Access Library Inc. / ©2024	This research focuses on an AI-driven chatbot designed to help students manage stress, anxiety, and depression. It utilizes AI-based CBT techniques for emotional support and self-help strategies. Results indicate improved student engagement, though privacy concerns and ethical considerations regarding AI-based therapy remain key challenges.
Supporting Mental Health Self-Care Discovery Through a Chatbot	2023	Frontiers in Digital Health / ©2023	The paper examines how conversational agents can assist users in discovering self-care strategies for mental health. A comparative study with a traditional DSS found that chatbots increased engagement but raised trust concerns regarding system security and AI transparency. Recommendations include improving chatbot reliability and ethical AI practices.

Table 2.1: Literature Summary Table

2.3 Findings and Proposals

From the literature review of the three research papers, several key findings have emerged, providing valuable insights into the development of a chatbot-integrated mental health support system.

The first paper emphasizes the role of AI-driven chatbots in providing accessible mental health support, focusing on emotion recognition, risk detection, and user engagement. The study highlights that AI chatbots, when properly designed, can outperform traditional therapy apps in engagement and responsiveness, but still face challenges in real-time intervention and response accuracy.

The second paper builds upon this by exploring the integration of Cognitive Behavioral Therapy (CBT) in AI chatbots, aiming to provide structured, evidence-based interventions to students struggling with stress and anxiety. The findings indicate that AI-driven CBT strategies can effectively improve student well-being, but issues such as user trust, privacy concerns, and AI adaptability need to be addressed for widespread adoption. The study suggests that better AI personalization and enhanced security protocols will be essential to increase trust and usability.

The third paper shifts focus towards mental health self-care discovery, investigating how chatbots can guide users in identifying personalized self-care techniques. The study shows that while chatbots enhance engagement in self-care practices, trust issues related to AI decision-making remain a major barrier. Users preferred the interactivity of chatbots but expressed concerns over privacy and response transparency.

Among the three research papers reviewed, the second paper aligns most closely with my project, as it explores the integration of Cognitive Behavioral Therapy (CBT) in AI chatbots to provide structured, evidence-based interventions for mental health support. My project builds upon this by implementing AI-driven chatbot interactions to assess mental health conditions, provide self-care recommendations, and guide users toward professional intervention if needed. Additionally, key concerns highlighted in the reference papers, such as user trust, privacy, and AI adaptability, are directly addressed in my project through enhanced security measures, AI personalization, and transparent chatbot communication. By combining insights from all three papers, my project aims to develop a holistic mental health support system that not only engages users effectively but also ensures ethical AI practices and real-time crisis detection.

3 SYSTEM ANALYSIS

3.1 Module Description

User Management Module

This module is responsible for handling user authentication, registration, and profile management to ensure secure access to the platform. Users can create accounts, log in securely, and manage their profiles for a personalized experience. It includes password recovery and management features to help users regain access when needed. The module implements authentication mechanisms to protect user data and maintain confidentiality. By offering a user-friendly interface, it enhances accessibility while ensuring security through proper authentication protocols.

Mental Health Assessment Module

This module conducts an initial mental health assessment using a structured quiz, enabling users to evaluate their mental well-being. The quiz comprises customizable questions designed to assess different mental health aspects like stress, anxiety, and depression. Based on real-time analysis of user responses, the module categorizes results and provides meaningful insights. These results help in guiding the chatbot to offer personalized support tailored to the user's mental state. This assessment acts as a foundation for recommending self-care activities, tracking mental health progress, and determining if professional help is needed.

Chatbot Support Module

The chatbot support module integrates OpenAI's API to provide an AI-powered virtual assistant for mental health support. It engages users in real-time conversations, offering personalized assistance based on their responses and mental health status. The chatbot suggests self-care activities, coping strategies, and additional resources, making interactions more helpful and tailored. It continuously adapts based on user progress, ensuring consistent support. By fostering meaningful engagement, this module enhances accessibility to mental health support while maintaining user privacy and confidentiality.

Self-Care and Resource Module

This module provides a collection of self-care activities, exercises, and helpful resources, such as YouTube videos, to support users in improving their mental well-being. It suggests activities based on the user's emotional trends and assessment results, ensuring that recommendations are relevant and effective. Users can explore different self-care techniques, including relaxation exercises, mindfulness practices, and physical activities. The module enhances engagement by allowing users to track their self-care journey. By offering a variety of resources, it empowers users to develop healthy coping strategies and maintain emotional balance.

Mental Health Tracking and Reminder Module

This module helps users monitor their mental health progress and maintain consistency with self-care activities. It provides graphical representations of emotional changes over time, enabling users to visualize their progress. Users can set up personalized schedules for self-care routines and receive timely reminders for important tasks. Additionally, it sends notifications for upcoming appointments and significant mental health events. By incorporating tracking and reminders, the module ensures that users stay engaged in their mental well-being journey. This structured approach promotes consistency and encourages users to follow their self-care plans effectively.

Therapist Referral Module

The therapist referral module connects users with professional mental health support when the system detects a need for specialized care. It recommends therapists based on user data, ensuring that suggestions are tailored to individual requirements. The module facilitates secure communication channels, allowing users to request referrals and receive professional guidance. Users also have the option to directly contact therapists for further assistance. By integrating this referral system, the platform bridges the gap between self-care and professional help, ensuring users receive the right level of support when necessary.

3.2 Business Rules

- Users must register and authenticate themselves before accessing any core functionalities, including quizzes, chatbot support, self-care recommendations, and therapist consultations.
- The system enforces role-based access control, ensuring that Patients, Caretakers, Therapists, and Administrators have permissions specific to their roles.
- Only verified mental health professionals can access and manage patient consultations, respond to therapy requests, and provide recommendations.
- All user data, including quiz responses, therapy records, and self-care preferences, must be encrypted and stored securely in MongoDB to prevent unauthorized access.
- Self-care recommendations, including activities, exercises, videos, articles, and books, must be dynamically displayed.
- Notifications must be sent for reminders, therapist appointment updates, and missed self-care activities to encourage consistent mental health management.
- System administrators have exclusive access to modify quiz questions, manage user accounts, verify therapist registrations, and update platform functionalities.
- All therapist registrations must undergo an approval process by administrators before granting access to provide consultations.
- Security measures, including authentication, role-based access control, and data encryption, must be enforced to protect sensitive mental health information.
- The platform should ensure seamless integration between modules, providing a smooth and engaging user experience across all features.

3.3 Feasibility Analysis

The mental health support system project is highly feasible from a technical, economic, and operational perspective. With the increasing awareness and need for mental health solutions, the system provides a timely solution for users to access support and guidance. The platform is designed to scale effectively accommodate an expanding user base and data. Below is a detailed analysis of the technical, economic, and operational feasibility of the project.

3.3.1 Technical Feasibility

The system is technically feasible as it leverages well-established and reliable technologies such as Flask, Python, MongoDB, and OpenAI API. Flask, a lightweight yet powerful web framework, ensures efficient request handling, secure authentication, and seamless interaction between different modules. MongoDB, a NoSQL database, provides a flexible and scalable solution for storing user data, quiz results, chatbot interactions, self-care activities, and therapist appointments. The integration of OpenAI's API allows for the development of an AI-powered chatbot that offers personalized mental health support by analyzing user queries and guiding them with appropriate self-care suggestions. Additionally, the self-care module effectively stores activities, reminders, and user engagement history, ensuring a comprehensive mental health management experience. With cloud infrastructure or containerized deployment using Docker, the system can efficiently handle an increasing number of users, maintain secure data access, and support real-time updates. Overall, the choice of technology ensures a robust, scalable, and maintainable platform for mental health assistance.

3.3.2 Economical Feasibility

The project is economically feasible as it primarily relies on open-source technologies, significantly reducing development and operational costs. Flask and MongoDB are free and open-source, eliminating the need for expensive software licenses. The application can be hosted on low-cost cloud services such as AWS, Google Cloud, or DigitalOcean, with scalable infrastructure that grows as the user base increases. The chatbot functionality utilizes OpenAI's API, which may incur some costs depending on usage; however, API calls can be optimized to minimize expenses. Additionally, the cost of maintaining the self-care module, tracking user progress, and sending reminders is minimal, as MongoDB efficiently stores these details without requiring high-performance computing. Given the growing demand for mental health solutions, the platform holds potential for monetization through premium features, therapist consultations, or subscription-based models, ensuring long-term sustainability and a high return on investment.

3.3.3 Operational Feasibility

The system is highly operationally feasible due to its intuitive user interface and well-structured workflow. The platform ensures a seamless experience by guiding users through mental health assessments, chatbot interactions, self-care suggestions, and therapist recommendations in a structured manner. The user dashboard provides real-time tracking of self-care activities, quiz results, and reminders, enabling users to monitor their mental well-being effectively. The chatbot is designed for easy interaction, allowing users to seek guidance through natural conversations, ensuring accessibility for individuals with varying levels of technical expertise. Furthermore, the therapist referral module streamlines the process of connecting users with professionals, making the platform a comprehensive mental health support system. Continuous performance monitoring, security updates, and user feedback mechanisms will ensure that the system remains reliable and adapts to future enhancements, supporting long-term operational success.

3.4 System Environment

3.4.1 Software Environment

Various software tools and frameworks were utilized in the development of this application to ensure its functionality and performance. By leveraging these software tools and technologies, the development team aimed to create a robust, scalable, and user-friendly application that meets the needs and expectations of its users. The primary technologies employed include:

Python: Python is a high-level programming language that lets developers work quickly and integrate systems more efficiently. This model is developed using many Python libraries and packages such as:

- **Flask:** Flask is a micro web framework written in Python. It provides functionalities like URL routing, form handling, template rendering using Jinja2, and REST API support. It is lightweight, easy to scale, and ideal for building modular applications like the Mental Health Support System.
- **PyMongo:** PyMongo is the Python driver for MongoDB, enabling seamless communication between Flask and the MongoDB database. It allows operations such as insertion, querying, updating, and deletion of documents, used extensively for handling users, quiz results, therapists, reminders, and activities.
- **APScheduler:** APScheduler is used for scheduling reminders and background jobs. It plays a crucial role in managing reminder notifications for self-care activities, allowing users to stay consistent in their mental wellness routines.

- **Datetime:** The `datetime` module is used to manage and compare reminder times, trigger alerts, and calculate missed activities, enhancing time-based functionalities of the system.

HTML and CSS: HyperText Markup Language is used to design the structure of web pages including login, dashboard, quiz, and suggestions pages. Cascading Style Sheets (CSS) are applied for styling, maintaining a soft and friendly theme of light pink and light blue throughout the application to create a calm visual experience.

JavaScript: JavaScript is used for front-end interactivity such as dynamic dropdowns, real-time validation, interactive UI features for reminders, notifications, and activity status updates without reloading pages.

Visual Studio Code: Visual Studio Code is a lightweight code editor used for writing Python, HTML, CSS, and JavaScript files. It provides integrated terminal support, syntax highlighting, extensions for Flask and MongoDB, and Git integration, enhancing developer productivity.

MongoDB and MongoDB Compass: MongoDB is a NoSQL database used to store all application data such as user profiles, quiz questions, therapist records, appointments, and notifications. MongoDB Compass is the GUI client that allows easy visualization, testing, and manipulation of the database collections.

OpenAI API: The OpenAI API is integrated to build an intelligent chatbot that interacts with users empathetically. Based on quiz results and user queries, it offers mental health advice, activities, and motivational responses, enhancing the support experience.

Git and GitHub: Git is used for version control throughout the development lifecycle. GitHub is the remote repository platform for hosting the source code, enabling collaborative development, code management, and issue tracking.

3.4.2 Hardware Environment

Hardware configuration significantly impacts software development, influencing performance and scalability. Choices regarding CPU, RAM, storage, and network infrastructure directly affect application execution and data handling. Moreover, considerations extend to system architecture and compatibility, ensuring efficiency and reliability in modern software solutions.

- **Processor:** Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz 2.40GHz
- **Memory:** 512 GB SSD
- **RAM:** 8 GB
- **System Type:** 64-bit operating system, x64-based processor

- **Operating System:** Windows 11 Home Single Language, Version 22H2
- **OS Build:** 22621.4317
- **Feature Experience Pack:** 1000.22700.1041.0
- **Internet Connectivity:** Stable and high-speed internet used for API integration, database access, and code synchronization.

3.5 Actors and roles

Actors and Their Roles

- **User**
  - Takes the mental health assessment quiz to evaluate their well-being.
  - Interacts with the chatbot for personalized mental health support.
  - Receives self-care suggestions, reminders, and resources to improve their mental health.
  - Tracks their progress and emotional trends over time.
  - Saves self-care activities and sets reminders for them.
  - Views and requests appointments with therapists when needed.
  - Receives notifications for upcoming self-care tasks, therapist appointments, and progress updates.
- **System Admin**
  - Manages user accounts, ensuring secure access to personalized data and services.
  - Maintains and updates the mental health quiz, users,therapists and system functionalities.
  - Monitors the platform's performance and resolves any user-related issues.
  - Ensures data protection and compliance with privacy regulations.
  - Verify therapist requests
- **Mental Health Professional**
  - Offers insights and suggestions based on user data (with consent).
  - Provides professional support or referral options for users needing further assistance.
  - - Accepts or rejects appointment requests from users.
- **User Caretaker**
  - Supports the user in managing their mental health progress.
  - Helps the user set reminders for activities and appointments.
  - Ensures the user engages with self-care suggestions and activities.
  - Helps users to get connected with therapists.

4 SYSTEM DESIGN

4.1 Use case model

A UML diagram is a diagram based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artefacts, or classes, in order to better understand, alter, maintain, or document information about the system.

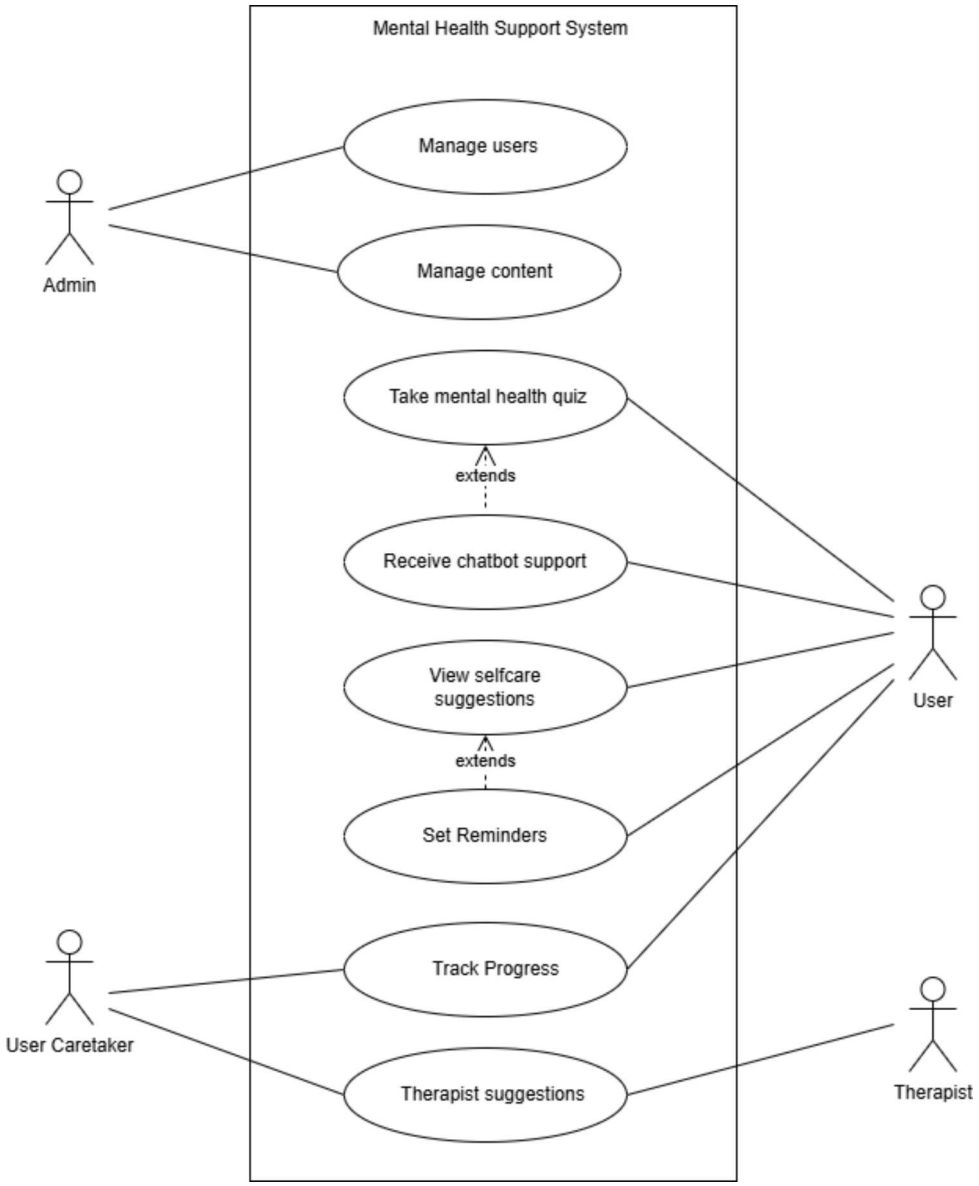

Figure 4.1: Usecase Diagram of the Mental Health Support System

The use case diagram represents the interactions between different actors and the various functionalities within the Mental Health Support System. The primary actors identified in the system are the User, User Caretaker, Admin, and Therapist. The User represents individuals utilizing the platform to assess and manage their mental health. The User Caretaker assists users in maintaining engagement with self-care activities and tracking progress. The Admin is responsible for managing user accounts and content within the system, ensuring its smooth operation. The Therapist provides professional guidance and support to users who require specialized care.

Within the diagram, different use cases are depicted as ovals, representing specific actions that each actor can perform. The functionalities available to users include taking a mental health quiz, receiving chatbot support, viewing self-care suggestions, setting reminders, tracking progress, and accessing therapist suggestions. The Admin is responsible for managing users and content, ensuring system stability and relevance. The User Caretaker plays a supportive role by tracking user progress and assisting in therapist suggestions, while the Therapist provides expert advice when required.

The diagram also showcases extends relationships between use cases, signifying optional or additional functionalities. For instance, the ”Take Mental Health Quiz” use case extends ”Receive Chatbot Support”, indicating that chatbot assistance can be provided based on quiz results. Similarly, ”View Self-Care Suggestions” extends ”Set Reminders”, meaning users can set reminders for suggested activities to ensure consistent engagement.

Additionally, actor-use case associations are represented by lines connecting actors to specific use cases. These associations define the scope of interaction for each participant. For example, the User is linked to multiple functionalities such as the mental health quiz, chatbot support, and self-care suggestions, while the Admin is associated with content and user management. The User Caretaker is connected to progress tracking and therapist suggestions, reinforcing their supportive role.

Overall, this use case diagram provides a comprehensive view of the Mental Health Support System by outlining the interactions between actors and functionalities. It serves as an essential tool for understanding system behavior, ensuring smooth development, and facilitating effective communication among stakeholders.

4.2 Activity Diagram

The Activity Diagram for the Mental Health Support System illustrates the workflow from the start point to the finish point, detailing various decision paths and interactions within the system. It represents how users, caretakers, therapists, and the admin engage with the system’s functionalities.

4.2.1 Mental health assessment and chatbot support

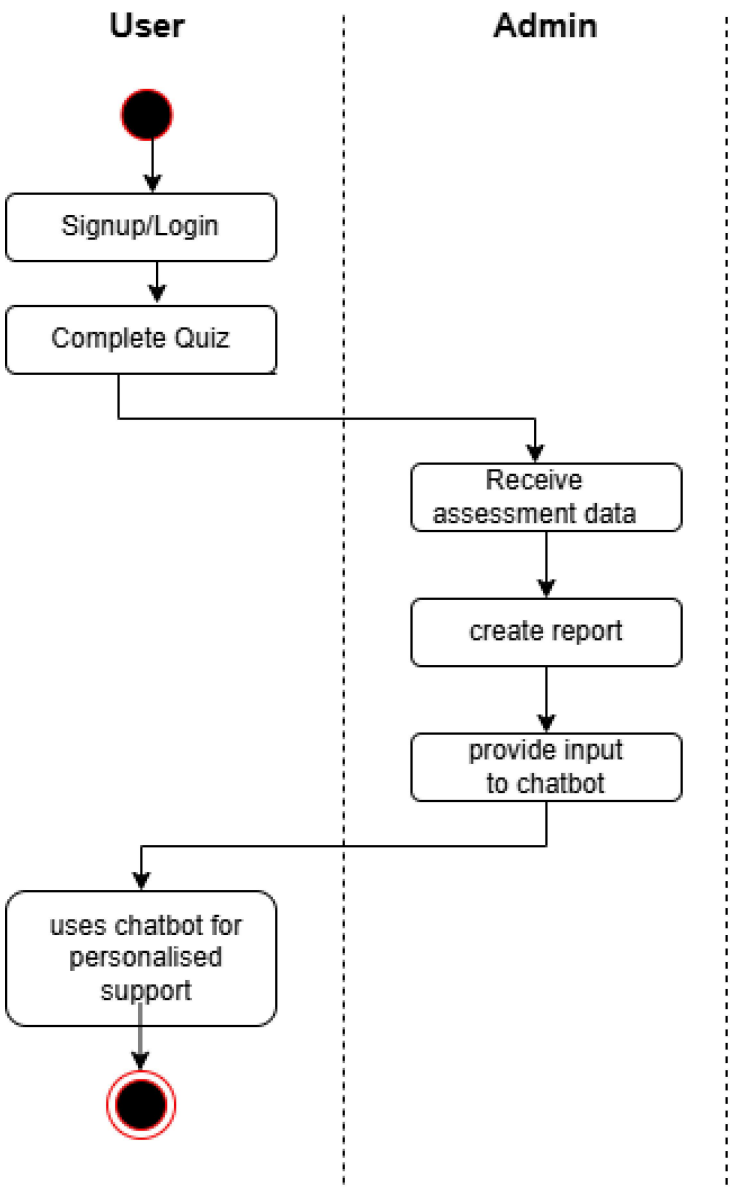

Figure 4.2: Activity Diagram of the Mental Health Assessment and chatbot support

The activity diagram of the Mental Health Support System represents the structured workflow between the User and Admin, ensuring an efficient and personalized mental health assessment and support process. The process begins when a User registers or logs into the system, establishing a secure and authenticated session. This step ensures that only authorized users can access the platform and utilize its mental health assessment features. Once logged in, the user proceeds to complete a mental health assessment quiz, which consists of structured questions designed to evaluate their emotional and psychological well-being. The quiz aims to detect symptoms of mental health conditions such as anxiety, depression, or stress, based on scientifically validated assessment criteria.

Once the user submits the quiz, the system automatically forwards the assessment data to the Admin for processing. The admin (or an automated system) plays a crucial role in handling and analyzing the submitted data. The first step in this process involves receiving the assessment data, ensuring that all responses are correctly recorded and securely stored. The admin then creates a report based on the assessment results, summarizing key findings that indicate the user's mental health condition. This report provides valuable insights into the user's mental state and serves as the foundation for the next step in the process.

After generating the report, the admin processes the collected data and provides the necessary input to the chatbot. The chatbot, integrated into the system, leverages this data to deliver personalized mental health support. This interaction ensures that the user receives tailored guidance, self-care recommendations, and coping strategies relevant to their specific mental health condition. The chatbot acts as an interactive support system, assisting users with advice, exercises, and resources designed to improve their mental well-being.

Finally, the user engages with the chatbot to receive personalized mental health support. The chatbot provides recommendations based on the user's assessment results, offering insights on managing stress, anxiety, or depression effectively. This step ensures that users receive continuous support tailored to their specific mental health needs. The process concludes once the user has interacted with the chatbot, receiving guidance and resources to aid in their mental well-being journey.

This structured workflow effectively integrates mental health assessment, data processing, and AI-powered chatbot support to create a seamless and user-friendly experience. By enabling real-time mental health evaluation and personalized assistance, the system enhances accessibility to mental health resources, fostering a proactive approach to mental well-being management.

4.2.2 Selfcare Management and Progress Tracking

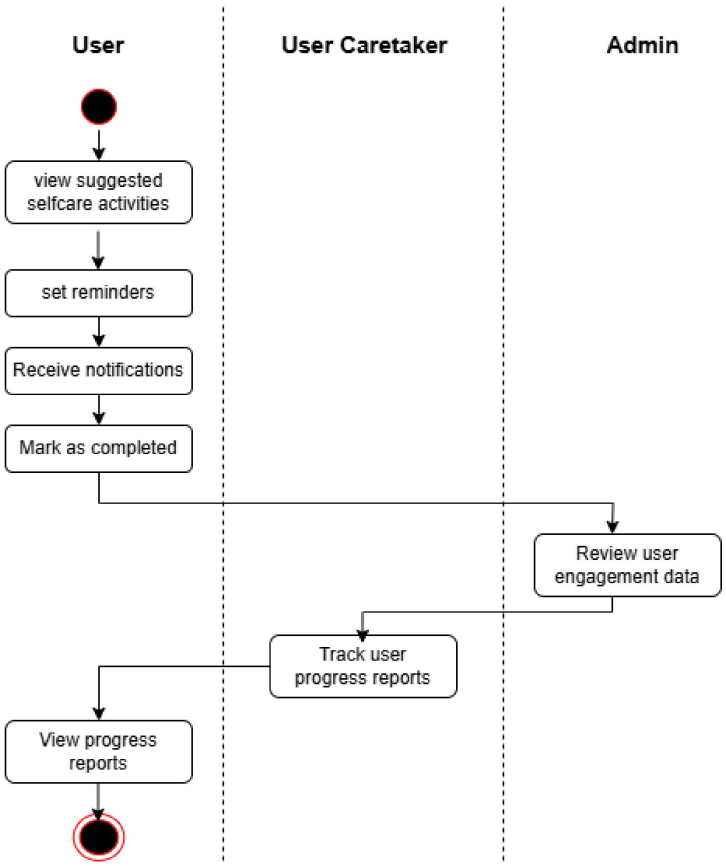

Figure 4.3: Activity Diagram of the Selfcare Management and Progress Tracking

The activity diagram of the self-care module in the mental health support system outlines the structured workflow involving three key participants: user, user caretaker, and admin. This process ensures that users receive personalized self-care recommendations, track their progress, and receive support from caretakers and administrators for a holistic mental well-being experience.

The process begins with the user accessing the system and viewing suggested self-care activities tailored to their mental health condition. These activities may include meditation, breathing exercises, journaling, physical activities, or guided therapy sessions, all designed to help users manage stress, anxiety, and depression effectively. After exploring the self-care suggestions, the user proceeds to set reminders for selected activities, ensuring consistency and adherence to their self-care routine. The system then sends notifications to the user at scheduled times, reminding them to engage in the planned activities.

Once the user completes an activity, they can mark it as completed, helping them keep track of their engagement with self-care practices. This action also enables the system to record user participation and track adherence. The next step involves the user caretaker, who has access to the user's self-care engagement history and can track user progress reports. This feature allows

caretakers to monitor improvements in the user's mental well-being and provide additional support if needed.

Simultaneously, the admin plays a crucial role in overseeing user engagement trends. The system enables the admin to review user engagement data, analyzing the effectiveness of self-care activities and identifying patterns in user behavior. This data-driven approach allows for continuous improvements in the self-care module by refining recommendations and enhancing user experience.

Finally, the user can view their progress reports, allowing them to assess their consistency in following self-care practices and track improvements in their mental well-being. The progress report serves as a motivation tool, encouraging users to maintain their self-care routine and make necessary adjustments based on insights from the caretaker and admin.

This structured workflow ensures a seamless and interactive approach to mental health management by integrating self-care activities, user engagement tracking, caretaker monitoring, and administrative insights. By providing personalized support and structured progress tracking, the system enhances users' commitment to self-care while fostering a collaborative environment for improved mental well-being.

4.2.3 Therapist suggestions

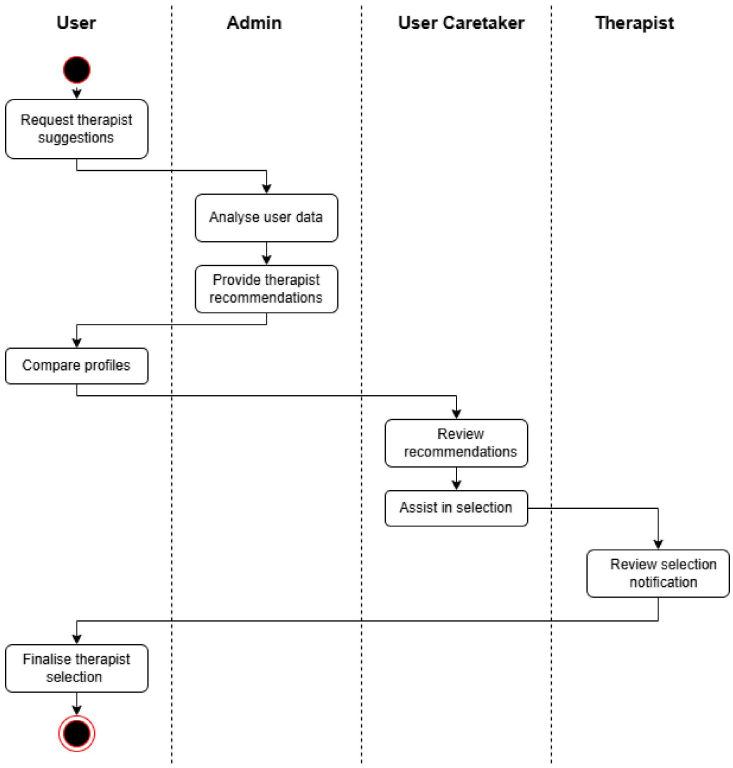

Figure 4.4: Activity Diagram of the Therapist suggestions

The activity diagram illustrates the therapist selection process in the mental health support system, involving four key participants: user, admin, user caretaker, and therapist. This structured workflow ensures that users receive personalized therapist recommendations based on their mental health needs and can make informed decisions with assistance from caretakers and administrative insights.

The process begins with the user requesting therapist suggestions within the system. This request triggers an analysis by the admin, who reviews the user's data, including mental health assessment results, engagement history, and any relevant details that can help determine the most suitable therapists. Based on this analysis, the admin provides a list of recommended therapists, ensuring that the options align with the user's needs and preferences.

Once the recommendations are available, the user proceeds to compare therapist profiles. This step allows the user to assess various factors such as therapist expertise, specialization, availability, and user reviews. During this stage, the user caretaker plays a supportive role by reviewing the therapist recommendations and assisting the user in making an informed selection. If the user needs additional guidance, the caretaker can provide insights to help choose the best therapist based on the available options.

After evaluating the profiles and receiving assistance if needed, the user finalizes the therapist selection. Once a therapist is selected, the system sends a selection notification to the respective therapist. The therapist then reviews the notification, confirming their availability and readiness to proceed with the session or consultation.

This structured approach ensures that users receive tailored therapist recommendations based on data-driven insights, enabling them to make informed decisions with the help of caretakers. The seamless coordination between the user, admin, user caretaker, and therapist enhances the accessibility and effectiveness of mental health support, ensuring that users connect with the right professionals for their well-being.

4.3 Sequence Diagram

A sequence diagram is a visual representation that illustrates the chronological order of interactions between different components within a system. It provides a clear understanding of the process flow, showcasing how various entities communicate with each other over time. By mapping out the sequence of messages, requests, and responses exchanged between system components, a sequence diagram helps in analyzing system behavior, identifying dependencies, and ensuring smooth coordination between different actors.

4.3.1 Mental Health assessment and chatbot support

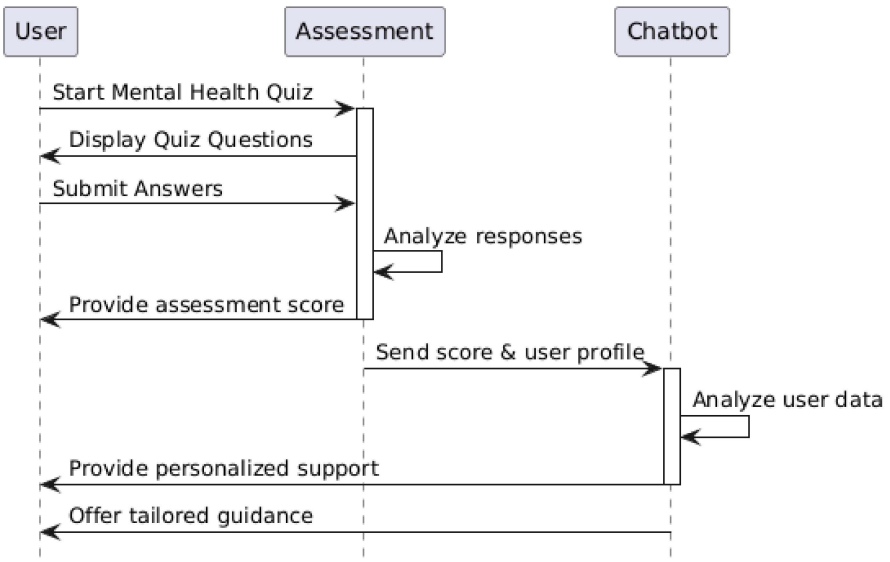

Figure 4.5: sequence diagram for mental health assessment and chatbot support

The sequence diagram visually represents the interaction between the user, assessment module, and chatbot in a mental health support system. It illustrates the step-by-step process that takes place when a user engages with the system for mental health assessment and receives personalized support. The diagram highlights how data flows between the entities and how they communicate in sequence to achieve the desired outcome.

The process begins with the user initiating a mental health assessment by starting the mental health quiz. Upon this request, the assessment module responds by displaying the quiz questions to the user. The user then proceeds to submit their answers, which are sent back to the assessment module for further processing. The assessment module takes these responses and begins the process of analyzing them to determine the user's mental health condition. Once the analysis is complete, the module provides the assessment score back to the user.

After generating the assessment score, the assessment module sends the user's score along with their profile data to the chatbot for further evaluation. The chatbot then processes this information by analyzing the user's data to understand their mental health condition better. Based on this analysis, the chatbot proceeds to provide personalized support tailored to the user's specific needs. Finally, the chatbot offers tailored guidance, which could include self-care suggestions, coping strategies, or recommendations for professional help, ensuring the user receives the most relevant support based on their assessment results.

This sequence diagram effectively captures the seamless interaction between the user, the assessment system, and the chatbot, ensuring a smooth flow of information and personalized mental health support.

4.3.2 Selfcare management and progress tracking

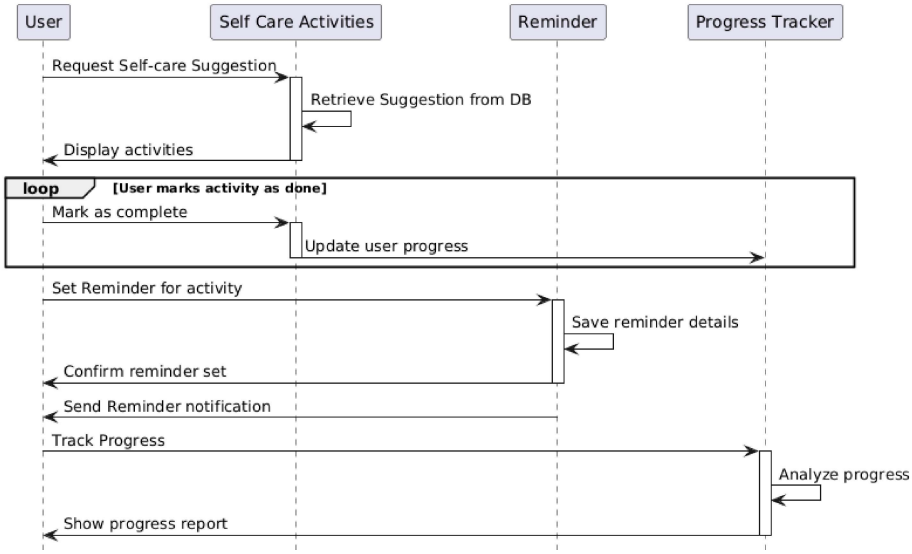

Figure 4.6: sequence diagram for selfcare management and progress tracking

The sequence diagram illustrates the interactions between the user, self-care activities module, reminder system, and progress tracker in a mental health support system. It provides a clear representation of the step-by-step communication flow that enables users to receive self-care suggestions, track their activities, set reminders, and monitor their progress.

The process begins when the user requests a self-care suggestion. The self-care activities module retrieves relevant suggestions from the database and displays them to the user. Once the user selects and engages with an activity, they have the option to mark it as complete. This action triggers a loop that updates the user's progress in the system.

The user can also set a reminder for a specific activity. Upon receiving this request, the reminder system saves the reminder details and confirms that the reminder has been set. When the scheduled time arrives, the system sends a reminder notification to the user, ensuring they stay consistent with their self-care activities.

Additionally, the system enables users to track their progress. The progress tracker module analyzes user engagement and generates a progress report, which is then displayed to the user. This feedback mechanism helps users stay motivated and informed about their self-care journey.

This sequence diagram effectively captures the structured flow of interactions between different system components, ensuring an efficient and user-friendly self-care management process.

4.3.3 Therapist suggestions

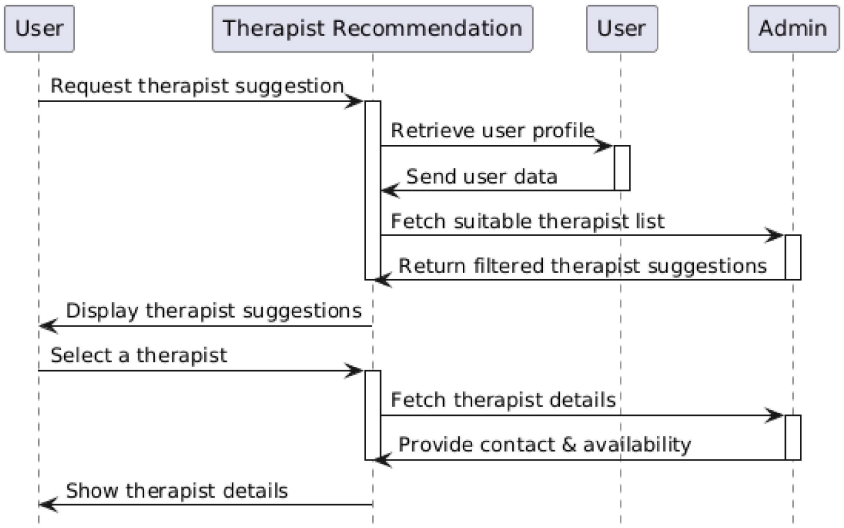

Figure 4.7: sequence diagram for therapist suggestions

The sequence diagram illustrates the process of recommending therapists to users in a mental health support system. It outlines the interactions between the user, therapist recommendation module, and admin, showing how therapist suggestions are retrieved and displayed.

The process begins when the user requests a therapist suggestion. The therapist recommendation module retrieves the user profile and sends the relevant user data to analyze their needs. Based on this information, the system fetches a list of suitable therapists and filters the results according to the user's requirements. The filtered therapist suggestions are then returned and displayed to the user.

After reviewing the therapist suggestions, the user selects a therapist. The system then fetches the selected therapist's details, including their contact information and availability. This data is provided by the admin module and returned to the therapist recommendation module. Finally, the therapist details are displayed to the user, enabling them to proceed with scheduling an appointment.

This sequence diagram effectively represents the structured workflow involved in recommending therapists, ensuring a seamless and efficient process for users seeking professional mental health support.

4.4 List of identified classes, attributes, and their relationships

Classes serve as fundamental building blocks in object-oriented programming, providing a blueprint for creating objects with shared characteristics and behaviors. At its core, a class encapsulates data attributes and methods, defining the structure and behavior of its instances. The primary role of a class is to define the attributes, methods, and applicability of its instances.

4.4.1 Identified Classes

- User
- UserCaretaker
- SystemAdmin
- Assessment
- ProgressTracker
- Reminder
- Chatbot
- Therapist
- SelfcareActivity

4.4.2 Identified Attributes

The **User** class represents the primary participant of the system, responsible for interacting with the platform to assess and manage their mental health. Each user has a unique identifier, personal details such as name, email, age, and gender, and secure authentication credentials. The class provides methods for user registration, login, profile updates, taking assessments, and tracking progress. Users can engage with the chatbot, receive self-care recommendations, and monitor their mental health journey over time. The encapsulated password ensures data security, and interactions with the system facilitate a personalized and seamless user experience.

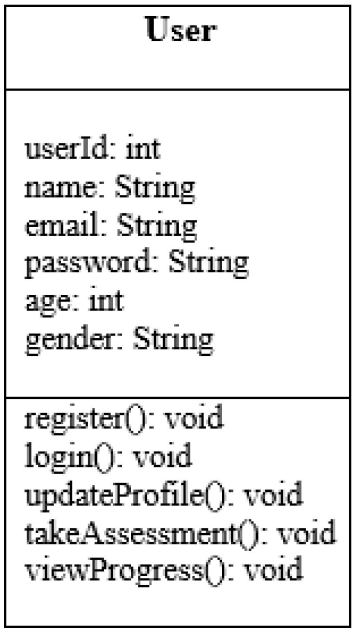

Figure 4.8: User class

The **UserCaretaker** class represents an individual who assists the user in their mental health journey. A caretaker can view the user's progress, set reminders for self-care activities, and ensure engagement with the platform. This class helps in monitoring and supporting the user while maintaining communication through notifications.

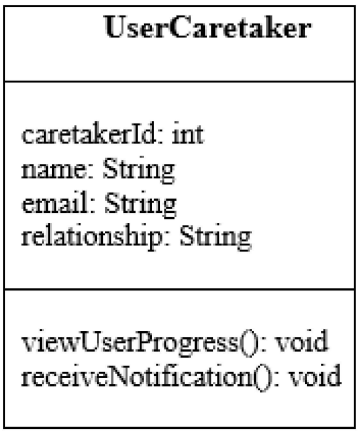

Figure 4.9: UserCaretaker class

The **SystemAdmin** class is responsible for managing the platform's functionality and ensuring a secure environment. Admins handle user accounts, monitor system performance, approve therapist registrations, and oversee the chatbot's responses. They also ensure compliance with privacy policies and address user-related issues.

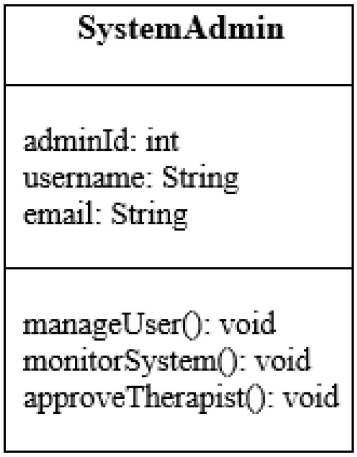

Figure 4.10: SystemAdmin class

The **Therapist** class represents mental health professionals who provide additional support. Therapists can respond to user queries, suggest appropriate interventions, and offer professional guidance based on user assessments. This class facilitates secure communication between users and therapists for better mental health management.

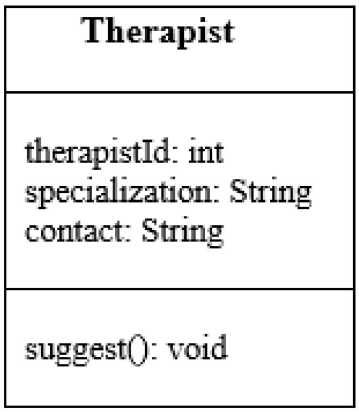

Figure 4.11: Therapist class

The **Assessment** class is responsible for conducting mental health evaluations through quizzes. It records user responses, calculates scores, and determines their mental health category (e.g., Anxiety, Depression, Stress, or Normal). The results guide the chatbot and self-care module in providing tailored recommendations.

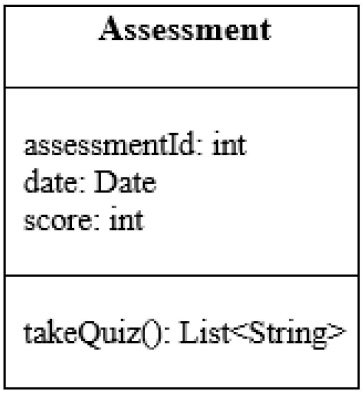

Figure 4.12: Assessment class

The **ProgressTracker** class monitors users' mental health changes over time. It maintains assessment scores, tracks emotional trends, and generates progress reports. This class enables users and caretakers to visualize improvements and identify areas that need more attention.

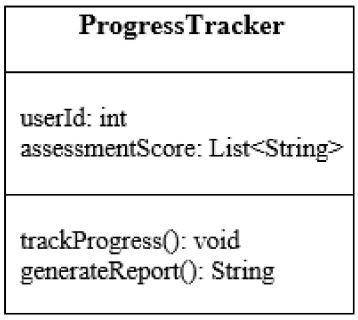

Figure 4.13: ProgressTracker class

The **Reminder** class allows users to schedule and manage self-care activities. Users can set reminders for meditation, exercise, or therapy sessions, ensuring consistency in mental health improvement. Notifications are sent at predefined times to help users stay on track.

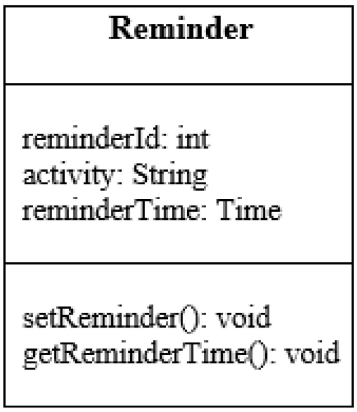

Figure 4.14: Reminder class

The **Chatbot** class integrates AI-based mental health support using OpenAI's API. It engages users in meaningful conversations, provides guidance based on assessments, and offers self-care suggestions. The chatbot ensures continuous user support through an interactive interface.

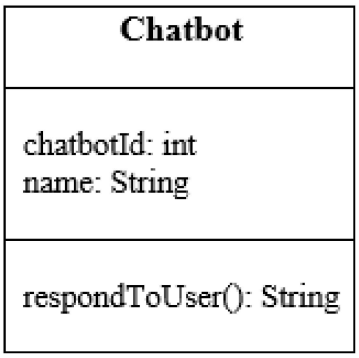

Figure 4.15: Chatbot class

The **SelfcareActivity** class offers users a variety of self-care resources, including activities, exercises, articles, and videos. It provides personalized recommendations based on users' mental health assessments. This class plays a key role in improving emotional well-being by ensuring users engage in beneficial self-care activities.

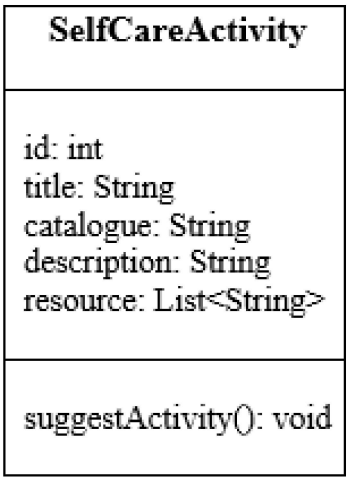

Figure 4.16: SelfcareActivity class

4.4.3 Identified Relationship

Class	Related Class	Association Name	Cardinality
User	UserCaretaker	Assisted by a caretaker	1 - *
User	SystemAdmin	Managed by admin	1 - *
User	Assessment	Takes mental health assessment	1 - *
User	ProgressTracker	Tracks emotional well-being	1 - 1
User	Chatbot	Interacts for support	1 - 1
User	SelfcareActivity	Receives self-care suggestions	1 - *
UserCaretaker	Reminder	Sets reminders for user	1 - *
SystemAdmin	Therapist	Approves therapists	1 - *
SystemAdmin	User	Manages user accounts	1 - *
Therapist	Assessment	Reviews user assessment	1 - *
Assessment	ProgressTracker	Stores assessment scores	1 - 1
Assessment	Chatbot	Generates responses based on results	1 - 1
ProgressTracker	SelfcareActivity	Suggests activities based on progress	1 - *
Reminder	User	Sends reminders for activities	1 - 1
Chatbot	User	Provides mental health support	1 - 1
SelfcareActivity	User	Recommended for mental well-being	1 - *

Table 4.1: Identified Relationships in the System

4.5 Class Diagram

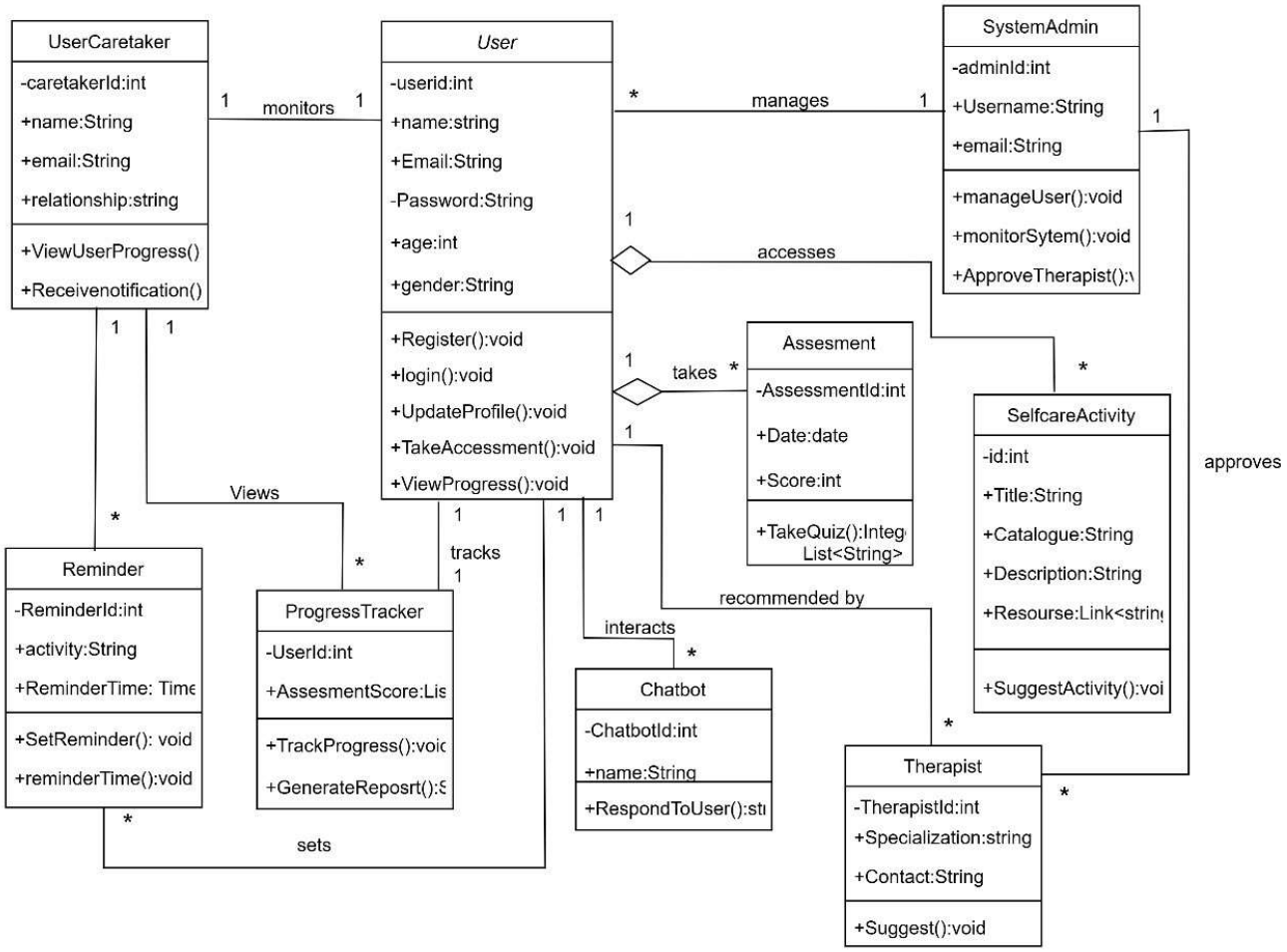

Figure 4.17: Class Diagram of Chatbot Integrated mental health support system

4.6 Database Design

Table: User
Primary Key: user_id

Attribute	Datatype	Constraints	Description
user_id	ObjectId	Primary Key, Unique	Unique identifier for each user
name	String	Not Null	Full name of the user
email	String	Unique, Not Null	User's email (unique)
password	String	Not Null	Encrypted password
age	Number	Not Null, Positive	Age of the user
gender	String	Not Null	Gender of the user
interests	Array	Nullable	List of user interests for analysis

Table 4.2: User Table

Table: Assessment
Primary Key: assessment_id

Attribute	Datatype	Constraints	Description
assessment_id	ObjectId	Primary Key, Unique	Unique identifier for each assessment
user_id	ObjectId	Foreign Key (User)	Reference to User who took the assessment
date	Date	Not Null	Date of assessment
score	Number	Not Null, Positive	Total score of the quiz
category	String	Not Null	Assessment category (Anxiety, Depression, Stress, Normal)

Table 4.3: Assessment Table

Table: ProgressTracker
Primary Key: progress_id

Attribute	Datatype	Constraints	Description
progress_id	ObjectId	Primary Key, Unique	Unique identifier for tracking progress
user_id	ObjectId	Foreign Key (User)	Reference to User
assessment_id	ObjectId	Foreign Key (Assessment)	Reference to Assessment
progress_data	Array	Not Null	List of progress scores over time
generated_report	String	Nullable	Link to generated progress report

Table 4.4: Progress Tracker Table

Table: Reminder
Primary Key: reminder_id

Attribute	Datatype	Constraints	Description
reminder_id	ObjectId	Primary Key, Unique	Unique identifier for each reminder
user_id	ObjectId	Foreign Key (User)	Reference to User
activity	String	Not Null	Activity for which reminder is set
reminder_time	Date	Not Null	Time for the reminder

Table 4.5: Reminder Table

Table: Chatbot
Primary Key: chatbot_id

Attribute	Datatype	Constraints	Description
chatbot_id	ObjectId	Primary Key, Unique	Unique identifier for chatbot interaction
user_id	ObjectId	Foreign Key (User)	Reference to User
message	String	Not Null	User's query to the chatbot
response	String	Not Null	Chatbot's response

Table 4.6: Chatbot Table

Table: SelfcareActivity
Primary Key: activity_id

Attribute	Datatype	Constraints	Description
activity_id	ObjectId	Primary Key, Unique	Unique identifier for each activity
title	String	Not Null	Activity title
description	String	Not Null	Detailed description
resource_link	String	Nullable	External link for more info

Table 4.7: Selfcare Activity Table

Table: Therapist
Primary Key: therapist_id

Attribute	Datatype	Constraints	Description
therapist_id	ObjectId	Primary Key, Unique	Unique identifier for therapist
name	String	Not Null	Therapist name
specialization	String	Not Null	Area of expertise
contact_info	String	Not Null, Unique	Contact details

Table 4.8: Therapist Table

Table: SystemAdmin
Primary Key: admin_id

Attribute	Datatype	Constraints	Description
admin_id	ObjectId	Primary Key, Unique	Unique identifier for admin
username	String	Not Null, Unique	Admin username
email	String	Not Null, Unique	Admin email
password	String	Not Null	Encrypted password

Table 4.9: System Admin Table

Table: QuizQuestions
Primary Key: question_id

Attribute	Datatype	Constraints	Description
question_id	ObjectId	Primary Key, Unique	Unique identifier for quiz questions
question	String	Not Null	Text of the quiz question
options	Array	Not Null	List of answer options
correct_answer	String	Not Null	Correct answer for the question

Table 4.10: Quiz Questions Table

Table: UserCaretaker
Primary Key: caretaker_id

Attribute	Datatype	Constraints	Description
caretaker_id	ObjectId	Primary Key, Unique	Unique identifier for caretaker
user_id	ObjectId	Foreign Key (User)	Reference to User
name	String	Not Null	Caretaker name
contact_info	String	Not Null	Caretaker contact details

Table 4.11: User Caretaker Table

4.7 UI Design

The User Interface (UI) design of the system is crafted to provide a seamless, intuitive, and efficient user experience. It follows a modern and visually appealing design approach, ensuring ease of navigation, accessibility, and responsiveness across different devices. A well-structured layout, consistent color scheme, and user-friendly components contribute to an engaging and professional interface.

The design accommodates multiple user roles, each with customized functionalities and dashboards tailored to their specific needs. Dynamic forms and interactive elements allow for personalized user interactions, while structured navigation ensures a smooth workflow. The system integrates real-time updates, interactive panels, dropdowns, notifications, and AI-based support to enhance usability and efficiency.

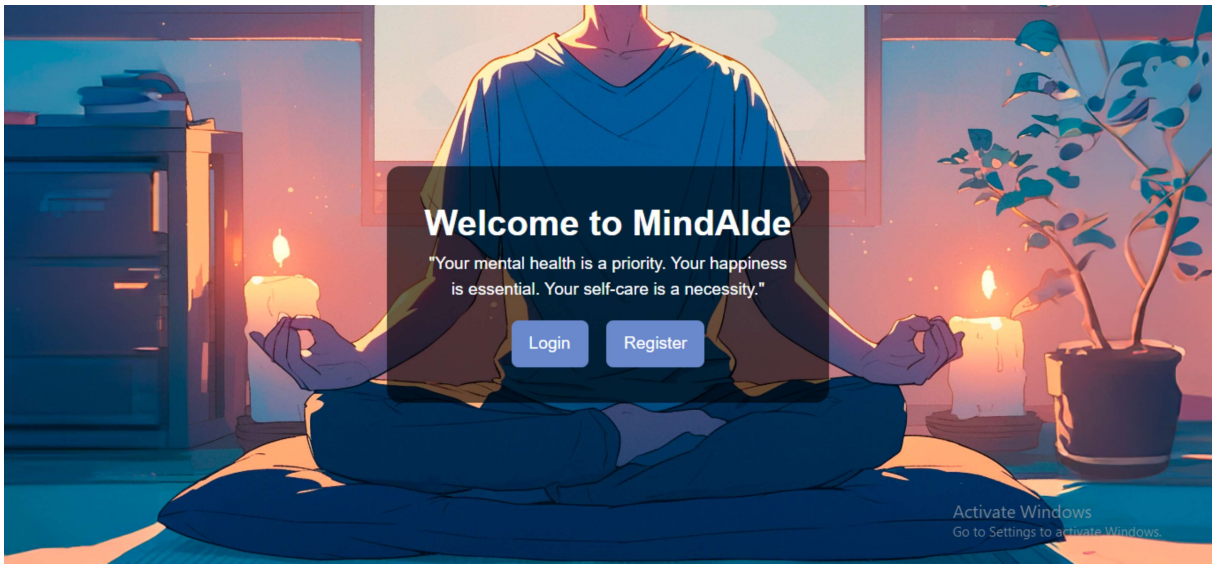

Figure 4.18: home page of mental health support system

The home page serves as the entry point of the mental health support system, offering a clean and welcoming interface. Users are given the option to either log in if they already have an account or register as a new user. The page ensures easy navigation, guiding users based on their needs while maintaining a visually appealing design.

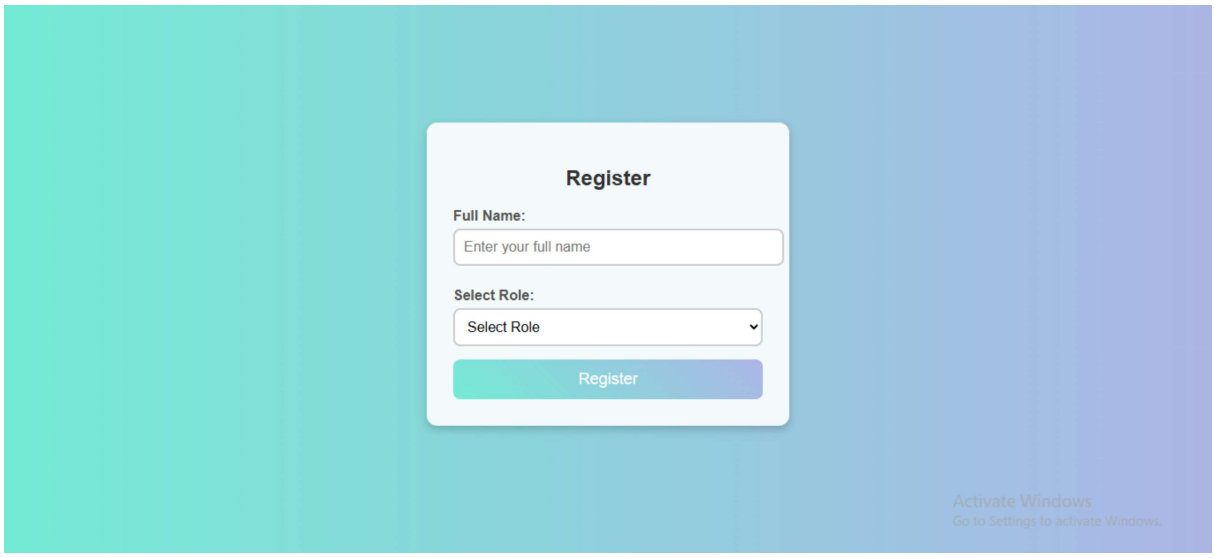

Figure 4.19: Registration page of mental health support system

If a user selects ”Patient,” the registration form updates to include patient-specific fields such as name, age, gender, mental health concerns, and contact information. This dynamic functionality ensures that only relevant details are collected, making the registration process smooth and efficient.

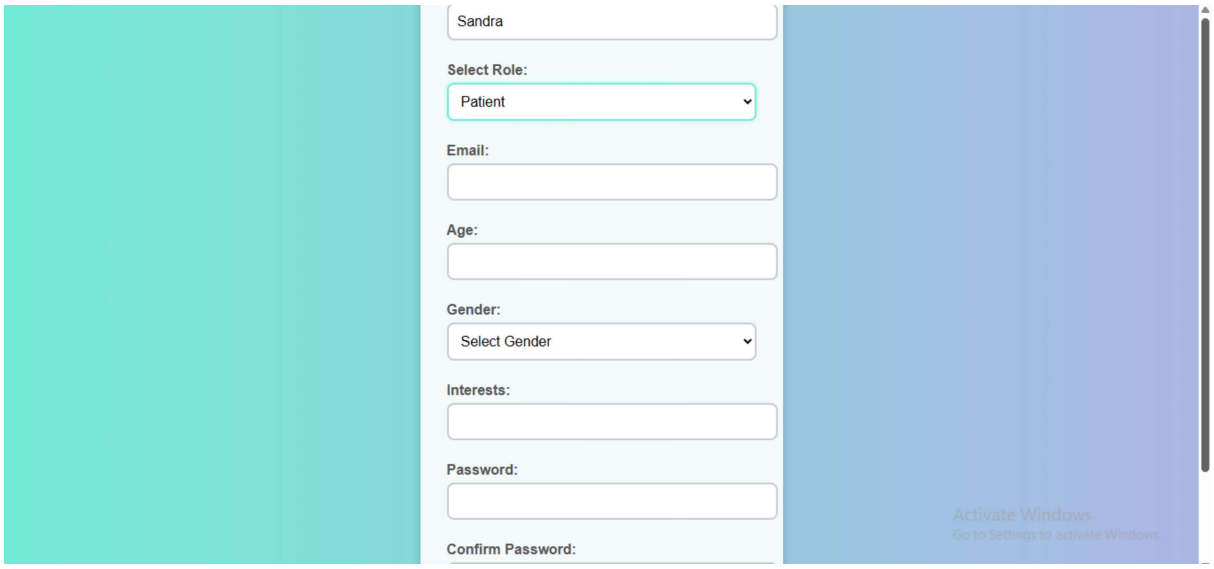

Figure 4.20: Dynamic registration page for patient

If a user selects ”Patient,” the registration form updates to include patient-specific fields such as name, age, gender, mental health concerns, and contact information. This dynamic functionality ensures that only relevant details are collected, making the registration process smooth and efficient.

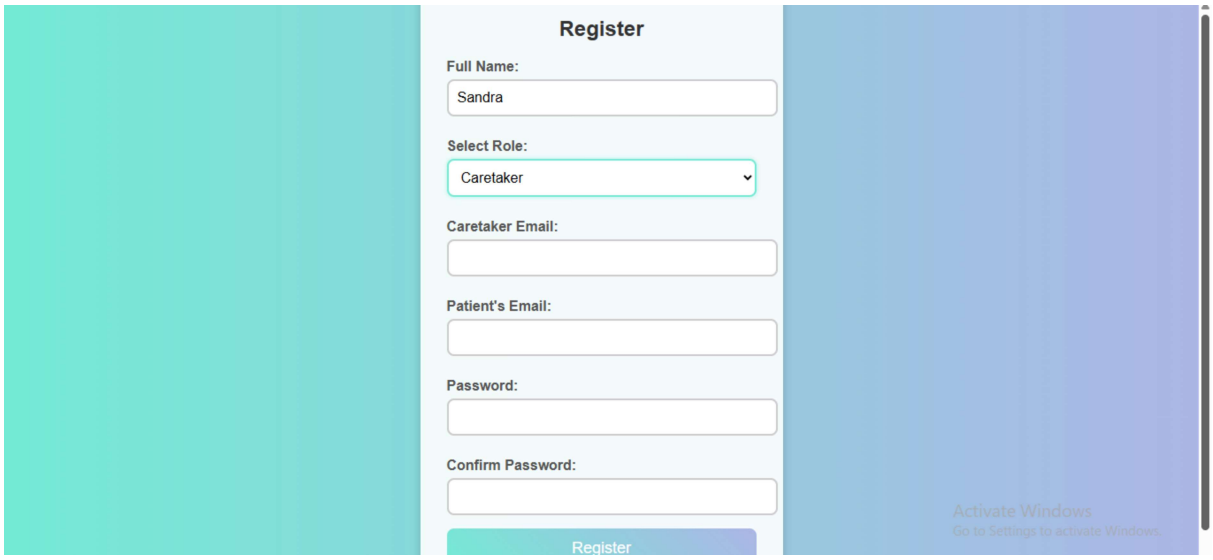

Figure 4.21: Dynamic registration page for caregiver

When ”Caretaker” is selected, the form adapts to collect details about the caretaker’s relationship with the patient, experience in caregiving, and additional support preferences. This customization helps in tailoring the system’s features to assist both the caretaker and the patient effectively.

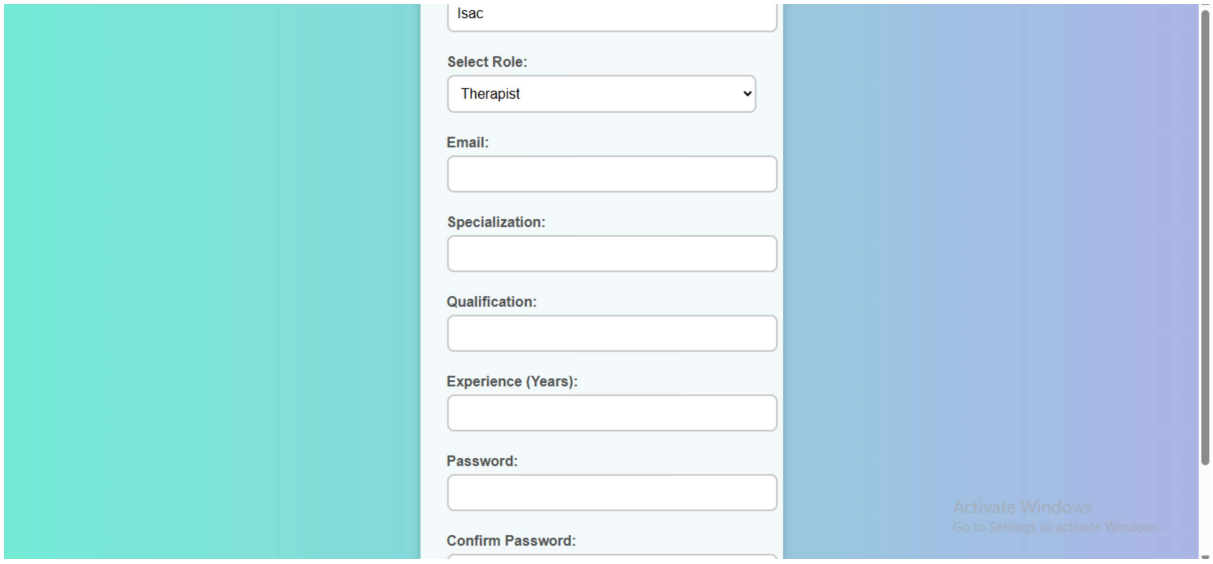

Figure 4.22: Dynamic registration page for therapist

If a user chooses ”Therapist,” the registration form updates to include professional details such as qualifications, specialization areas, years of experience, and availability for consultations. This ensures that only verified therapists are onboarded to the system, maintaining credibility and trustworthiness.

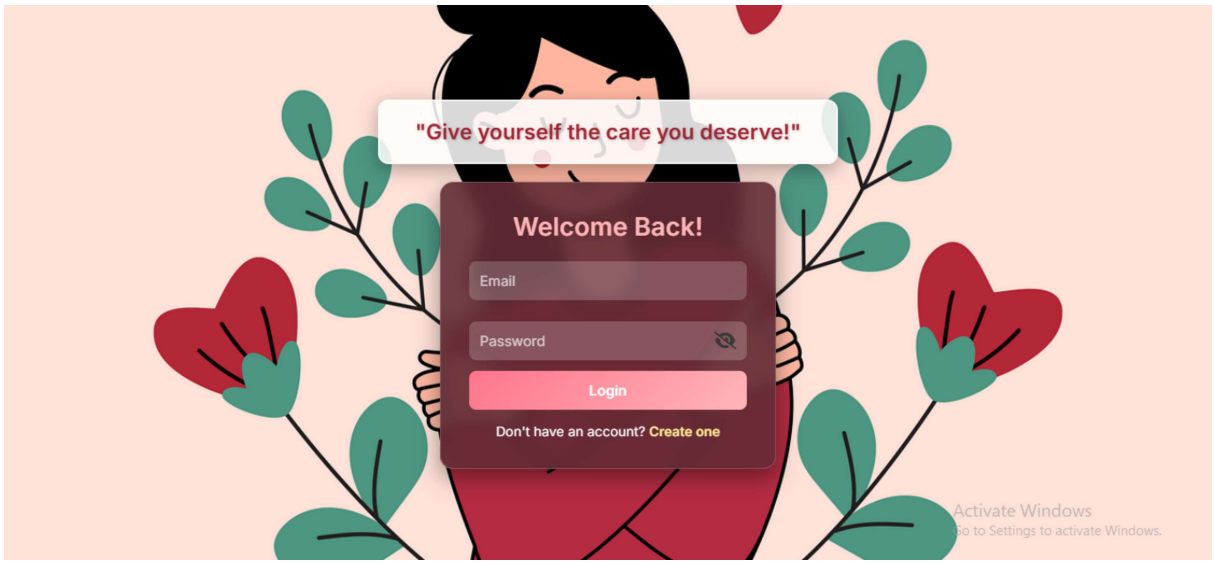

Figure 4.23: login page of mental health support system

The login page provides a secure and user-friendly interface for registered users to access their respective dashboards. Users must enter their email and password, and authentication mechanisms ensure security. If credentials are incorrect, proper validation messages guide the user, ensuring a smooth experience.

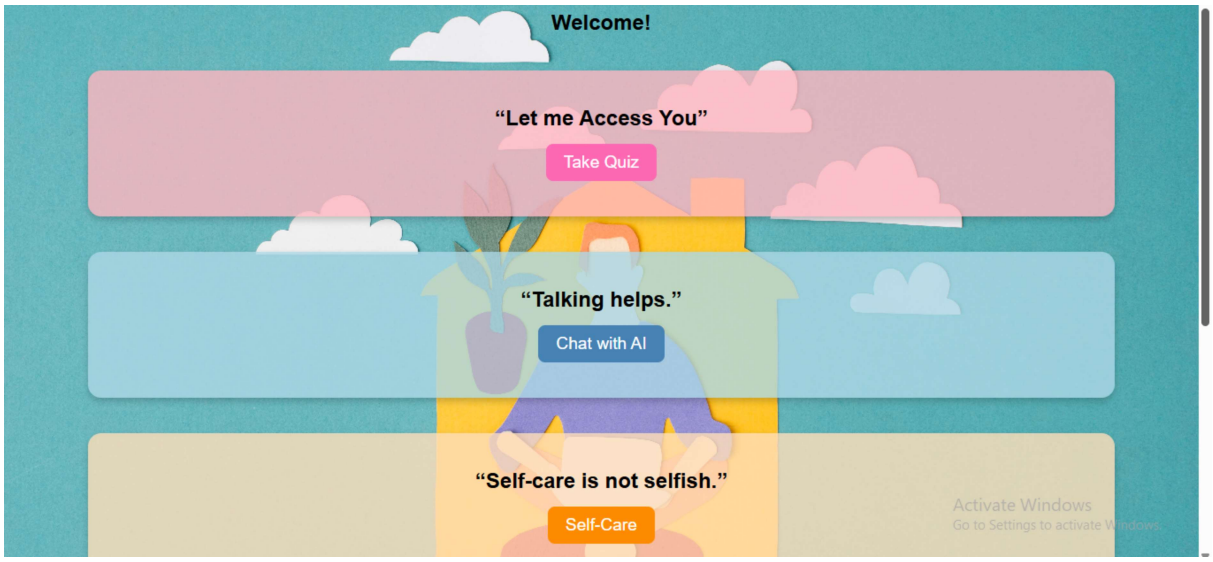

Figure 4.24: Patient dashboard

The patient dashboard acts as a central hub where users can access various features such as mental health quizzes, self-care resources, reminders, therapist recommendations, and progress tracking. The interface is designed for easy navigation, providing a personalized experience for users seeking mental health support.

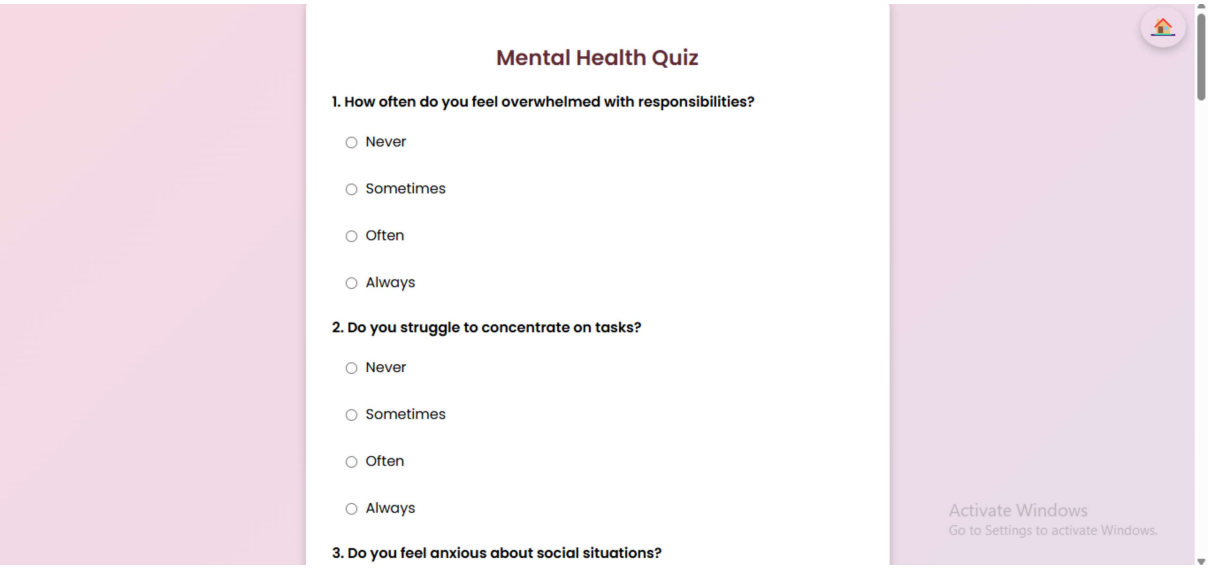

Figure 4.25: Quiz questions page

This section features an interactive quiz designed to assess a user's mental health status. It consists of multiple-choice questions covering aspects of anxiety, depression, and stress. The system processes responses to categorize users and provide personalized insights based on their quiz results.

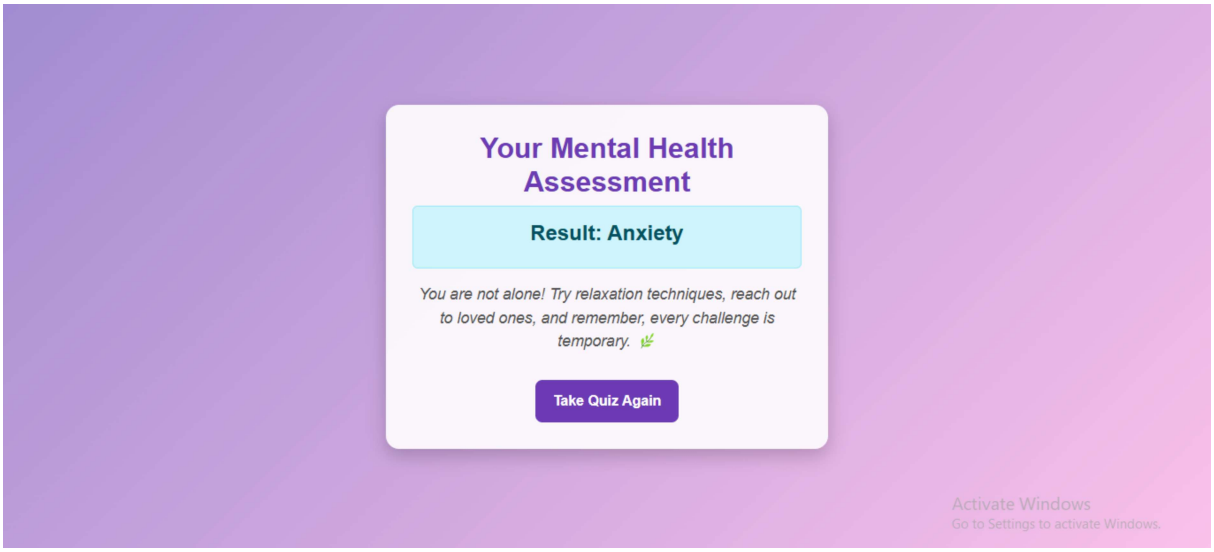

Figure 4.26: Quiz result page

After completing the quiz, users are shown their results, categorizing them under Anxiety, Depression, Stress, or Normal. Based on the assessment, the system suggests suitable self-care activities, resources, and, if needed, therapist recommendations. The results page ensures clarity, helping users take the next steps toward better mental health.

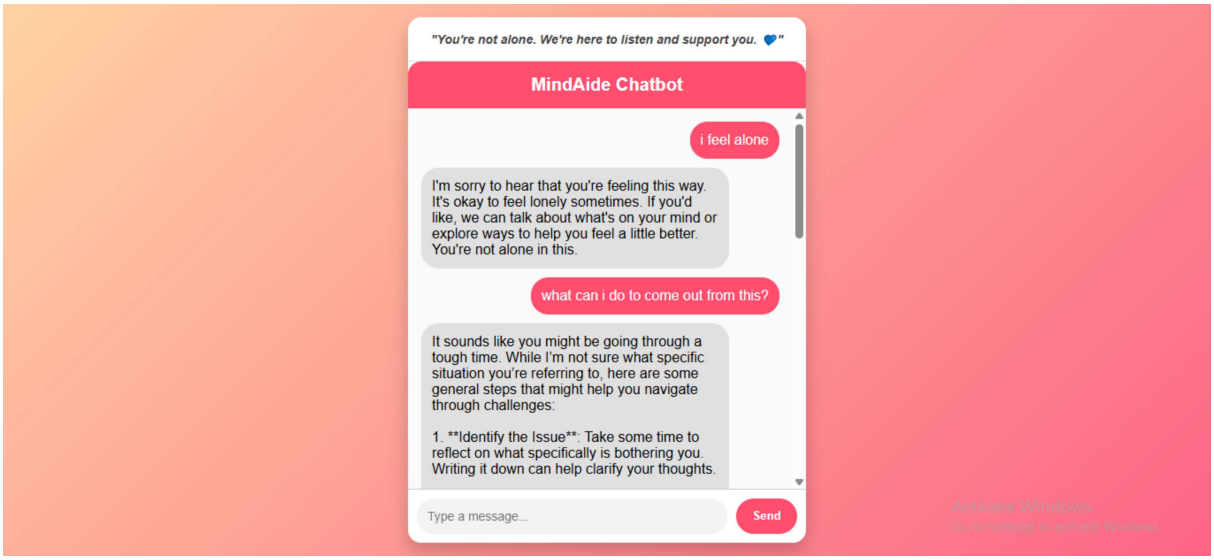

Figure 4.27: Chatbot conversation

The chatbot support feature acts as an interactive assistant, providing instant responses to user queries related to mental health. Integrated with AI, the chatbot analyzes user inputs and offers personalized guidance, including self-care tips, professional advice, and resource links. Users can engage in conversations regarding their emotions, stress levels, or mental well-being and receive immediate support. The chatbot also directs users to relevant self-care activities, therapists, or emergency contacts if needed, ensuring comprehensive assistance.

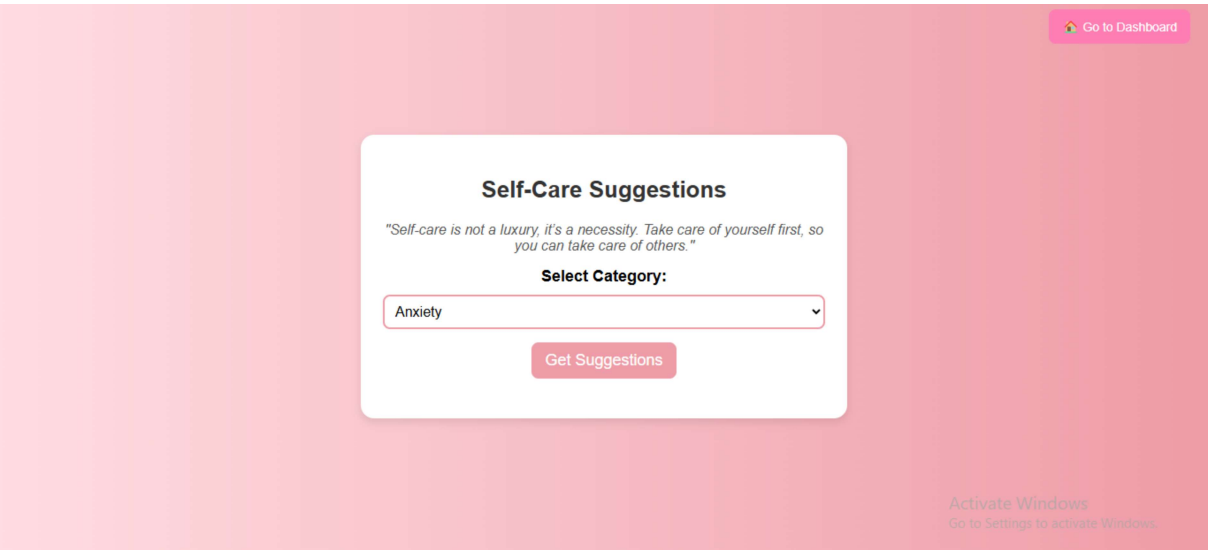

Figure 4.28: Selfcare page

The self-care page serves as a gateway to personalized mental health resources. Users can select their current mental health condition—Anxiety, Depression, Stress, or Normal—via a dropdown menu. Upon selection, the system dynamically retrieves relevant self-care suggestions tailored to the user's needs. This feature ensures that users receive appropriate

coping mechanisms, self-help strategies, and wellness resources suited to their condition.

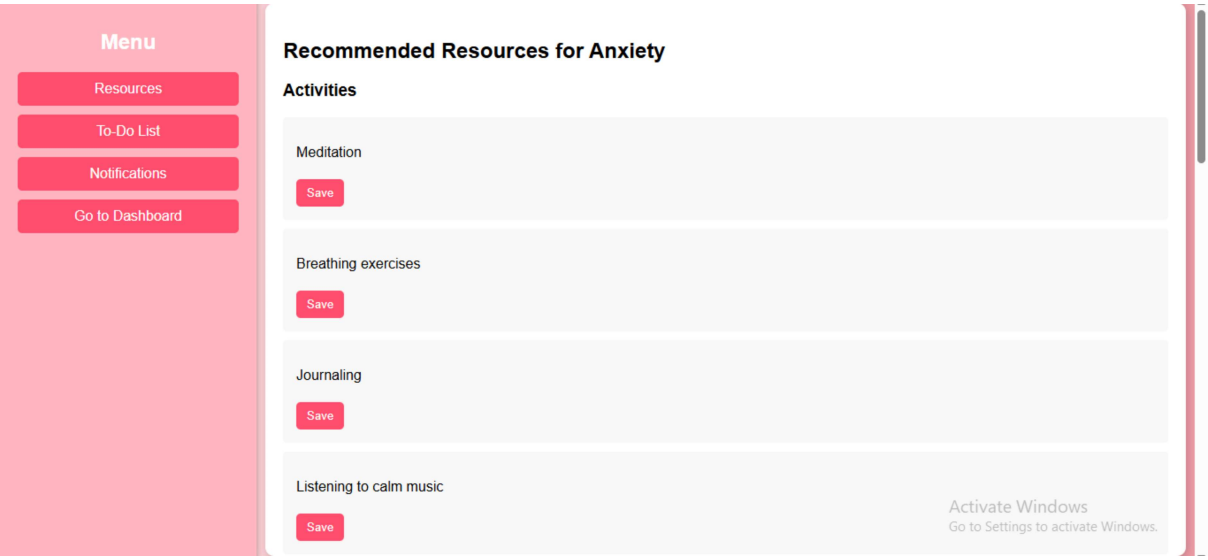

Figure 4.29: Selfcare recommendations

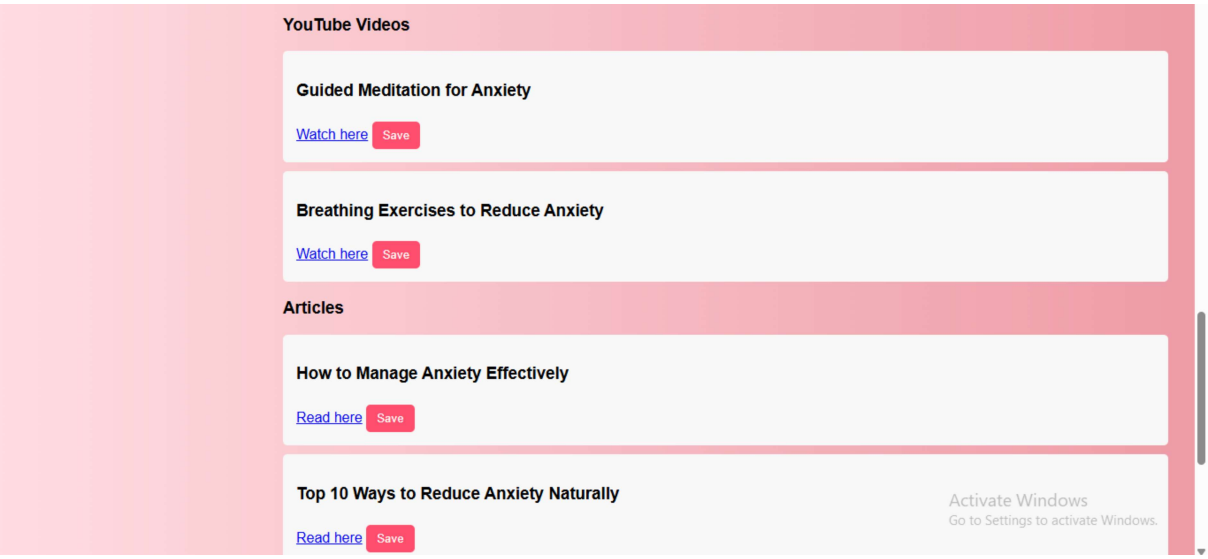

Figure 4.30: Selfcare recommendation page2

After selecting a condition, users are directed to this page, where a variety of self-care recommendations are displayed. These include engaging activities, guided exercises, educational YouTube videos, informative articles, and book recommendations. The left panel provides quick access to saved resources, a to-do list, and notifications, helping users keep track of their mental health journey. The seamless layout ensures easy navigation, while reminders and progress tracking keep users motivated to follow through with suggested activities.

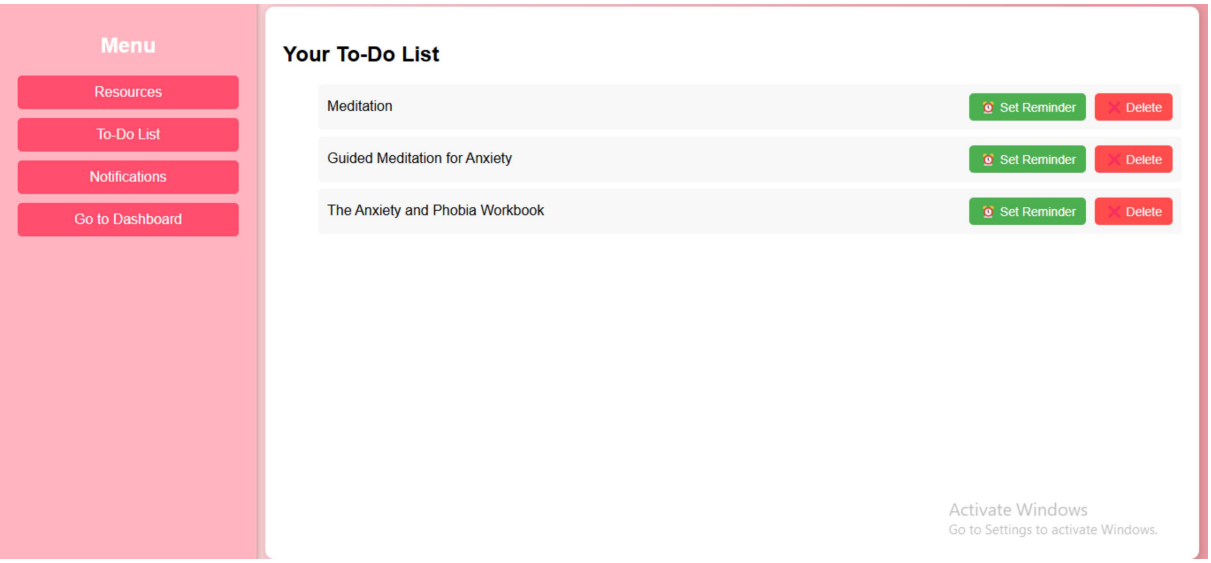

Figure 4.31: The todo list of patient

The to-do page allows users to create a personalized list of self-care tasks. Users can add activities from the self-care suggestions or create their own tasks. Each item can be assigned a reminder to ensure consistency in mental health care. Tasks are categorized as ”Pending,” ”Done,” or ”Missed,” allowing users to track their progress effectively. Notifications alert users about upcoming tasks, making it easier to stay committed to their self-care routine.

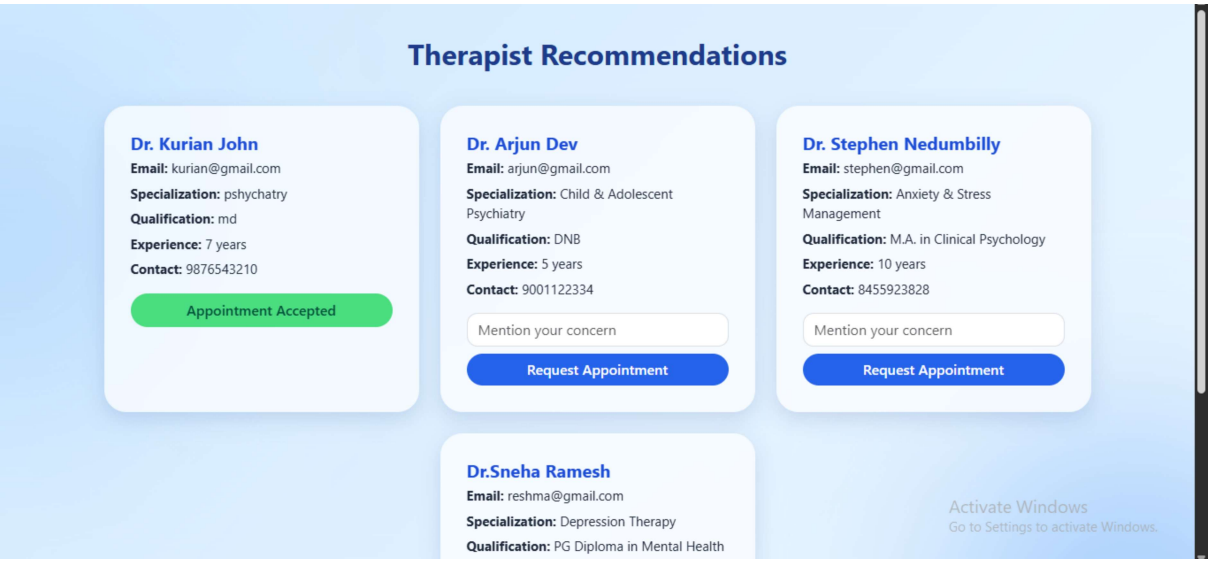

Figure 4.32: Therapist suggestions

This section enables users to explore available therapists based on their mental health needs. Users can view therapist profiles, which include qualifications, experience, availability, and areas of expertise. They can request appointments with therapists and track the status of their requests. This feature bridges the gap between users and mental health professionals, making it easier to seek professional support when needed.

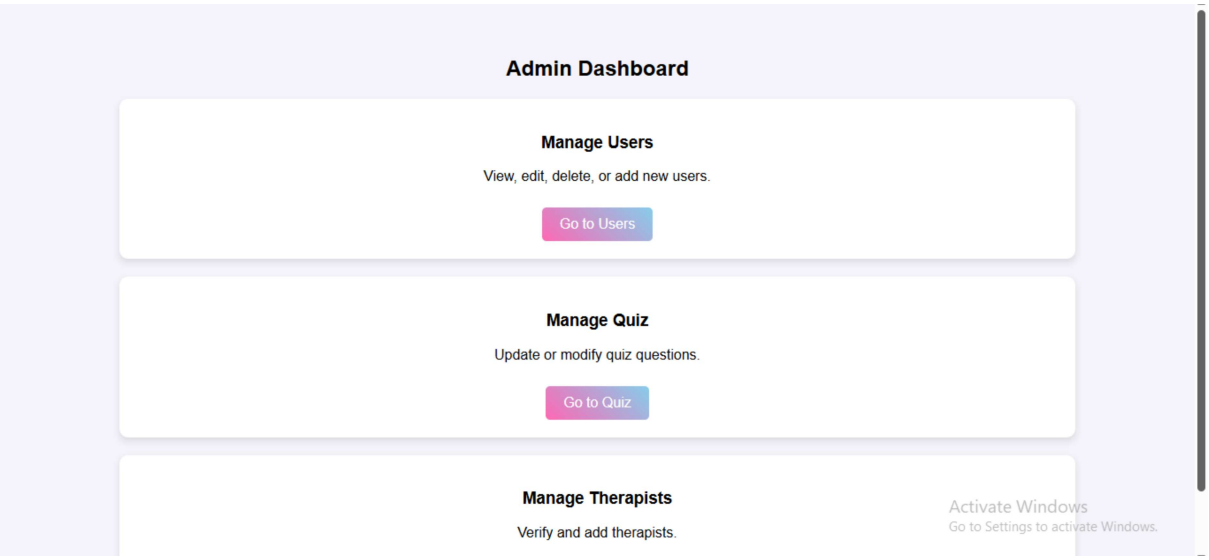

Figure 4.33: Admin dashboard

The admin dashboard provides a centralized control panel for managing the mental health support system. It offers an overview of registered users, therapists, quiz submissions, and appointment requests. Admins can monitor system activities, approve therapist registrations, manage quiz questions, and oversee user interactions. The dashboard is designed for efficient navigation, ensuring smooth system administration.

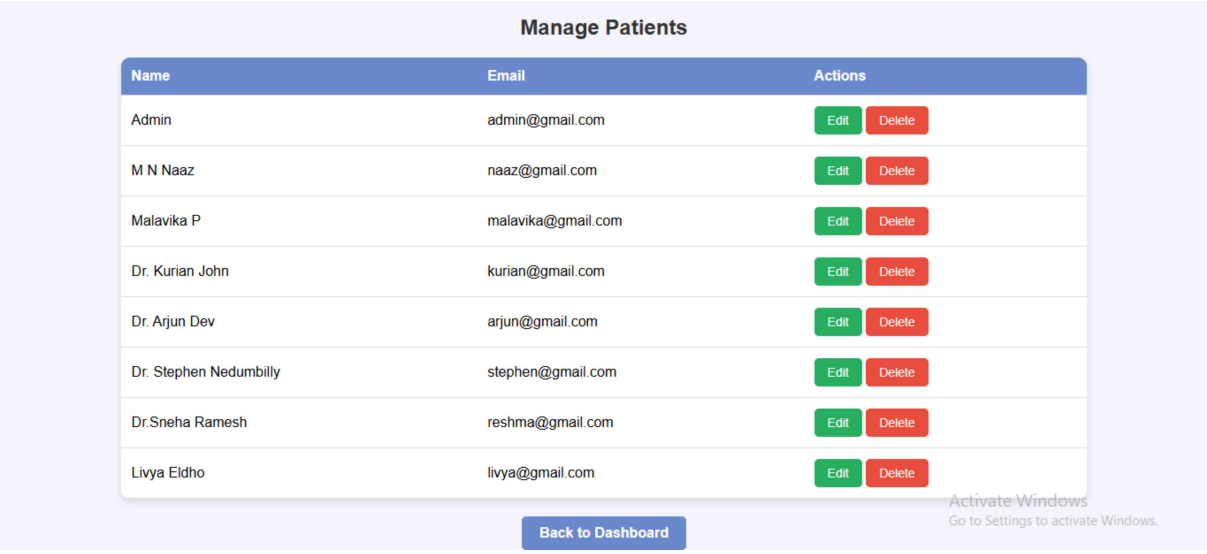

Figure 4.34: Manage users page

This page lists all registered users, categorized by roles such as Patients, Caretakers, and Therapists. Admins can filter users, search by name or email, and take necessary actions, including editing profiles, suspending accounts, or verifying identities. This feature ensures proper user management and system security.

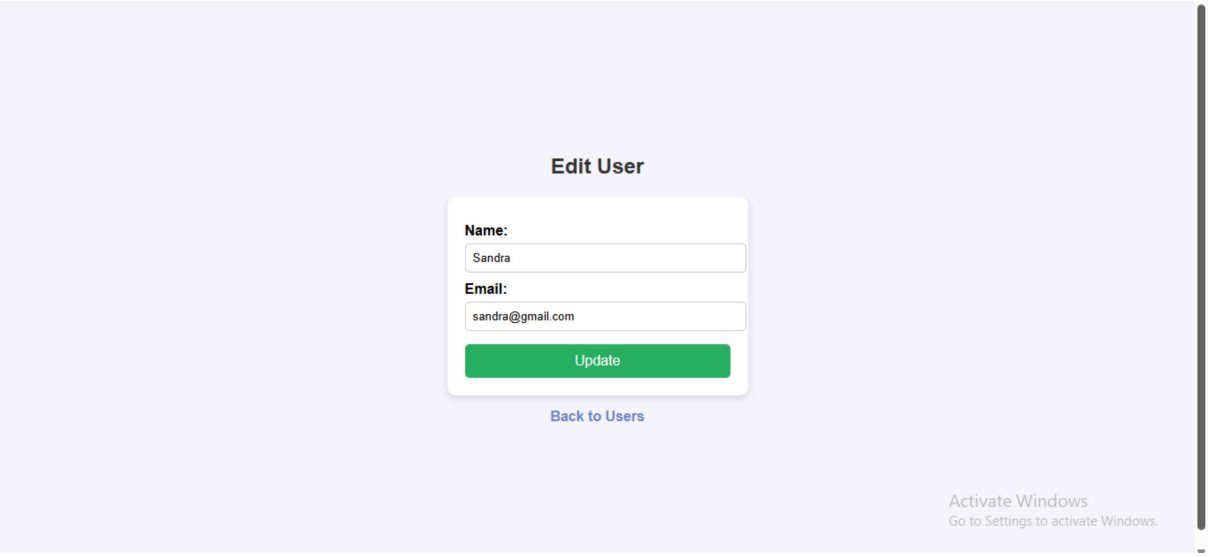

Figure 4.35: Edit users page

The edit user page allows admins to update user details, such as name, contact information, and role. Admins can also reset passwords, activate/deactivate accounts, or update other essential settings. This feature ensures that user information remains accurate and up-to-date for a smooth user experience.

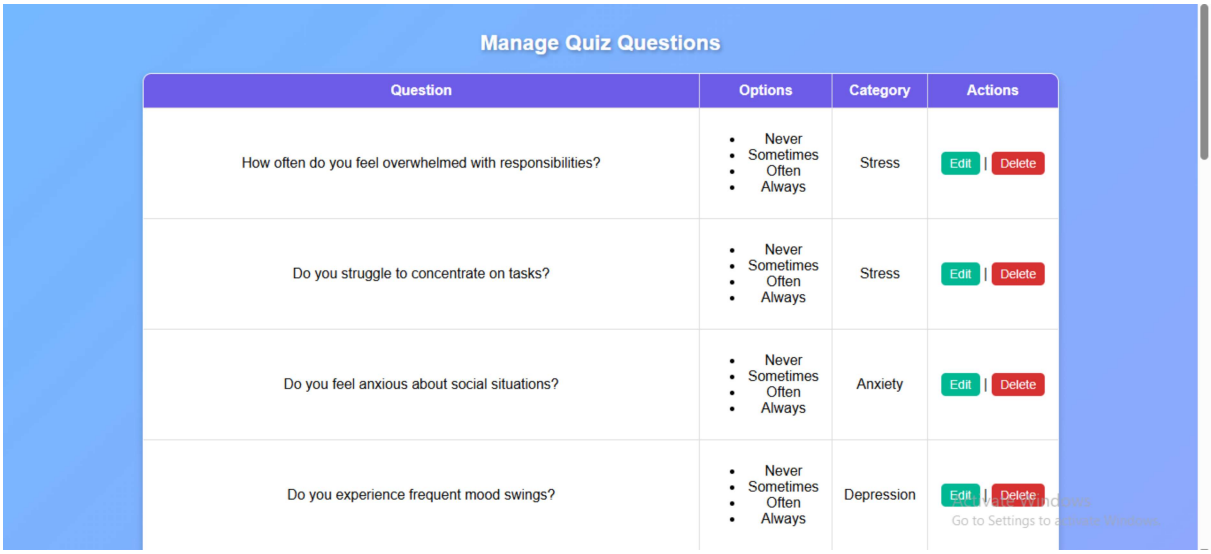

Figure 4.36: Manage quiz page

This page provides an interface for managing the mental health quiz. Admins can view all quiz questions, edit existing ones, delete outdated ones, and add new questions. Ensuring that the quiz remains relevant is crucial for accurate mental health assessment.

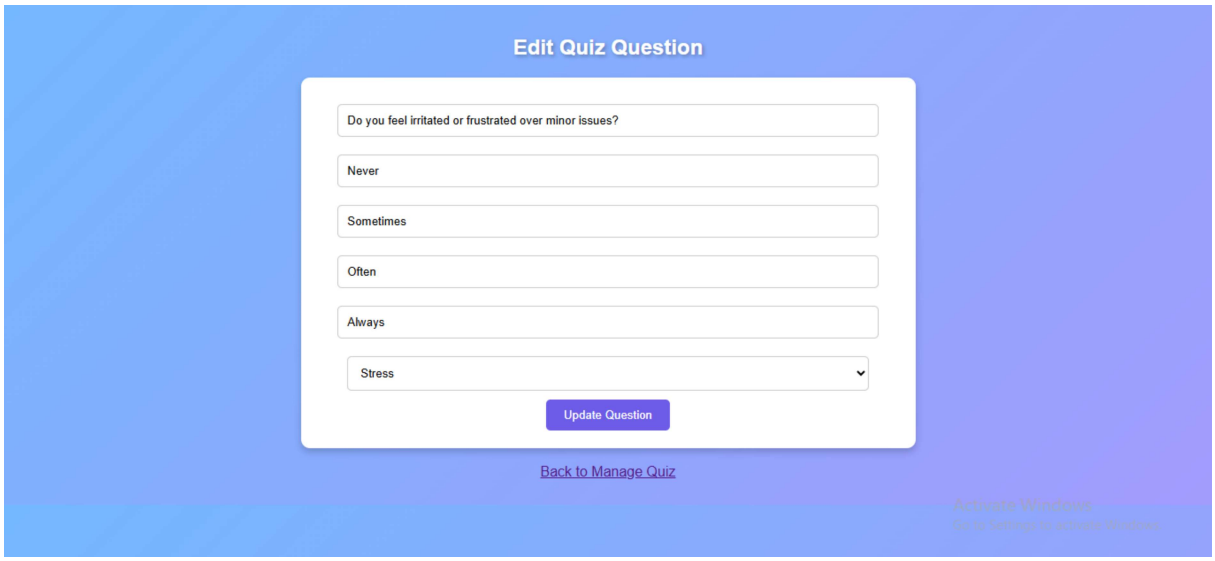

Figure 4.37: Edit Quiz page

Here, admins can modify existing quiz questions, change answer options, and update scoring mechanisms to improve assessment accuracy. This functionality helps refine the quiz to provide better mental health insights.

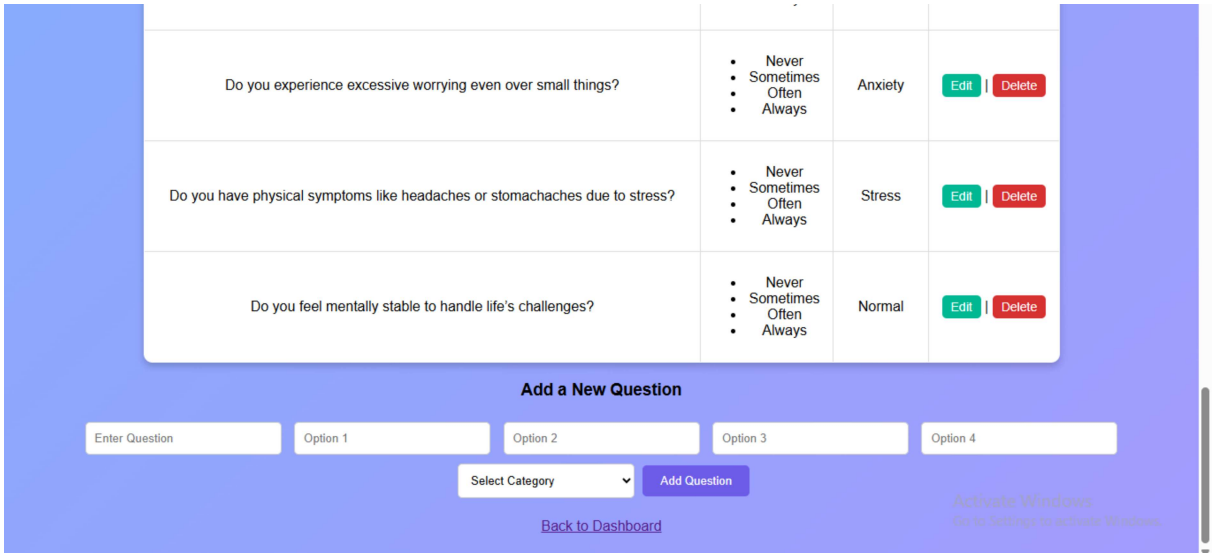

Figure 4.38: Add Quiz page

This page allows admins to expand the mental health quiz by adding new questions. Each question is assigned a category (Anxiety, Depression, Stress, or Normal), and appropriate answer choices are defined to ensure effective mental health evaluation.

Figure 4.39: Manage therapists page

The manage therapists page provides an overview of all registered therapists. Admins can review therapist credentials, verify qualifications, and approve or reject new registrations. This ensures that only certified professionals are available for consultations.

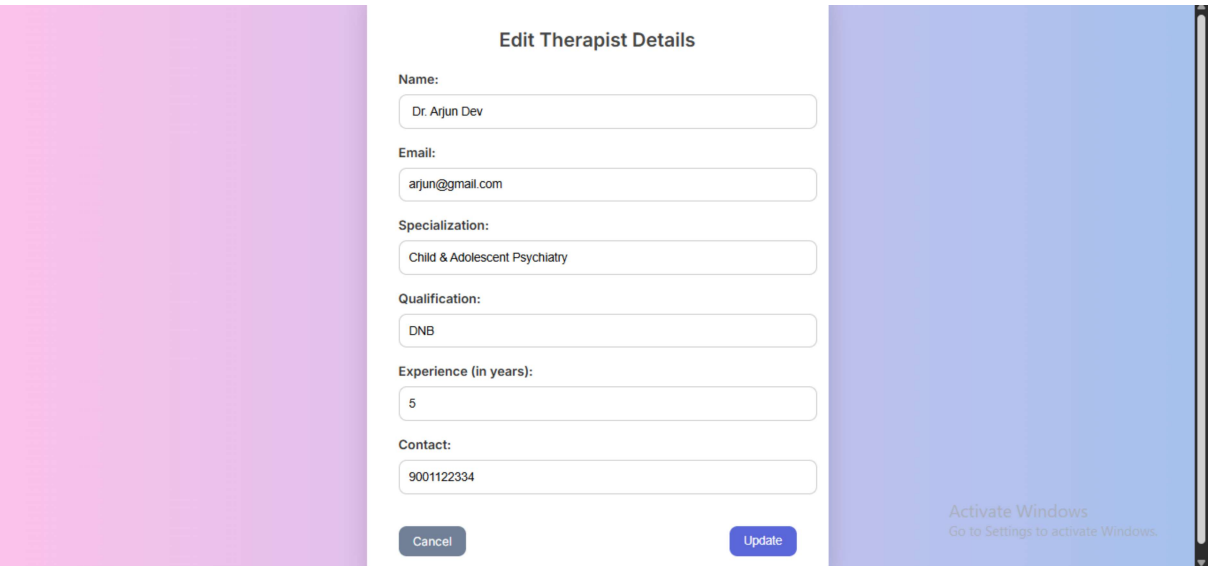

Figure 4.40: Edit therapists page

The edit therapists page allows admins to update therapist details, including name, specialization, availability, and contact information. Ensuring accurate therapist records helps maintain the credibility of the mental health support system.

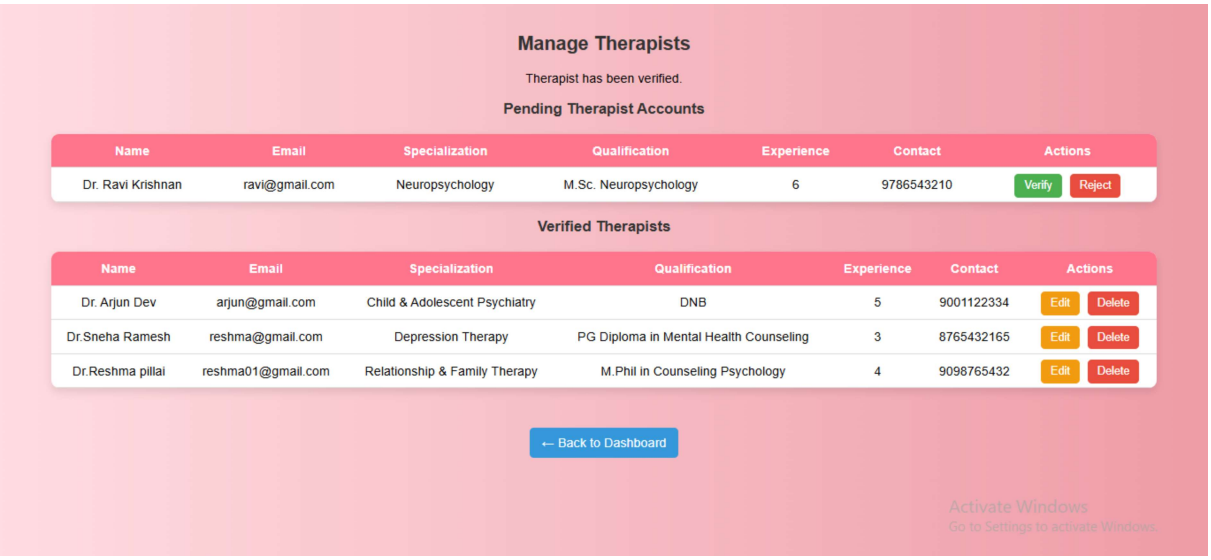

Figure 4.41: Verify Therapists page

This page categorizes therapists based on their verification status. Pending therapists are those awaiting admin approval, while verified therapists are those who have been approved and can offer consultations. Admins can review pending applications, check provided credentials, and decide whether to approve or reject therapists, ensuring only qualified professionals are onboarded.

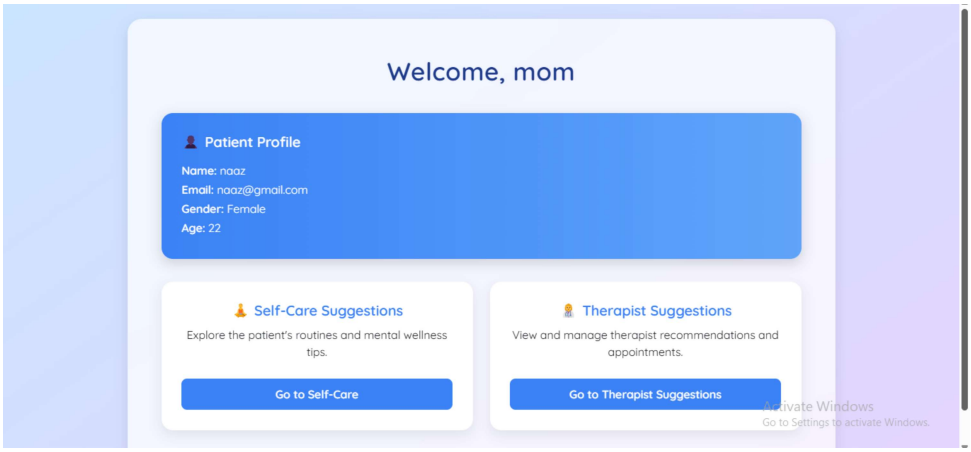

Figure 4.42: Caretaker dashboard page

The Caretaker Dashboard provides caretakers with an overview of their linked patient's mental health journey. It displays the patient's basic profile information and offers quick access to key support features like Self-Care Suggestions and Therapist Recommendations. This allows caretakers to stay involved, monitor progress, and assist patients in managing their well-being more effectively through a supportive and user-friendly interface.

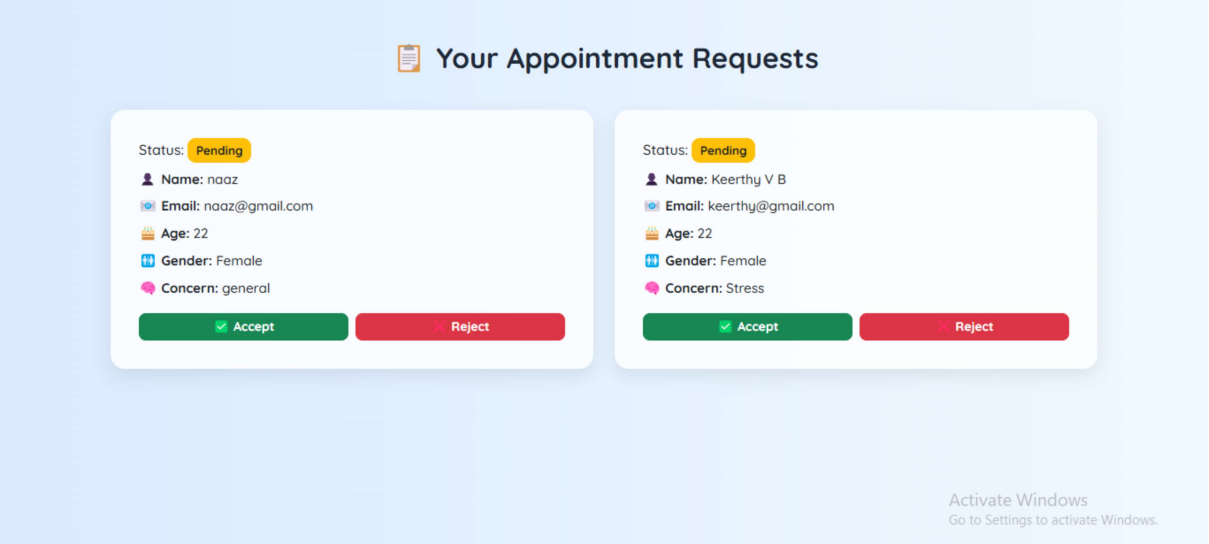

Figure 4.43: Therapist dashboard page

This page displays the therapist's dashboard for managing appointment requests from patients. Each request card shows the patient's details—name, email, age, gender, and mental health concern—along with the current status marked as Pending. Therapists can choose to accept or reject requests using clearly labeled buttons, enabling efficient appointment handling in a clean and modern interface.

5 TESTING

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, and code generation. Once the source code has been generated, the program should be executed before the customer receives it, with the specific intent of finding and removing all errors. Tests must be conducted systematically and designed using disciplined techniques.

The importance of software testing and its implications with respect to software quality cannot be overemphasized. Software testing is a critical element of software quality assurance and represents the ultimate review of specifications, design, and code generation. Once the source code has been generated, software must be tested to uncover as many errors as possible before delivery to the customer. To find the highest possible number of errors, tests must be conducted systematically, and test cases must be designed using disciplined techniques.

As the number of possible tests for even simple software components is practically infinite, all software testing uses some strategy to select tests that are feasible given the available time and resources. As a result, software testing typically (but not exclusively) attempts to execute a program or application with the intent of finding software bugs.

Software testing can be stated as the process of verifying and validating that a software application is bug-free, meets the technical requirements as guided by its design and development, and meets the user requirements effectively and efficiently by handling all exceptional and boundary cases.

The process of software testing aims not only to find faults in the existing software but also to find measures to improve the software in terms of efficiency, accuracy, and usability. It mainly aims at measuring specification, functionality, and performance of a software program or application. Software testing also helps to identify errors, gaps, or missing requirements contrary to the actual requirements. It can be conducted either manually or using automated tools.

Some prefer classifying software testing as White Box and Black Box testing. Software testing can be divided into two major steps:

- **Verification:** It refers to the set of tasks that ensure that software correctly implements a specific function.
- **Validation:** It refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements.

Phases of Software Testing The testing process can be divided into five phases:

1. **Unit Testing**
2. **Integration Testing**
3. **Final/System Testing**
4. **Backend Testing**
5. **GUI Testing**

5.1 Unit Testing

Unit testing is the process of testing individual components or functions of a software system in isolation to ensure they work correctly. It is a fundamental part of software testing that helps identify and fix errors at an early stage, improving the overall reliability and maintainability of the application. Unit tests are typically written by developers and executed automatically to verify that a specific module, function, or class performs as expected. This testing method focuses on testing small, manageable parts of the software separately before they are integrated into the entire system. By conducting unit testing, developers can ensure that each unit of the software meets the specified requirements, functions correctly under various inputs, and handles edge cases effectively.

In my mental health support system, unit testing focuses on verifying the core functionalities of individual components before integrating them into the full system. Each module, including user authentication, quiz assessment, chatbot interaction, self-care activity management, and reminders, is tested separately to ensure they function correctly. For example, unit testing verifies that user login and registration handle both valid and invalid credentials appropriately. The quiz assessment module is tested to ensure accurate score calculation and category determination based on user responses. Similarly, the chatbot module is tested to generate correct responses based on different queries. The self-care module is validated by checking whether users receive appropriate recommendations based on their mental health assessment results. Additionally, reminder notifications are tested to confirm they are triggered at the correct time. Through systematic unit testing, potential errors can be identified and fixed early, ensuring that the system functions smoothly before moving on to integration and system testing.

5.1.1 Unit Test Cases

Sl. No	Procedure	Expected Output	Actual Output	Status
1	Validate User Registration with valid data	Registration successful	Registration successful	Pass
2	Validate User Registration with missing or invalid data	Registration failed	Registration failed	Pass
3	Validate User Login with correct credentials	Login successful	Login successful	Pass
4	Validate User Login with incorrect credentials	Login failed	Login failed	Pass
5	Fetch and display quiz questions from database	Questions displayed successfully	Questions displayed successfully	Pass
6	Submit quiz with valid responses	Quiz results calculated and stored	Quiz results calculated and stored	Pass
7	Validate chatbot response for predefined queries	Correct response displayed	Correct response displayed	Pass
8	Validate chatbot response for unknown query	Default help message displayed	Default help message displayed	Pass
9	Save a self-care activity	Activity saved successfully	Activity saved successfully	Pass
10	Set a reminder for a self-care activity	Reminder set successfully	Reminder set successfully	Pass
11	Validate therapist request submission	Request sent successfully	Request sent successfully	Pass
12	Validate therapist appointment approval	Appointment approved	Appointment approved	Pass

Table 5.1: Unit Test Cases

5.2 Integration Testing

Integration testing ensures that individual modules in a software system work together correctly when combined. It systematically constructs the system while identifying errors in interactions between integrated components. The primary goal is to verify data flow, communication, and dependencies among different modules to ensure they function seamlessly as a unified system. Integration testing follows different approaches, such as top-down, bottom-up, and hybrid testing, depending on the system architecture. By conducting integration testing, potential defects in module interaction can be detected early, ensuring the reliability and consistency of the application.

In the mental health support system project, integration testing plays a vital role in verifying the seamless interaction between different modules such as user authentication, quiz assessment, chatbot, self-care module, reminders, and therapist suggestions. The top-down integration approach is followed, where core modules such as authentication and dashboard functionalities are tested first, followed by the integration of the quiz module, chatbot, and self-care functionalities.

Initially, user authentication is integrated with the dashboard, ensuring that different user roles (patients, therapists, and admins) are redirected to their respective dashboards. The quiz module is tested to confirm that responses are correctly processed and categorized into anxiety, depression, stress, or normal, displaying the corresponding recommendations. The chatbot integration is verified to ensure it provides responses based on quiz results and user queries. The self-care module is tested for proper storage and retrieval of recommended activities, while the reminder system ensures timely notifications and task tracking. Lastly, the therapist suggestion module is validated to check proper appointment scheduling and status updates. Through systematic integration testing, errors in module interaction are identified and resolved, leading to a stable and functional system.

5.2.1 Integration Test Cases

Sl.no	Procedure	Expected Output	Actual Output	Status
1	Integration of user authentication and dashboard	Users should be redirected to their respective dashboards based on role	Users are redirected correctly	Pass
2	Integration of quiz module with assessment result	Quiz answers should be processed, and the correct mental health category should be displayed	Quiz assessment result displayed correctly	Pass
3	Integration of chatbot with mental health categories	The chatbot should provide responses based on the quiz assessment result	Chatbot provides appropriate responses	Pass
4	Integration of self-care module with quiz results	Self-care recommendations should be displayed based on mental health category	Recommendations displayed correctly	Pass
5	Integration of self-care module with reminders	Users should be able to set reminders for self-care activities, and notifications should be sent on time	Reminders are set and notifications are received	Pass
6	Integration of therapist suggestions with user dashboard	Users should be able to view therapist profiles and request appointments	Therapist profiles and appointment requests are functional	Pass
7	Integration of appointment requests with therapist approval system	Therapists should be able to approve or reject appointment requests, and users should see status updates	Appointment approvals and rejections are updated correctly	Pass
8	Integration of completed activities with user progress tracking	Users should be able to mark activities as 'Done' or 'Missed,' and progress should be tracked	Activity status updates and progress tracking work correctly	Pass

Table 5.2: Integration Test Cases

5.3 System Testing

System testing involves evaluating the entire system to ensure that it meets its specified requirements and functions correctly in its intended environment. It is a critical phase in the software development lifecycle, especially for the mental health support system project, as it ensures that all modules, components, and functionalities work seamlessly together. Unlike unit testing, which focuses on individual components, and integration testing, which tests the interaction between modules, system testing assesses the system as a whole.

The primary goal of system testing is to validate that the system performs as expected and meets both functional and non-functional requirements. Functional testing verifies that the system functions as intended, while non-functional testing ensures that the system performs efficiently, securely, and in a scalable manner. System testing also covers aspects like security, usability, performance, and compatibility across various devices and platforms.

In the context of the mental health support system, system testing plays a vital role in ensuring that all integrated modules—such as user authentication, the mental health quiz, self-care suggestions, reminders, chatbot interaction, and therapist suggestions—work together flawlessly. By conducting end-to-end testing, the system is validated under real-world conditions, ensuring that user interactions, data flow, and module communication function as expected.

System testing is carried out by executing predefined test cases, comparing the actual output with the expected output, and resolving any discrepancies. The aim is to identify defects early in the development cycle, improve the overall quality of the system, and ensure that it meets user expectations.

5.3.1 System Testing Test Cases

Test Case: User Registration and Login Process

Expected Output:

1. User Registration: User details stored, confirmation message displayed.

2. Login Process: User redirected to the respective dashboard based on the role (patient, therapist, admin).

Actual Output:

1. User Registration: Successful storage, confirmation shown.

2. Login Process: User redirected to respective dashboards correctly.

Status: Pass

Test Case: Mental Health Quiz Submission and Result Categorization

Expected Output:

1. Quiz Submission: Responses processed, categorized as Anxiety, Depression, Stress, or Normal.

2. Result Display: Correct mental health category displayed with recommendations for self-care activities.

Actual Output:

1. Quiz Submission: Responses processed and categorized correctly.

2. Result Display: Correct mental health category displayed with the relevant suggestions.

Status: Pass

Test Case: Chatbot Response Based on User Queries

Expected Output:

1. Chatbot Interaction: User inputs queries, chatbot provides responses based on quiz results.

2. Relevant Responses: Chatbot gives appropriate suggestions or mental health support based on the detected condition.

Actual Output:

1. Chatbot Interaction: User queries processed, chatbot responded as expected.

2. Relevant Responses: Chatbot provided correct responses and support based on user conditions.

Status: Pass

Test Case: Self-Care Module Suggestions and Reminder Setup

Expected Output:

1. Self-Care Suggestions: Correct activities, exercises, and resources displayed based on the user's category.

2. Reminder Setup: Users can successfully set reminders for saved activities.

Actual Output:

1. Self-Care Suggestions: Activities and resources displayed correctly for each mental health category.

2. Reminder Setup: Reminders set successfully and notifications sent at the scheduled time.

Status: Pass

Test Case: Therapist Appointment Request and Status Update

Expected Output:

1. Therapist Listing: Users can view available therapists and request appointments.

2. Appointment Status: Therapists receive requests and can accept or reject them.

3. User Notification: Users receive updates on their appointment request status.

Actual Output:

1. Therapist Listing: Users can successfully view therapist profiles and request appointments.

2. Appointment Status: Therapists correctly receive and process appointment requests.

3. User Notification: Users receive real-time updates on appointment status.

Status: Pass

Test Case: Notifications for Saved Activities and Progress Tracking

Expected Output:

1. Saved Activities: Users can view and manage saved self-care activities.

2. Notifications: Timely notifications sent for pending activities.

3. Progress Tracking: Users can mark activities as ”Done” or ”Missed” for tracking.

Actual Output:

1. Saved Activities: Saved activities displayed correctly in the user dashboard.

2. Notifications: Notifications sent as expected for scheduled activities.

3. Progress Tracking: Users able to mark activities as ”Done” or ”Missed” successfully.

Status: Pass

Test Case: Secure Data Storage and Retrieval

Expected Output:

1. User Data: Personal details, quiz results, and therapy appointments securely stored in MongoDB.

2. Data Retrieval: System retrieves data efficiently without any inconsistency.

Actual Output:

1. User Data: All data securely stored and protected in the database. 2. Data Retrieval: Retrieval works as expected, with accurate and consistent data.

Status: Pass

By executing these system test cases, the Mental Health Support System ensures smooth module interaction, correct feature functionality, and reliable system performance.

5.4 Backend Testing

Backend testing is a crucial process that ensures the reliability, security, and efficiency of a system by focusing on databases, server logic, and APIs. Unlike frontend testing, which evaluates the user interface, backend testing verifies that data is correctly processed, stored, and retrieved without errors. This involves checking database interactions, API responses, business logic execution, and system security. Tools like Postman, SQL queries, and automated scripts help validate data transactions, including insertions, deletions, and updates, ensuring they adhere to business rules. It also detects critical issues such as data corruption, deadlocks, and security vulnerabilities, preventing system failures and performance bottlenecks. Security testing plays a key role by verifying authentication mechanisms, preventing unauthorized access, and ensuring sensitive data protection.

In my chatbot-integrated mental health support system, backend testing is essential for validating user authentication, mental health quizzes, self-care recommendations, therapist appointment management, and progress tracking. The system relies on a MongoDB database to store user data, quiz results, reminders, and therapist interactions. Testing ensures that all data transactions function correctly, preventing inconsistencies or loss of critical information. Flask-based APIs are rigorously tested using Postman to verify that they handle user requests efficiently, return accurate JSON responses, and maintain system performance. For example, when a user submits a mental health quiz, the backend must correctly process responses, categorize results, and store them for future tracking. Similarly, therapist appointment requests must be recorded accurately, with status updates reflecting approvals or rejections. Security testing ensures that private user data, including mental health assessments and therapy records,

remains protected from unauthorized access. By thoroughly testing the backend, my system guarantees seamless integration between all modules, delivering a secure, reliable, and user-friendly mental health support experience.

5.4.1 Backend Testing test cases

Sl. No	Procedure	Expected Output	Actual Output	Status
1	User registers in the system	User details stored in the database	User details successfully stored	Pass
2	User logs into the system	Authentication validated, user redirected to the dashboard	Login successful, user redirected correctly	Pass
3	User submits a mental health quiz	Quiz responses stored, category determined	Responses stored, category assigned correctly	Pass
4	Therapist profile is created	Profile data is inserted into the database	Profile successfully stored and retrievable	Pass
5	User requests an appointment with a therapist	Appointment request entry created	Request stored, therapist notified	Pass
6	Therapist accepts or rejects an appointment request	Status updated in the database	Status correctly updated	Pass
7	Self-care activity is added	Activity stored in the database	Activity stored and displayed to the user	Pass
8	Reminder is set for a self-care activity	Reminder entry created	Reminder stored and displayed in notifications	Pass
9	User marks a self-care activity as completed	Activity status updated	Status changed to 'Done' successfully	Pass
10	Notification is triggered for pending self-care tasks	Notification entry created and displayed	Notification appears as expected	Pass

Table 5.3: Backend Testing for Mental Health Support System

5.5 GUI Testing

GUI Testing ensures that the graphical user interface of the system functions correctly and provides a seamless user experience. It involves testing visual elements such as buttons, input fields, menus, navigation, color schemes, responsiveness, and usability. The primary goal of GUI testing is to validate that the interface aligns with design specifications and provides an intuitive, accessible, and error-free experience for users. A well-designed GUI improves user engagement, reduces confusion, and enhances the overall efficiency of interactions with the system. Testing ensures that all components are visually appealing, properly aligned, and responsive across various devices and screen sizes. Additionally, it verifies that interactive elements, such as buttons and links, function as expected and that proper error messages are displayed when needed.

In the context of my mental health support system, GUI testing is crucial to ensuring that users, including patients, therapists, and administrators, can navigate the system effortlessly. The system incorporates various UI components, such as quiz forms, chatbot interfaces, self-care recommendations, therapist booking sections, and progress tracking dashboards. Proper GUI testing ensures that quiz questions and options are displayed correctly, chatbot interactions remain smooth, and self-care suggestions are visually clear and accessible. Furthermore, therapist appointment management pages must be tested to ensure that users can view therapist profiles, request appointments, and track their request status without issues. The light pink and light blue theme must be consistent across all pages to maintain a cohesive design. Special attention is given to responsiveness testing, ensuring that the application works seamlessly on different screen sizes, including mobile devices, tablets, and desktops. By conducting thorough GUI testing, the system guarantees a visually appealing, accessible, and user-friendly experience, making mental health support more approachable and effective.

5.5.1 GUI Testing test cases

Sl. No	Procedure	Expected Output	Status
1	Check input fields for registration and login	Users should be able to enter data into input fields	Pass
2	Verify validation for email, password, and phone number	Proper validation messages should be displayed for incorrect inputs	Pass
3	Check quiz question display and response selection	Questions and options should be clearly visible and selectable	Pass
4	Verify chatbot interface	Chatbot should respond appropriately with smooth interactions	Pass
5	Check therapist profile display	Profiles should be clear, properly aligned, and readable	Pass
6	Test appointment request and status tracking UI	Booking request should be clearly displayed with correct status updates	Pass
7	Verify alignment of buttons, text fields, and labels	All elements should be properly aligned and aesthetically pleasing	Pass
8	Test responsiveness on mobile devices	UI should adjust properly across different screen sizes	Pass
9	Verify navigation between pages	Navigation should be smooth and without delays	Pass
10	Check button functionality	All buttons should respond correctly when clicked	Pass

Table 5.4: GUI Testing for Mental Health Support System

6 DEPLOYMENT

The deployment phase of the Mental Health Support System marks the transition from development to a fully operational platform accessible to users. This phase is crucial as it ensures that the system is properly configured, optimized, and integrated into a real-world environment. The deployment process involves setting up the necessary infrastructure, ensuring database connectivity, configuring the web server, and making the platform available for users, including patients, therapists, and administrators.

One of the primary challenges in deploying this system is ensuring seamless integration with various components, including the Flask backend, MongoDB database, chatbot API, and front-end interface. Given that the system relies on multiple external services, such as the OpenAI API for chatbot support, it is essential to ensure that all dependencies are correctly installed and configured. Additionally, security considerations, such as data encryption, user authentication, and secure API communication, play a vital role in safeguarding sensitive mental health data.

To streamline deployment, the project follows a structured approach, including server setup, database migration, and environment configuration. The system is hosted on a cloud-based server to ensure high availability and scalability. MongoDB, used for data storage, is configured with appropriate indexing and security measures to enhance performance and protect user information. The front end is designed for responsiveness, ensuring compatibility across different devices, including desktops, tablets, and mobile phones.

User accessibility is another key focus during deployment. The system provides a user-friendly interface for mental health assessment, self-care recommendations, therapist suggestions, and chatbot interactions. Thorough testing is conducted post-deployment to verify that all features function as expected and that users can navigate the system effortlessly. Additionally, a monitoring mechanism is implemented to track system performance, identify potential issues, and ensure timely updates and maintenance.

By addressing these deployment challenges and optimizing the system for real-world use, the Mental Health Support System is successfully transitioned into an operational state. With proper planning, security measures, and scalability considerations, the system is well-equipped to provide seamless mental health support, enhancing accessibility and assistance for users in need.

7 GIT HISTORY

Git is a distributed version control system that facilitates collaborative software development by tracking changes to source code. Developed by Linus Torvalds, it offers a robust platform for managing project histories, enabling developers to work concurrently on codebases without conflicts. Git's decentralized architecture allows for seamless branching, merging, and versioning, empowering teams to experiment and iterate efficiently. With features like lightweight branching, fast performance, and robust support for non-linear development workflows, Git has become the de facto standard for version control in the software industry, fostering collaboration, transparency, and productivity across diverse development environments.

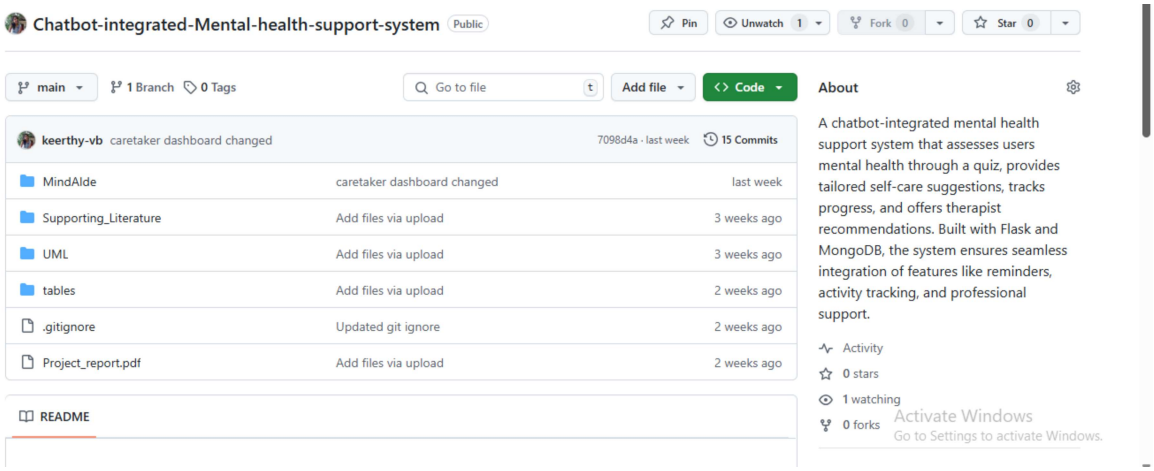

Figure 7.1: Git history

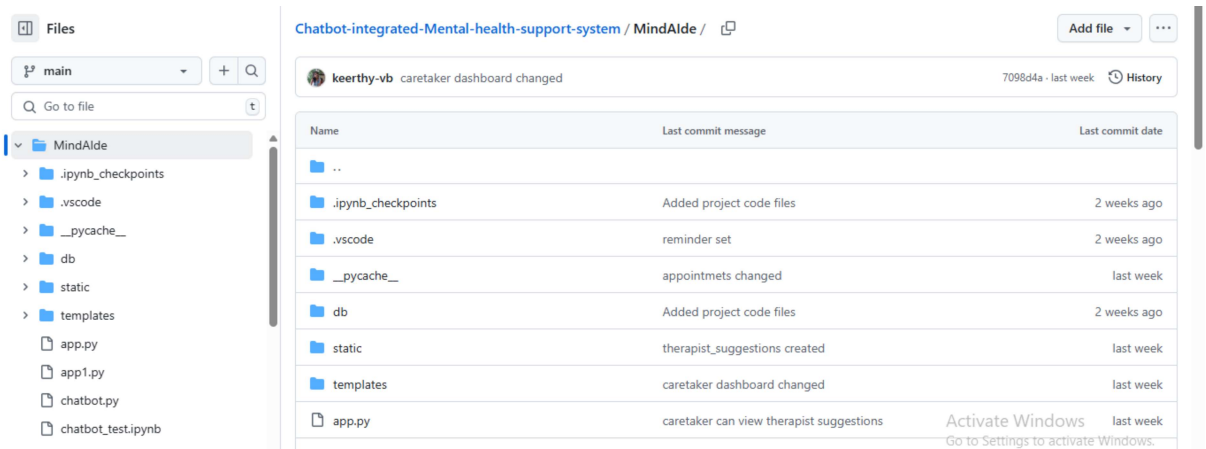

Figure 7.2: Git history of backend

8 CONCLUSIONS

The mental health support system developed in this project is a well-structured and user-friendly platform designed to help individuals assess, monitor, and improve their mental well-being. The project integrates modern technologies such as Flask, MongoDB, and OpenAI's chatbot API to create an interactive and effective system that provides valuable mental health support. The main goal is to ensure that users receive timely assistance, helpful self-care suggestions, and access to professional help when needed. The system consists of multiple modules, including a mental health quiz, self-care activities, progress tracking, chatbot support, and therapist recommendations, all working together to provide a complete mental health solution.

One of the key features of this system is its ability to offer personalized support based on user responses. The mental health quiz helps categorize users into different mental health states, such as stress, anxiety, depression, or normal. Based on the results, the system suggests self-care activities, exercises, and helpful resources like YouTube videos and articles. This personalized approach ensures that users receive the most relevant support for their mental health needs. Additionally, the chatbot provides real-time guidance by answering questions, offering coping strategies, and encouraging users to take positive steps toward better mental well-being. This makes the platform more interactive and engaging, ensuring that users feel supported at all times.

Another important aspect of this project is its ability to connect users with professional therapists. The therapist recommendation module analyzes user profiles and suggests suitable therapists based on their specific needs. Users can view therapist details, check availability, and request appointments, making it easier to seek professional help. This feature is particularly useful for individuals who may feel hesitant or unsure about reaching out for therapy. By providing a simple and accessible way to connect with professionals, the system helps bridge the gap between users and mental health care providers.

The self-care module also plays a significant role in encouraging users to maintain consistency in their mental health improvement journey. Users can set reminders for activities, track completed tasks, and receive notifications to stay on schedule. This ensures that they stay motivated and engaged with their self-care routines. The system saves all user interactions and progress in a MongoDB database, allowing for accurate tracking and easy retrieval of information. The progress tracking feature helps users monitor their improvements over time, making them more aware of their mental health patterns and the effectiveness of different self-care strategies.

To enhance the overall experience, the user interface has been designed to be visually appealing and easy to navigate. A light pink and blue theme has been used to create a calming and welcoming environment. The seamless integration of different modules ensures that users can easily switch between features without any confusion. This thoughtful design makes the platform not only functional but also comfortable to use, reducing stress and making the process of seeking mental health support more accessible.

This project highlights the potential of technology in addressing mental health challenges. By combining artificial intelligence, database management, and user-friendly interfaces, the system offers a digital solution that is effective, scalable, and accessible to a wide range of users. It empowers individuals to take control of their mental health by providing the necessary tools, guidance, and professional support. The chatbot feature, in particular, enhances the user experience by offering instant responses, which can be especially helpful during times of distress.

In conclusion, this mental health support system demonstrates how technology can be used to improve mental well-being. It provides a structured approach to mental health care by combining self-assessment, personalized recommendations, professional guidance, and progress tracking into a single platform. With future improvements such as expanding chatbot capabilities, adding more therapy resources, and incorporating advanced machine learning techniques, the system can become even more effective. This project lays a strong foundation for digital mental health interventions and has the potential to make a significant difference in the lives of individuals seeking mental health support.

9 FUTURE WORK

The future development of this mental health support system offers several exciting possibilities for improvement, ensuring a more effective and user-friendly experience. As mental health awareness continues to grow, integrating more advanced technologies and expanding the system's functionalities can make it even more beneficial for users. Future work on this project will focus on enhancing chatbot capabilities, improving therapist recommendations, expanding the self-care module, incorporating more data-driven insights, and optimizing the user interface for a seamless experience.

One of the key areas for future enhancement is the chatbot functionality. Currently, the chatbot provides general support, answers basic queries, and offers self-care suggestions. In the future, the chatbot can be upgraded to use natural language processing (NLP) and sentiment analysis to understand users' emotions more accurately. By detecting distress or negative emotions in real-time conversations, the chatbot can provide immediate emotional support and recommend urgent professional help if needed. Additionally, integrating voice-based interaction will make the chatbot more accessible to users who may find typing difficult during moments of distress.

Another significant improvement involves enhancing therapist recommendations by incorporating machine learning algorithms for better matchmaking. Currently, the system filters therapists based on user preferences and mental health conditions. However, an AI-driven recommendation engine can analyze past user interactions, preferences, and therapist success rates to provide more personalized and accurate therapist suggestions. Furthermore, enabling real-time appointment booking and live chat options with therapists will streamline the process of seeking professional help, making it more convenient for users.

The self-care module can also be expanded to include more diverse and interactive activities. Currently, it provides exercises, YouTube videos, and reading materials. In the future, gamification techniques can be introduced to make self-care more engaging. Users could earn rewards or achievements for completing activities, helping them stay motivated. Additionally, integrating virtual support groups or community discussions can encourage peer support, allowing users to share their experiences and coping strategies. Another enhancement could be AI-generated personalized self-care plans, where the system dynamically adjusts suggestions based on the user's progress and preferences.

To improve mental health progress tracking, the system can introduce data visualization tools that display users' mood trends, completed self-care activities, and therapy sessions over time. By integrating machine learning models, the system can analyze user behavior patterns and provide predictive insights, such as identifying early signs of worsening mental health. This

feature can alert users to seek professional help before their condition worsens, making the system a preventive tool rather than just a reactive one.

Another important modification would be the integration of wearable device support. Many users track their physical health using smartwatches and fitness bands. By allowing the system to sync with wearable devices, it can monitor heart rate, sleep patterns, and activity levels, using this data to assess the user's mental state. For example, an increase in heart rate combined with reduced sleep might indicate rising anxiety, prompting the system to suggest relaxation techniques or therapist consultations.

To ensure a more inclusive and globally accessible platform, multi-language support can be implemented. Currently, the system operates in English, but mental health issues affect people across different linguistic backgrounds. By adding support for multiple languages and dialects, the system can cater to a wider audience, making mental health support accessible to non-English speakers. Additionally, cultural sensitivity can be integrated into chatbot responses and self-care recommendations, ensuring that suggestions align with users' cultural backgrounds and beliefs.

Security and data privacy enhancements will also be a crucial focus for future updates. Since mental health data is highly sensitive, advanced encryption techniques and multi-factor authentication (MFA) can be implemented to strengthen user data protection. Additionally, providing users with the option to anonymously use the system without entering personal details can encourage more people to seek help without fear of stigma.

In conclusion, the future work on this mental health support system will focus on improving chatbot intelligence, therapist recommendations, self-care activities, progress tracking, wearable integrations, multi-language support, UI/UX customization, security, and professional collaborations. These advancements will make the platform more effective, engaging, and accessible for users, ensuring that they receive the best possible support for their mental well-being. As technology continues to evolve, integrating AI-driven features and data-driven insights will make this system a powerful tool in the field of digital mental health support, empowering individuals to take control of their mental wellness in a more personalized and proactive way.

10 APPENDIX

10.1 Minimum Software Requirements

To ensure the smooth operation of the Mental Health Support System, the following software requirements must be met:

1. **Operating System**
    - Windows 10 or later / macOS / Linux (Ubuntu 18.04+)
    - Android 8.0 or later / iOS 12 or later (for mobile users)
2. **Programming and Frameworks**
    - Python 3.11 or later
    - Flask framework (installed via pip)
    - Django template engine
3. **Database**
    - MongoDB (Community or Atlas version)
    - PyMongo library for Python
4. **Browser Compatibility (For Web Application)**
    - Google Chrome / Mozilla Firefox / Microsoft Edge / Safari (latest versions)
    - JavaScript and cookies enabled
5. **Additional Requirements**
    - OpenAI API access for chatbot functionality
    - Email service for notifications
    - Internet access for database synchronization and API communication

10.2 Minimum Hardware Requirements

The system requires the following minimum hardware specifications:

1. **Processor (CPU)**
    - PC/Laptop: Dual-core 2.0 GHz or higher
    - Mobile: Quad-core 1.8 GHz or higher
2. **Memory (RAM)**
    - Minimum: 4GB RAM
    - Recommended: 8GB RAM for better performance
3. **Storage (SSD/HDD)**
    - Minimum: 500MB free space for application installation and logs
    - Recommended: 1GB+ for smooth performance
4. **Graphics and Display**
    - PC/Laptop: Full HD (1920×1080) resolution recommended
    - Mobile: 720p resolution or higher
5. **Internet Connectivity**
    - Stable broadband or mobile network (Wi-Fi/4G/5G)

11 REFERENCES

[1] Kang, Boyoung, and Munpyo Hong. ”Development and Evaluation of a Mental Health Chatbot UsingChatGPT 4.0: Mixed Methods User Experience Study With Korean Users.”JMIRMedicalInformatics 13 (2025): e63538. doi: 10.2196/63538

[2] Oghenekaro, Linda Uchenna, and Christopher Obinna Okoro. ”Artificial intelligence-basedchatbotforstudent mental health support.” Open Access Library Journal 11.5 (2024): 1-14 doi: 10.2196/63538

[3] Moilanen, Joonas, et al. ”Supporting mental health self-care discovery through a chatbot.” Frontiersindigital health 5 (2023): 1034724. DOI 10.3389/fdgth.2023.1034724

[4] MongoDB Official Documentation - https://www.mongodb.com/docs/

[5] https://www.draw.io.

[6] https://plantuml.com/.