LIST OF FIGURES

TABLE OF CONTENT

CHAPTER-1

INTRODUCTION

1.1 Overview of the AI Therapist Concept

AI Therapist is a software-driven solution designed to simulate aspects of traditional therapy by offering mental health support through conversational interfaces and personalized interventions. It uses artificial intelligence (AI) techniques such as Natural Language Processing (NLP) and machine learning to understand user inputs, assess emotional states, and recommend appropriate therapeutic resources or actions.

The growing demand for accessible and affordable mental health services has driven the development of AI-based therapeutic tools. Unlike traditional therapy, which requires physical appointments, an AI therapist is available 24/7, offering consistent support without the need for scheduled visits. By automating certain aspects of mental health care, AI can alleviate pressure on healthcare systems and provide immediate assistance to those in need.

1.2 Current Challenges in Mental Health

Mental Health Crisis

Mental health issues such as anxiety, depression, and stress-related disorders are at an all-time high globally. According to the World Health Organization (WHO), depression is the leading cause of disability worldwide, and suicide rates have increased significantly. Despite the growing awareness, many people still suffer in silence due to a lack of accessible resources.

Barriers to Accessing Therapy

Several obstacles prevent individuals from seeking professional help:

- **Cost:** Traditional therapy sessions can be expensive, making them unaffordable for many.
- **Stigma:** Societal norms and misconceptions about mental illness discourage people from seeking help.
- **Availability:** In rural or underdeveloped areas, there is often a shortage of qualified therapists, leaving many without professional support.

1.3 Role of Technology in Mental Health

Emergence of Digital Therapeutics

In recent years, digital therapeutics have transformed mental health care. Mobile applications, chatbots, and virtual reality environments are now being used to provide therapeutic interventions. These tools complement traditional therapy by offering self-help options, mood tracking, and mindfulness exercises**.**

Artificial Intelligence in Healthcare

AI has proven to be a valuable tool in healthcare, particularly in mental health. Its ability to analyze large datasets and identify patterns enables it to deliver personalized care. AI-driven mental health applications can assess emotional states through voice, text, and behavioral data, offering tailored recommendations or exercises to users.

1.4 Front-End Design: Bridging Users and Technology

The front end is the first point of contact for users, where they interact with the AI Therapist. It has been designed with simplicity and accessibility in mind, ensuring that people of all ages and technical backgrounds can use it without difficulty.

Responsive Design: Using tools like HTML, CSS, JavaScript, and frameworks such as node.js, the system ensures compatibility across various devices— whether it's a smartphone, tablet, or desktop.

- **Accessibility Features**: Support for features like voice input, dark mode, and easy navigation ensures inclusivity for a wide range of users.

1.5 Fire store Database: The Backbone of Data Handling

A critical part of the system is the Fire store database, which securely stores and organizes user data.

- **Real-Time Capabilities**: Firestore allows the system to instantly sync data such as chat logs, user preferences, and mood scores between the front end and back end.
- **Security**: With built-in encryption and Firebase Authentication, user data is protected from unauthorized access.
- **Scalability**: The database is designed to handle a growing number of users without compromising performance, making it ideal for large-scale adoption.

1.6 Natural Language Processing: Understanding User Conversations.

- **Language Understanding**: By leveraging pre-trained language models like BERT or GPT, the chatbot can comprehend user input, including enhanced questions or emotional expressions.
- **Context Awareness**: The system retains the context of ongoing conversations, ensuring continuity and relevance in its responses.
- **Adaptability**: It can handle diverse conversational topics related to mental health, from stress management to tips for improving mood.

1.7 Sentiment Analysis: Gauging Emotional State.

Understanding how a user feels is essential for providing meaningful support.

- **Emotion Detection**: Sentiment analysis algorithms analyze user inputs to determine if they are feeling positive, neutral, or negative. For instance, words like “overwhelmed” or “anxious” trigger the system to recognize distress.
- **Tracking Emotional Trends**: By analysing conversations over time, the system identifies patterns in user moods, offering insights into their emotional well-being.
- **Personalized Interventions**: Based on sentiment scores, the system can offer specific suggestions, such as calming exercises, encouraging messages, or resource links.

1.8 Hybrid Recommendation System: Tailoring Suggestions

The AI Therapist provides users with personalized recommendations, such as articles, mindfulness exercises, or crisis helplines.

- **Content-Based Recommendations**: By analysing the content of resources and matching it with the user’s preferences or current emotional state, the system ensures relevance.
- **Collaborative Filtering**: Drawing from the experiences of other users with similar profiles, the system suggests resources that have been effective for others.
- **Hybrid Approach**: The combination of both methods ensures diversity and personalization, catering to individual needs while incorporating collective wisdom.

1.9 Flask Framework: Powering the Backend

Flask serves as the backbone of the system, managing the logic and ensuring smooth operation.

- **Integration Hub**: Flask acts as the bridge connecting the chatbot, database, and front- end interface, enabling seamless data flow between all components.
- **Gemini APIs**: Using Flask, the system provides APIs that manage chatbot conversations, deliver sentiment analysis results, and share personalized recommendations.
- **Scalability**: Flask's lightweight and flexible design, the system can easily grow to support new features or a larger user base without significant changes.

1.10 Conversational Model using Gemini API: The Heart of the System

At the core of the AI Therapist is its conversational model, designed to be a supportive virtual mental health companion.

- **Thoughtful Interaction:** Powered by the Gemini API, the model creates meaningful responds to each user's unique needs.
- **Proactive Engagement**: It doesn't just wait for questions; it starts conversations, encourages reflection, and offers timely emotional support.
- **Expanded capabilities**: With the Gemini API, the model connects to external resources like mindfulness apps and crisis hotline directories, making it even more helpful and versatile.

CHAPTER-2

LITERATURE SURVEY

[1] M.V. Patil, Subhawna, Priya Shree, Puneeth Singh, "AI BASED HEALTHCARE CHAT BOT SYSTEM"., July 2021

The paper "AI-based Healthcare Chatbot System" focuses on developing a chatbot to address limited access to medical care, particularly in rural and government hospitals. Using Artificial Intelligence and Natural Language Processing (NLP), the chatbot interacts with users to diagnose diseases based on symptoms and provides related information. Inspired by earlier works like ELIZA and advancements such as Flora Amato's use of Watson language services, the system employs Python libraries like Pandas, NumPy, and Scikit-learn for machine learning tasks, including decision tree classifiers for symptom-based diagnoses.
Unlike existing systems like "Your.MD," this chatbot uses a simple Yes/No interaction model, identifying diseases and listing related symptoms to offer a more comprehensive health assessment. The system aims to make healthcare more accessible and affordable while reducing dependency on immediate professional consultation. Future enhancements include personalized care through integration with patient histories and improved language processing for empathetic interactions. This work highlights AI's potential in revolutionizing healthcare accessibility.

[2] Lalith Abualigah, Hamza Essam Alfar, Mohammed Shehab and Alhareth Mohammed Abu Hussein, "SENTIMENT ANALYSIS IN HEALTH CARE"., January 2020., DOI:10.1007/978-3-030-34614-0_7

The paper *"Sentiment Analysis in Healthcare"* explores the use of sentiment analysis in deriving insights from patient-generated text data in healthcare. Sentiment analysis, a technique combining data mining and natural language processing (NLP), evaluates emotions and opinions expressed in text from social media, blogs, and healthcare forums. This enables the identification of trends in patient feedback, such as satisfaction with treatments, hospital services, or medication effectiveness, supporting improved healthcare delivery and decision-making.
The study highlights various sentiment analysis techniques, including lexicon-based approaches, supervised machine learning methods like Support Vector Machines and Naïve Bayes, and deep learning models such as Convolutional Neural Networks (CNNs) and Long

Short-Term Memory (LSTM) networks. These methods provide insights into patient experiences, allowing healthcare providers to enhance service quality and address patient concerns. However, challenges such as language diversity, sarcasm, and a lack of healthcare-specific datasets complicate the process.

Applications of sentiment analysis extend to monitoring public health trends, detecting adverse drug reactions, and understanding public perceptions during health crises. For example, analysing tweets during a pandemic can reveal real-time concerns and misinformation. Future advancements include developing healthcare-specific lexicons, addressing linguistic complexities, and integrating sentiment analysis with AI systems for personalized recommendations.

[3] Siddharth Biju, Dolly Golani, Aditya Khatavkar, Noopur Lade, and Pratush Jadoun., "THERAPISTGPT:AN AI POWERED THERAPIST"., May 2023

The paper *"TherapistGPT: An AI-Powered Therapist"* explores the development of an advanced conversational AI system designed to provide mental health support. Built using Natural Language Processing (NLP) and Machine Learning (ML), TherapistGPT addresses critical challenges such as the shortage of mental health professionals and high therapy costs, making mental health care more accessible and scalable. It employs large language models (LLMs) trained on a diverse corpus of therapy-related texts, enabling it to handle a wide range of mental health topics effectively. The system features personalized therapy sessions, 24/7 availability, multilingual support, and strict adherence to confidentiality standards.

TherapistGPT integrates evidence-based therapeutic techniques, including Cognitive Behavioural Therapy (CBT), Dialectical Behavioural Therapy (DBT), and Acceptance and Commitment Therapy (ACT). A robust dialogue management layer ensures coherence in interactions, while self-attention mechanisms in the transformer-based architecture enhance its ability to understand and respond contextually. Trials conducted to evaluate its performance showed comparable results to human therapists in patient satisfaction and symptom reduction, while outperforming other AI-based solutions in these metrics.

The paper emphasizes ethical considerations, including robust privacy protections through end- to-end encryption and compliance with regulations like GDPR and HIPAA. Bias mitigation strategies are implemented by training the AI on diverse datasets to ensure inclusivity and fairness. Future directions include integrating TherapistGPT with existing mental health care systems, expanding its scope to areas like substance abuse and family therapy, and enhancing its capabilities through multilingual support and continuous technological improvements.

TherapistGPT signifies a promising step in leveraging AI to democratize mental health care, offering a scalable, efficient, and effective solution that complements traditional therapy practices while addressing global mental health challenges.

[4] Riddhi Shetty, Ankita Bhosale, Pankaj Verma, Ashwini Phalke, "TITLE:AI-BASED HEALTHCARE CHATBOT"., May 2022

This paper explores various efforts to develop AI-based healthcare chatbots and their unique approaches. Md. Moshiur Rahman's chatbot used machine learning algorithms, with SVM performing the best, though it struggled with collecting detailed symptom data. Rohit Binu Mathew's system relied on KNN to match symptoms with diseases and recommend treatments but couldn't assess the severity of conditions.

Similarly, Divya Madhu focused on personalized medicine advice, while Chin-Yuan Huang developed a chatbot using Dialog flow to offer health tips like diet and exercise plans, though its high cost and inability to suggest doctors for severe cases were limitations.

Urmil Bharti's chatbot incorporated voice interaction and preventive health advice but faced similar cost challenges. Prakhar Srivastava combined multiple algorithms like KNN, SVM, and Naive Bayes to detect symptoms and assess disease severity effectively. Finally, Prathamesh Kandpal applied deep learning and natural language processing, though high costs and limited decision-making were hurdles.

Despite their differences, these projects highlight the potential of AI chatbots to make healthcare more accessible and efficient, while also pointing to challenges like cost and functionality that still need to be addressed.

[5] Mark Lawrence, MD Istiyak , Mohd Aman., "HEALTHCARE CHATBOT SYSTEM"., March 2024

This paper explores how healthcare chatbots are transforming medical services. Patil et al. developed chatbots to help people in rural areas or those struggling to access medical advice, while Athota et al. designed AI-powered chatbots capable of diagnosing illnesses and cutting healthcare costs. Liu et al. introduced a hybrid model using text similarity and knowledge graphs to tackle complex medical queries.

Hossain et al. created "MR. Dr.," a chatbot for private health consultations, and Shinde et al. focused on making diagnoses faster and more affordable. Kalakota highlighted the potential of chatbots for improving patient engagement and providing treatment suggestions. Denecke and Achananuparp emphasized the need for better data security and user trust, particularly for mental health applications. Meanwhile, Biju et al. used decision trees to predict diseases, demonstrating how chatbots can simulate real-life medical scenarios.

Overall, the survey shows the immense potential of healthcare chatbots to improve accessibility, reduce costs, and enhance efficiency. However, challenges like ensuring privacy, building user trust, and understanding complex user queries remain crucial areas for further development.

[6] Ayain John, Abhigna G, Adithi K V, Harusha R, Kavya A S ., "HEALTHCARE CHATBOT"., August 2022 ., DOI:10.26562/irjcs.2022.v0908.28

The paper highlights the growing role of healthcare chatbots in improving patient care and streamlining medical processes. Powered by AI and NLP, these chatbots allow users to interact in real-time, helping them identify symptoms, receive basic medical guidance, and even assist healthcare professionals by reducing administrative burdens.

Past research has shown chatbots' potential to enhance healthcare accessibility, offer virtual consultations, and provide personalized diagnoses. These systems use structured data and machine learning to deliver accurate predictions and support decision-making. Chatbots are also used in patient education and managing chronic conditions, making healthcare more proactive and accessible.

However, challenges remain, such as ensuring data privacy, handling varied user queries, and improving conversational accuracy. Despite these hurdles, healthcare chatbots show great promise in reducing wait times, improving efficiency, and delivering scalable, user-friendly solutions in the medical field.

[7] **Idayati Mazlan, Noraswaliza Abdullah, Norashikin Ahmad., "Exploring the Impact of Hybrid Recommender Systems on Personalized Mental Health Recommendations".,** **January 2023 ., DOI:**10.14569/IJACSA.2023.0140699.

This paper highlights the "Hybrid recommender systems" address the limitations of traditional methods, such as collaborative and content-based filtering, by integrating multiple techniques to improve accuracy, diversity, and personalization. Collaborative filtering predicts user preferences based on past behaviour but struggles with data sparsity, while content-based filtering matches resources to user attributes but often lacks diversity. Hybrid systems combine these strengths through strategies like weighted, mixed, and cascade models, delivering tailored and context-aware recommendations.

Key evaluation metrics include precision, recall, F1-score, and user satisfaction, assessed through offline (historical data) and online (real-time) methods. Despite their effectiveness, challenges such as data integration, algorithm fusion, and the cold-start problem persist. Addressing these requires leveraging diverse data sources (e.g., wearables, social media) and incorporating technologies like explainable AI to enhance transparency and trust.

Future research focuses on ethical considerations, real-time feedback, and advanced hybridization techniques to enhance recommendation quality. These systems show great potential to revolutionize mental health care by providing personalized, effective, and engaging support for users, ultimately improving outcomes and satisfaction.

[8] Eshan Shinde, Mahesh Shendage, Rahul Patil., "Artificial Intelligence Therapist"., August 2022

This paper outlines the development of an AI-based chatbot system designed to address mental health challenges by analyzing user emotions and offering personalized therapeutic recommendations. The system employs Deep Neural Networks, specifically Bi-LSTM and CNN models, to classify emotions into four categories: anger, fear, depression, and anxiety. Using Natural Language Processing (NLP), it extracts emotional cues from user inputs, while algorithms like Naive Bayes and Collaborative Filtering enhance classification accuracy. A recommender module draws from a database of therapeutic resources, including motivational content and stories, to deliver tailored support. Built with Python, the system integrates Flask for backend development, the Chatterbot library for dynamic conversational responses, and web technologies like HTML and CSS for a user-friendly interface. Iterative training improved the chatbot's response accuracy and processing time, enabling it to provide empathetic and contextually relevant support. This scalable solution bridges gaps in mental health care by offering accessible, AI-driven assistance while reducing reliance on human professionals. Future enhancements will focus on refining its therapeutic capabilities and response precision.

[9] Prof. Mamta Madan, Ms. Rishima Madan, Dr Praveen Thakur., “Analyzing the patient sentiments in healthcare domain using Machine learning”., April 2024

This paper, *"Analysing the Patient Sentiments in Healthcare Domain Using Machine Learning,"* explores how machine learning can help analyse patient feedback to evaluate healthcare facilities. By focusing on aspects like cleanliness, doctor availability, and patient-doctor interactions, the study uses Python's Text Blob library to identify the sentiment (positive, negative, or neutral) in patient reviews. It also calculates a "goodness score" for each healthcare facility, helping patients make informed decisions about where to seek care.

The authors review existing research on sentiment analysis in healthcare, including its use in analysing social media posts, monitoring mental health, and classifying electronic health records. They highlight the challenges of interpreting patient emotions and the need for better tools customized for healthcare data.

This study stands out by combining sentiment analysis with healthcare quality evaluation, paving the way for smarter, AI-driven tools to improve patient experiences. The authors suggest that future work could include advanced techniques like genetic algorithms and ensemble learning to enhance accuracy and overcome current limitations.

[10] Ameen Abdullah Qaid Aqlan, B. Manjula and R. Lakshman Naik., “A Study of Sentiment Analysis: Concepts, Techniques, and Challenges”., January 2019
DOI: 10.1007/978-981-13-6459-4_16

This paper explores sentiment analysis (SA), a powerful tool for understanding emotions in text data from sources like social media, blogs, and customer reviews. It outlines the five stages of SA: data collection, text preprocessing (using techniques like tokenization, stemming, and NLP), sentiment detection, classification, and output presentation. The study highlights classification methods, including lexicon-based approaches, machine learning algorithms (e.g., Naïve Bayes, SVM), and hybrid models, alongside feature extraction techniques like TF-IDF and Word2Vec. It also addresses the use of big data technologies, such as Hadoop and MapReduce, to manage large datasets in real time. By emphasizing model tuning, cross-validation, and evaluation metrics like accuracy and F1- score, the paper underscores the importance of refining

SA models. Applications span business intelligence, political trend forecasting, and social media monitoring, demonstrating SA's critical role in decision-making and predictive analytics.

[11] Eshan Shinde, Mahesh Shendage, Rahul Patil., "ARTIFICIAL INTELLIGENCE THERAPIST"., August 2022

This paper outlines the development of an AI-based chatbot system designed to address mental health challenges by analyzing user emotions and offering personalized therapeutic recommendations. The system employs Deep Neural Networks, specifically Bi-LSTM and CNN models, to classify emotions into four categories: anger, fear, depression, and anxiety. Using Natural Language Processing (NLP), it extracts emotional cues from user inputs, while algorithms like Naive Bayes and Collaborative Filtering enhance classification accuracy. A recommender module draws from a database of therapeutic resources, including motivational content and stories, to deliver tailored support. Built with Python, the system integrates Flask for backend development, the Chatterbot library for dynamic conversational responses, and web technologies like HTML and CSS for a user- friendly interface. Iterative training improved the chatbot's response accuracy and processing time, enabling it to provide empathetic and contextually relevant support. This scalable solution bridges gaps in mental health care by offering accessible, AI-driven assistance while reducing reliance on human professionals. Future enhancements will focus on refining its therapeutic capabilities and response precision.

[12] David B. Olawade, Ojima Z. Wada , Aderonke Odetayo Aanuoluwapo Clement David-Olawade, Fiyinfoluwa Asaolu , Judith Eberhardt ., "ARTIFICIAL INTELLIGENCE THERAPIST"., November 2023

This paper reviews the integration of Artificial Intelligence (AI) in mental healthcare, focusing on its applications, challenges, and future directions. It highlights how AI, combined with wearable technologies and mobile health apps, enables real-time data collection on behaviors such as sleep patterns and physical activity, improving risk predictions and personalized treatment plans. AI is also transforming personalized care by analyzing genetic profiles, behavioral patterns, and treatment histories to tailor interventions like medication and therapy. Virtual therapists and AI-driven chatbots are increasingly used to provide continuous mental health support, particularly in underserved regions.This paper underscores the importance of ethical considerations, including privacy, data security, and addressing biases in AI algorithms, to ensure equitable treatment outcomes. It stresses that AI should complement human therapists rather than replace them, maintaining a balance between technology and human interaction.

[13] Erion C, Maurizio Morisio ., "Hybrid Recommender Systems: A Systematic Literature Review" ., November 2017

This paper explores the generation and visualization of personalized explanations for hybrid recommender systems, focusing on enhancing recommendation quality through a hybrid probabilistic graphical model. Using the last.fm music platform for a user study, the research evaluates various explanation styles (user-based, item-based), their presentation formats (textual vs. visual), and the influence of user personality traits on preferences. A mixed-model statistical analysis is employed to assess the effectiveness of different explanations.

Key findings reveal that users generally prefer item-based and content-based explanations, with calmer users favoring popularity-based explanations and anxious users leaning toward item-based collaborative filtering. While users typically prefer three to four explanation styles, some are open to receiving no explanation at all. Interestingly, textual explanations are preferred over visual formats like Venn diagrams, contrasting previous research emphasizing the appeal of visual explanations.

The study underscores the importance of personalized explanations in hybrid recommender systems for improving user engagement and recommendation persuasiveness. It highlights the trade-offs between explanation styles and presentation formats, noting that preferences depend on personality traits and context. The hybrid model outperforms traditional collaborative filtering methods, demonstrating its potential for better recommendations.

[14] Pigi Kouki, James Schaffer, Jay Pujara, John O'Donovan, Lise Getoor ., "Hybrid Recommender Systems: A Systematic Literature Review" ., March 2019

The paper explores the generation and visualization of personalized explanations for hybrid recommender systems, which combine various data sources to improve recommendation quality. The authors focus on using a hybrid probabilistic graphical model to provide real-time recommendations and personalized explanations, using the last.fm music platform for their study. The research examines different explanation styles, such as user-based and item-based, and tests various presentation formats (textual vs. visual). Through a mixed-model statistical analysis, the study incorporates user personality traits to evaluate the effectiveness of different explanation types. The findings show that people generally prefer item-based explanations over user-based ones, with content-based explanations being the most persuasive. Calmer users tend to favor popularity-based explanations, while anxious individuals prefer item-based collaborative filtering explanations. On average, users liked having three to four explanation styles, although some were open to not receiving any explanation. The study also found that textual explanations were more popular than visual formats, such as Venn diagrams, despite users being familiar with visualization tools. From a technical perspective, the research emphasizes the role of personalized explanations in hybrid recommender systems, enhancing user engagement and the persuasiveness of recommendations. It also highlights the trade-offs between explanation styles and presentation formats, noting that user preferences depend on personality traits and the context of the system's use..

[15] Mehdi Elahi , Danial Khosh Kholgh , Mohammad Sina Kiarostami , Mourad Oussalah , Sorush Saghari ., "Hybrid recommendation by incorporating the sentiment of product reviews" ., January 2023

This study evaluates sentiment-aware hybrid recommender systems using Amazon Digital Music and Video Games datasets. Experiment A revealed distinct distribution patterns, with Digital Music reviews showing a half-normal distribution with positive sentiment, while Video Games had a normal distribution centered around mid-range sentiment and ratings. Experiment B found a stronger correlation between ratings and sentiment in Video Games (0.648) compared to Digital Music (0.284), with BERT performing best for Video Games and TextBlob for Digital Music. Experiment C showed improved recommendation accuracy in Digital Music, especially with hybrid models like YoutubeRanker and DeepFM. Experiment D demonstrated that incorporating sentiment scores into recommendation algorithms significantly boosted precision, with YoutubeRanker-S showing a 160% improvement in the Video Games dataset. In

Experiment E, YoutubeRanker-S outperformed other models with rating-based feedback, while ESCOFILTS excelled with sentiment data. The study concludes that sentiment data enhances recommendation quality and captures different aspects of user preferences. Future work will explore the impact of sentiment-aware techniques on larger datasets and real-time user interactions.

[16] P. Chinnasamy Ajmeera Kiran , Wing-Keung Wong , J. Chinna Babu , Ambeth Raja , Osamah Ibrahim Khalaf ., "Health Recommendation System using Deep Learning-based Collaborative Filtering"., December 2023 ., DOI: 10.1016/j.heliyon.2023.e22844

The Health Recommendation System (HRS) leverages advanced deep learning techniques, combining the Restricted Boltzmann Machine (RBM) with the Coevolutionary Neural Network (CNN), to deliver personalized healthcare recommendations. The system analyzes a range of health- related data, including structured data like lab results, semi-structured data like medical prescriptions, and unstructured data like doctor's notes. By integrating RBM for feature extraction and CNN for processing spatial data, such as medical images, the system captures relationships within various data types, providing recommendations for diagnoses, treatments, therapies, and lifestyle changes. The process includes several phases: data preprocessing to cleanse and structure health information, patient profile creation to ensure up-to-date records, sentiment analysis to respect patient comfort, and recommendation generation based on individual health data. Security measures like differential privacy and encryption ensure patient data remains protected. The system's performance, evaluated through metrics like accuracy, precision, and recall, has shown significant improvements over existing systems. Additionally, MapReduce-based Hadoop is used for efficient data processing, allowing the system to scale as healthcare data grows. Future developments will focus on enhancing data privacy, ethics, and transparency, while integrating with telehealth platforms and improving AI explainability to offer more personalized care.

CHAPTER-3

RESEARCH GAPS OF EXISTING METHOD.

3.1 Front-End Design: Limited User Experience.

Most AI mental health platforms don't offer highly personalized or inclusive user interfaces. For instance, features like accessibility for visually impaired users or culturally relevant designs are often missing. Additionally, many systems fail to provide a smooth experience across devices, leading to frustration for users switching between phones, tablets, or computers.

We can create an interface that adapts to individual preferences, offering features like voice input multilingual support, and even mood-tracking visuals. By making the platform accessible and user-friendly, we aim to ensure everyone feels comfortable using it, regardless of their technical skills or needs.

3.2. Fire store Database: Challenges with Real-Time Data and Privacy.

As user numbers grow, databases often slow down, making interactions feel sluggish. On top of that, not all platforms fully address privacy regulations like GDPR or HIPAA, leaving sensitive mental health data vulnerable. By optimizing Firestore's real-time capabilities, we can handle large user volumes while ensuring quick and seamless data updates. Additionally, implementing top-tier encryption and compliance measures will ensure users' data is safe and secure.

3.3 Natural Language Processing (NLP): Struggling to Understand Users

Current AI models often miss the mark when it comes to understanding complex emotions or maintaining long conversations. For example, if someone says, "I'm feeling happy but also a little nervous about tomorrow," most systems can't process both emotions simultaneously. By fine-tuning advanced NLP models like GPT or BERT, we can teach the AI to better understand nuanced emotions and context. This means the AI will remember earlier parts of the conversation, making its responses more coherent and personalized.

3.4 Sentiment Analysis: Surface-Level Understanding of Emotion:

Most systems categorize emotions into simple buckets like positive, negative, or neutral. But human emotions are far more complex and layered than that. Also, many platforms don't track how person's emotions evolve over time, missing key mental health patterns.

We aim to create more detailed sentiment analysis tools that can detect subtle emotional shifts, like distinguishing between frustration and sadness. By tracking emotions over weeks or months, the AI can help users see patterns in their mental health and provide more tailored support.

3.5 Hybrid Recommendation System: One-Size-Fits-All Suggestions.

Existing recommendation systems often struggle with "cold starts," where new users receive generic suggestions because the AI doesn't know enough about them yet. Recommendations can also feel repetitive or disconnected from the user's emotional state.

By combining user preferences with real-time sentiment analysis, our hybrid system can provide dynamic, context-aware recommendations. Even new users will get meaningful suggestions based on initial inputs, such as their emotional state or activity patterns. Flask-based platforms sometimes slow down when handling a large number of requests. Additionally, integrating third-party tools like external AI models or APIs can be unnecessarily complex.

By implementing advanced features like asynchronous request handling, we can make the system faster and more efficient. Streamlining API integrations will also make it easier to introduce new features in the future, ensuring the platform evolves with user needs.

3.6 Conversational Model: Conversations That Feel Robotic

A lot of chatbots still feel mechanical because they lack the ability to express empathy or adjust to individual conversational styles. Additionally, many systems aren't prepared to handle critical situations, like a user expressing suicidal thoughts.

We can train the chatbot on empathy-focused datasets, enabling it to respond in a way that feels human and supportive. By integrating crisis detection mechanisms, the system can recognize when a user needs urgent help and provide resources or escalate to human intervention.

3.7 Conversational model Implementation: Static and Reactive Interactions

Most chatbots today are reactive—they only respond when you ask questions or express how you're feeling, often following a set script. This can make conversations feel repetitive and less personal.

To improve this, we aim to develop a conversational model powered by the Gemini API that is not only reactive but also proactive and context-aware. For example, if the system detects that you've been feeling low over a few days, it can reach out to check in on you or suggest activities to help boost your mood. By continuously learning from your conversational style and adapting to your needs, the chatbot can offer a much more engaging, natural experience. Current systems often fail to truly connect with users, leaving people feeling like they're not being heard. For someone seeking mental health support, this can be especially frustrating. By using the Gemini API, our conversational model can bridge those gaps, offering a deeply human approach to care. It's not just about responding with a fixed answer—it's about providing thoughtful, personalized support that makes you feel understood.

This approach isn't just about improving technology; it's about creating a platform that genuinely listens and supports you, providing a meaningful connection when it matters most. By leveraging Gemini's advanced conversational abilities, we can create a conversational model that truly feels like a helpful companion on your mental health journey.

CHAPTER-4

PROPOSED METHODOLOGY

The AI therapist system combines sentiment analysis, hybrid recommendation algorithms, and natural language generation, in a single integrated solution that supports personalized, empathetic mental health care.

4.1 Sentiment Analysis

The sentiment analysis module works like a behavior analyst, detecting emotions such as sadness, anxiety, depression, and fear from your input text. By picking up on emotional cues like tone and word choice, it understands how you're feeling.

Once your emotional state is identified, the system customizes its response. For example, if it detects sadness, it might offer comforting words or suggest mood-lifting activities. If anxiety or fear is noticed, it could recommend calming techniques. This allows the chatbot to respond in a way that feels more empathetic and supportive, helping you feel understood and cared for.

4.2 Dataset Collection and Preprocessing

To train an emotionally aware chatbot, we collect a dataset of text samples labeled with emotions like sadness, anxiety, or joy. These texts are sourced from publicly available archives, where each entry includes the emotion it expresses.

The dataset goes through preprocessing, which involves cleaning and standardizing the text to make it easier for the system to analyze. It is then divided into three sets: training (for learning), validation (for fine-tuning), and testing (to assess performance). This process helps the chatbot recognize emotional cues and respond appropriately to users' feelings.

4.3 Preprocessing Techniques

Preprocessing ensures text data is clean and consistent. We remove non-alphabetical characters using regular expressions and convert all text to lowercase to eliminate case sensitivity. Then, we tokenize the text and remove stopwords using NLTK’s list, focusing on meaningful words.

Hyphenated words and variations are reduced to their root forms using optional stemming with the PorterStemmer. We also introduce a *textlength* feature to analyze text length and identify outliers in communication patterns.

4.4 Feature Extraction

Text data is converted into numerical representations using:

- Count Vectorizer: The authors build the term frequency from the test and the bag of words representation. Size 3,000 maximum features are selected for computer power.
- TF-IDF Vectorizer (for comparison): In this procedure, term frequency and inverse document frequency are not only weighted, but also considered when considering the word's relevance in the given corpus of texts.
- Machine Learning Models: In the system, machine learning models are employed to classify emotions. Each model was trained and tested to train and test the following:
- Models Used: Random Forest Classifier, Hyperparameters, Support Vector Machine (SVM), Logistic Regression, Multinomial Naive Bayes (MNB), Gradient Boosting Classifier.

4.5 Model Selection and Evaluation

The Voting Classifier is employed to fuse results of the most successful models, namely, RFC, Logistic Regression, Gradient Boosting, and SVM, to obtain higher prediction accuracy.

The evaluation metrics of the models used are:

Accuracy Score: This is the percentage of correct predictions.

Confusion Matrix: A visualization of true vs. predicted labels.

Classification Report: Precision, recall, and F1-score

4.6 Hyperparameter Tuning

To find the best-performing models, we start by manually tuning the hyperparameters, adjusting key settings to improve performance. After that, we use a more automated approach, GridSearchCV, to fine-tune these parameters and find the most optimal values. This process ensures the model is working at its best potential.

For example, in the case of the Support Vector Machine (SVM), we focus on optimizing two critical parameters: *C* and *gamma*. *C* controls the trade-off between achieving a low error on the training data and maximizing the margin between classes, while *gamma* defines how far the influence of a single training example reaches. By adjusting these values, we aim to find the ideal balance for accurate predictions.

For the Random Forest Classifier, we fine-tune parameters like the number of *estimators* (the number of trees in the forest) and *max depth* (the maximum depth of each tree). These adjustments help balance the model's complexity and its ability to generalize, improving its accuracy and robustness in making predictions.

4.7 Experimenting Workflow

A pipeline is structured in such a way to automatically accomplish the end-to-end experiments which are composed of:

Text Preprocessing Module: Performs data cleaning and feature engineering.

Feature Extraction Module: Converts the text to numerical formats with CountVectorizer

Model Training and Validation Pipeline: Learned the models from the preprocessed dataset and tested with validation data.

Voting Classifier Integration: Combined predictions from all classifiers to generate better results.

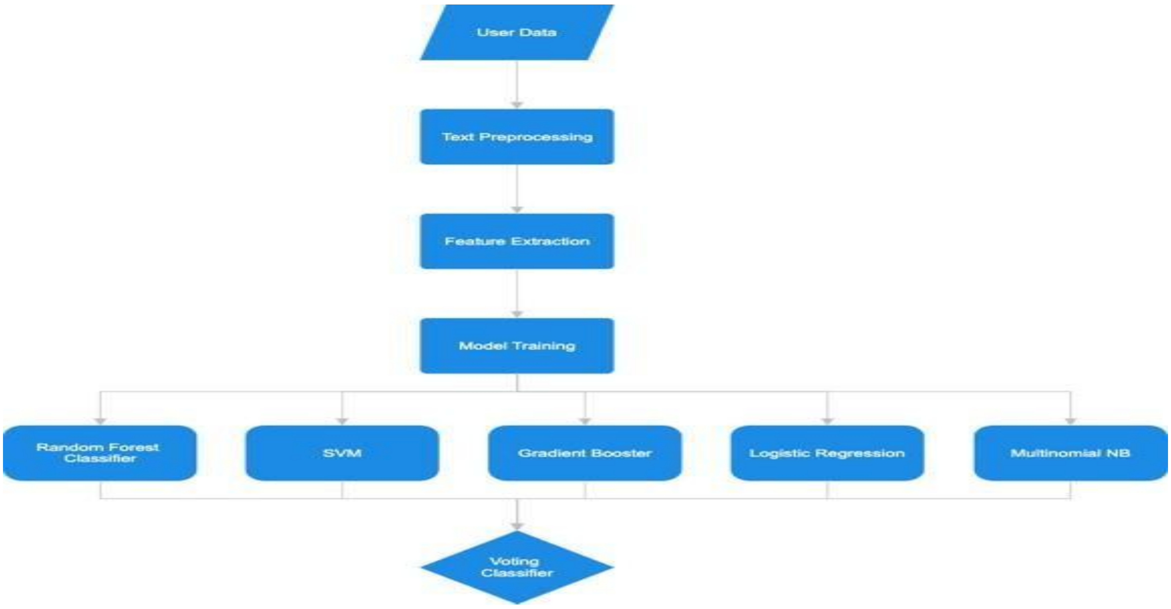


Figure 1.1

4.8 Visualization and Analysis

To comprehend the performance of the system in detail:

- Confusion Matrices for train, validation, and test set are visualized using Seaborn heatmaps. Bar plots of model accuracy scores indicated which method performed best.
- Representation and balance of emotional distributions in various datasets were also compared.

4.9 Custom Predictions

This methodology is to truly see how well our model works in real-life situations, we use *custom predictions*. This means feeding the system with new text—representing specific emotions—that it hasn't encountered during training. The model then predicts the emotion behind the text.After the prediction, we check how accurate the model is by comparing its response with what we expected. But it's not just about getting the right answer; we also look at how the model interprets the text and why it made that prediction. This helps us ensure that the system's decisions are both accurate and easy to understand, giving us confidence that it can handle real-world conversations effectively and meaningfully.

CHAPTER-5

OBJECTIVES

5.1 Create a Comprehensive and Accessible Mental Health Platform.

Design a web-based application that enables users to express their mental health concerns easily, fostering a safe, interactive, and inclusive space for all demographics. The application should prioritize accessibility, ensuring users from diverse technical backgrounds can engage effectively.

5.2 Advanced Emotion Detection Using NLP

Leverage state-of-the-art Natural Language Processing (NLP) techniques like feature extraction (e.g., TF-IDF, word embeddings) and pre-trained language models (e.g., BERT) to analyze user input. Accurately classify emotional states such as fear, anxiety, and depression based on linguistic patterns and sentiment analysis.

5.3 Develop a Personalized Hybrid Recommendation System

Create a recommendation engine that combines content-based filtering (using user profiles and emotional states) and collaborative filtering (leveraging collective user behavior) to suggest practical solutions. These include motivational quotes, relaxation exercises, therapeutic activities, and other mental health resources tailored to individual needs.

5.4 Implement Real-Time Data Management

Use Firestore database to store and manage user data securely, ensuring quick and synchronized real-time access. Focus on efficient storage, retrieval, and data. consistency across multiple sessions to provide a seamless user experience.

5.5 Design an Interactive and Intuitive Frontend

Build a clean, responsive, and user-friendly interface using HTML, CSS, and JavaScript. The design should focus on usability and aesthetics, providing users with smooth navigation and visual elements that promote calmness and engagement.

5.6 Develop a Robust and Scalable Backend

Create a backend system that integrates NLP models, feature extraction techniques, and recommendation algorithms. Ensure the system can handle a growing user base while maintaining consistent performance and fast response times.

5.7 Promote Mental Health Awareness Through Personalized Solutions

Provide actionable insights and tailored recommendations to users, such as mindfulness exercises, self-care routines, and motivational resources, that can be integrated into their daily lives. Ensure these recommendations are evidence-based and aligned with mental health best practices.

5.8 Focus on Data Privacy and Security

Prioritize user data confidentiality and protection by implementing encryption, secure authentication mechanisms, and compliance with data privacy regulations such as GDPR.

5.9 Conduct Rigorous Testing and Iterative Refinement

Perform integration testing, user acceptance testing, and stress testing to ensure all components function harmoniously. Gather feedback from target users to identify issues, refine recommendations, and improve user satisfaction.

5.10 Deploy the System and Support User Adoption

Launch the application with a focus on scalability, user onboarding, and ease of adoption. Provide resources such as tutorials and FAQs to guide users through the system, and set up a feedback mechanism to continually improve the platform.

5.11 Promote Well-Being and Reduce Mental Health Stigma

Aim to empower users by addressing mental health concerns in a stigma-free environment. The platform should help users feel supported and motivated to adopt healthier mental health practices.

5.12 Leverage User Feedback for Continuous Improvement

Incorporate insights from user interactions and feedback to enhance the system's accuracy and personalization over time. This iterative process ensures the system evolves to meet changing user needs effectively.

5.13 Empower Self-Reflection Through AI

Encourage users to self-reflect on their emotional state by providing instant, meaningful feedback on their input, helping them gain better awareness of their mental health.

5.14 Integrate Diverse Feature Extraction Techniques

Use multiple feature extraction methods such as Bag of Words (BoW), TF-IDF, and word embeddings (e.g., GloVe, Word2Vec) to improve the accuracy of emotion detection and enable the system to interpret complex, nuanced user inputs

CHAPTER-6

SYSTEM DESIGN & IMPLEMENTATION

6.1 System Design

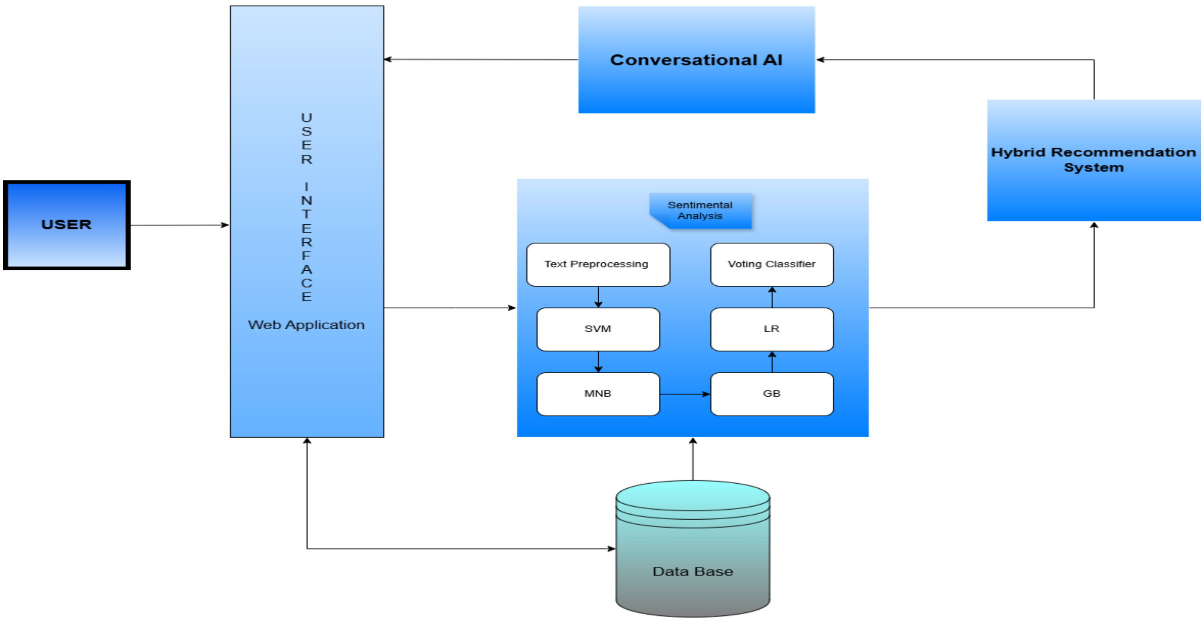


Figure 6.1.1 System Architecture.

6.2 System Overview

The AI Therapist is structured around a three-tier architecture that separates the user interface, processing logic, and data storage. This design ensures scalability, modularity, and ease of maintenance.

1. **Presentation Layer (Frontend)**:
   - Handles user interaction, including registration, login, problem submission, and results display.
   - Built with HTML, CSS, and JavaScript for a responsive and interactive user experience.
2. **Logic Layer (Backend)**:
   - Processes user input, performs NLP tasks, and classifies emotions using machine learning models.
   - Interacts with the database to retrieve user information and store session data.

- Generates personalized recommendations based on classified emotions and integrates conversational AI for a rich user experience.

3. **Data Layer (Database)**:
   - Manages user data securely, including login credentials and session logs.
   - Firebase Firestore is used for real-time data synchronization, ensuring the application remains responsive and up-to-date.

6.3 Workflow

The system operates through a structured workflow:

1. User Authentication: Users register or log in using email and password, verified by Firebase Authentication.
2. Problem Submission: Users describe their emotions or problems in a dialog box on the web interface.
3. NLP Processing: The backend processes the input text to remove irrelevant data and extract meaningful features.
4. Emotion Classification: A voting-based ensemble of machine learning models categorizes the input into
   one of five emotional states: fear, sadness, anxiety, anger, or depression.
5. Recommendations: Based on the classified emotion, the hybrid recommendation system generates
   actionable advice tailored to the user's mental health needs.
6. Conversational AI: Engages the user with follow-up interactions to provide additional guidance or
   collect further input.
7. Results Display: The frontend presents the classified emotion and recommendations to the user in a
   visually appealing manner.

6.4 System Components

Frontend

The frontend forms the user-facing aspect of the application. It is designed to be simple yet interactive, ensuring that users can navigate easily without requiring technical expertise.

Key features include:

- User Registration and Login
- Provides a secure mechanism for users to register using email and password.
- Implements validation checks for proper credential format.

Input Dialog:

- A text box allows users to describe their issues, which is subsequently sent to the
- backend for processing.

Dynamic Result Display:

- Outputs classified emotions and personalized recommendations in a clean

The frontend uses HTML for structure, CSS for styling, and JavaScript for interactivity.

Together, they create a seamless interface that enhances user experience.

Backend

The backend is the engine that powers the AI Therapist's functionality. Its responsibilities include:

1. **User Authentication**:

- Uses Firebase Authentication to validate and manage user credentials.
- Ensures that passwords are hashed and securely store

2. NLP Pipeline:

- Processes user inputs to prepare them for analysis.
- **Tokenization**: Breaking down text into individual words or phrases.
- **Stop Word Removal**: Filtering out common words that add no analytical value.
- **Lemmatization**: Reducing words to their base forms for consistency.

3. Emotion Classification:

Implements a voting classifier, which combines multiple machine learning models to improve accuracy.

Models include:

- Support Vector Machines (SVM)
- Random Forest
- Multinomial Naive Bayes
- Logistic Regression
- Linear Regression

Hybrid Recommendation System

The recommendation system provides tailored advice based on the user's classified emotion. It combines:

1. Content-Based Recommendations:

Uses predefined remedies for each emotion category. For example:

Anxiety: Mindfulness exercises and breathing techniques.

Depression: Journaling and physical activities.

2. Collaborative Filtering (Future Scope):

Could be implemented to refine recommendations based on user behavior and preferences.The system ensures that recommendations are practical, evidence-based, and easy for users to follow.

Conversational AI

Enhances the system by simulating real-time interactions with users. This feature:

- Engages users with follow-up questions or additional guidance.
- Responds dynamically based on user input, creating a more personalized experience.
- Provides continuity in sessions, ensuring that users feel supported throughout their interaction with the system.

Database

Firebase Firestore is the database of choice due to its scalability and real-time synchronization capabilities. It stores:

User Credentials:

- Maintains email and hashed passwords securely.

Session Data:

- Logs user inputs and system responses for future reference and system improvement.
- Firestore's flexibility ensures that the system can scale efficiently as the user base grows.

6.5 System Implementation

Frontend Implementation

The frontend is implemented to provide a user-friendly interface. Key considerations include:

- Ensuring responsive design for compatibility across devices.
- Implementing clear navigation paths for ease of use.
- Handling user inputs securely and efficiently.

Backend Implementation

The backend processes user inputs using Python and machine learning libraries. GEMINIAPIs facilitate communication between the frontend and backend, ensuring that the system remains modular and scalable.

Emotion Classification

The emotion classification module is trained on labelled datasets of emotional text. The ensemble voting classifier achieves high accuracy by combining the predictions of individual models.

Recommendation System

Recommendations are predefined based on the best practices in mental health. Each emotion has a tailored set of remedies that the system dynamically presents to the user.

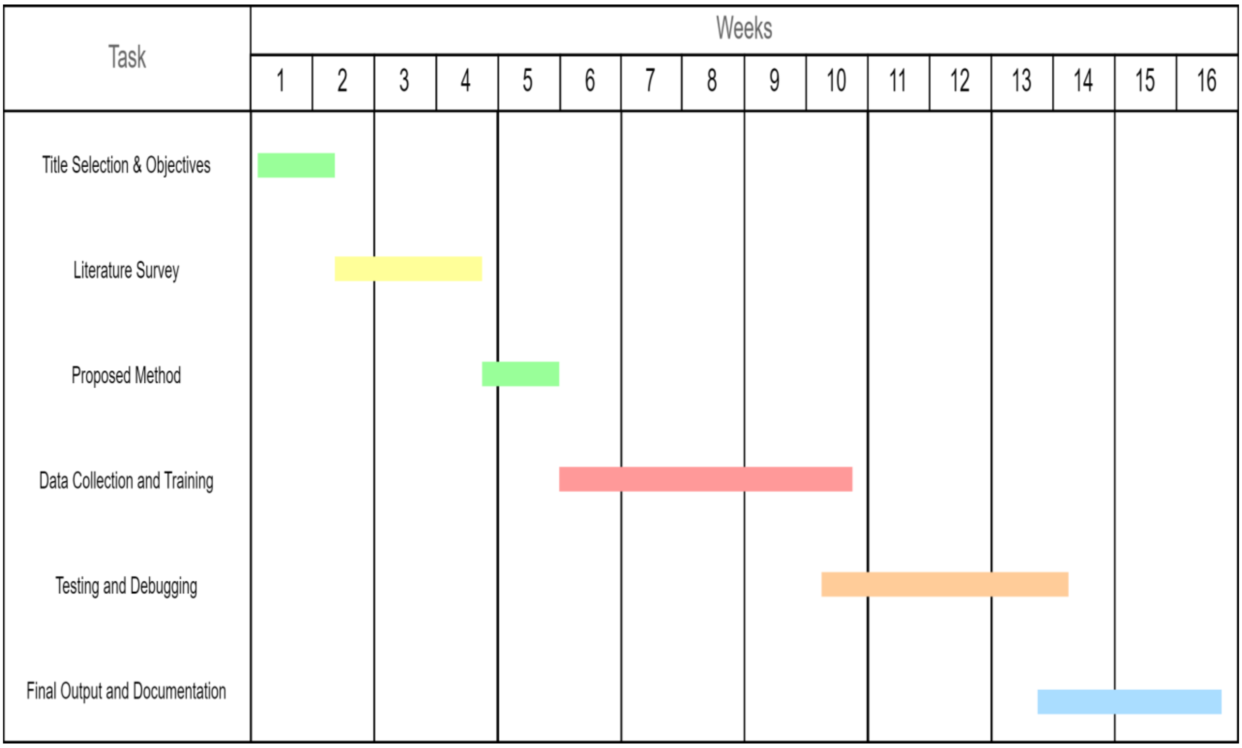
Conversational AI

Conversational AI is implemented using rule -based or transformer-based approaches , providing an engaging and natural user experience.

CHAPTER-7

TIMELINE FOR EXECUTION OF PROJECT

(GANTT CHART)



CHAPTER-8
OUTCOMES

8.1 Increased Access to Mental Health

The AI Therapist offers an affordable and accessible platform for those who may not be able to afford traditional therapy due to financial constraints, geographical limitations, or social stigma.

8.2 Personalized Support

The hybrid recommendation system of the project delivers tailored coping strategies, therapy suggestions, and resources that meet the unique emotional needs and preferences of users.

8.3 Enhanced User Experience

The conversational AI system enables empathetic and human-like conversations that make the user feel heard and cared for in a safe and judgment-free environment.

8.4 Higher Resilience to Emotional Bursts

The system helps the user develop resilience against the challenges of life through the effective provision of mindfulness exercise, stress management, and positive reinforcement tools.

8.5 Innovative Use of Technology in Healthcare

This project will show the potential for integrating AI, NLP, and recommendation systems in overcoming crucial health challenges, making it a benchmark for further technological advancements in the field of mental health.

8.6 Cost-Effective Solution

The AI Therapist is a low-cost mental health care system because it automates most of the aspects that characterize traditional therapy.

8.7 Continuous learning and adaptation:

With each interaction, the system learns from user feedback and improves its recommendations, making the quality of support better over time.

CHAPTER-9

RESULTS AND DISCUSSIONS

9.1 Frontend Development:

Results:

The frontend was developed using Node to create a dynamic, responsive interface that adapts to both mobile and desktop devices. Key features such as a mood tracker, chat interface, and personalized recommendations were implemented. The interface is designed to be intuitive, ensuring that users, regardless of their technical skills, can easily interact with the platform. Accessibility features such as voice support and language options were incorporated to ensure inclusivity.

Discussions:

- User Experience: The interface is user-friendly, with simple navigation that encourages prolonged engagement. The integration of a mood tracker and real-time chat feature enhances user interaction.
- Accessibility: Implementing features like voice support and multilingual options ensures that users from different backgrounds can benefit from the platform, which is crucial for a mental health-focused application.

9.2 Firestore Database:

Results:

The Firestore database was used to store user data in real-time, including interaction logs, emotional trends, and recommendations. The database facilitated seamless synchronization between different user sessions, ensuring that data was consistently available for analysis and recommendation generation.

Discussions:

- Real-time Synchronization: Firestore’s real-time data sync ensured that users received immediate feedback on their emotional state and recommendations. This was crucial for maintaining a continuous and personalized experience.
- Scalability: Firestore’s ability to scale with user growth makes it an ideal choice for this application, as it can handle large amounts of data and concurrent interactions without significant latency.

9.3 Natural Language Processing (NLP):

Results:

NLP models such as BERT and GPT were fine-tuned for mental health-related conversations. The AI demonstrated the ability to understand and generate contextually relevant and empathetic responses. NLP's integration allowed the system to recognize emotional cues and adapt responses to fit the user's current emotional state.

Discussions:

- Contextual Understanding: The ability of the NLP models to maintain context in multi- turn dialogues improved the coherence and relevance of responses. The system was able to handle diverse topics and sustain meaningful conversations.
- Empathy in Responses: While the models exhibited a good understanding of basic emotional cues, there was a noticeable gap in their ability to fully capture complex emotional subtleties, which will require further fine-tuning and training.

9.4 Sentiment Analysis:

Results:

The sentiment analysis model successfully detected emotions such as happiness, sadness, anxiety, and stress in real-time, based on user inputs. The AI adapted its responses depending on the detected sentiment, offering reassurance or suggesting helpful resources when negative emotions were identified.

Discussions:

- Accuracy: The sentiment analysis model performed well in detecting general emotions. However, challenges arose when dealing with ambiguous or mixed sentiments (e.g., a user expressing both frustration and hope), which affected the accuracy of responses.
- Personalization: The real-time sentiment detection allowed for personalized conversations, making the AI appear more empathetic and responsive to users' emotional needs. There is room to improve how it handles nuanced or complex emotional situations.

9.5 Hybrid Recommendation System:

Results:

The hybrid recommendation system, combining content-based filtering (emotional state) and collaborative filtering (user behavior), provided personalized suggestions for activities like meditation exercises, articles, and coping mechanisms. Recommendations evolved as the system tracked emotional trends and user preferences over time.

Discussions:

- Effectiveness: The hybrid recommendation system was effective in providing personalized content. Users reported finding the suggestions useful, particularly when the system recognized shifts in their emotional state.
- Room for Improvement: While the system performed well, there was a challenge in balancing the weighting between emotional state and user behavior. Future improvements could involve refining the algorithm to better prioritize emotional states when making recommendations.

9.6 Flask Framework Backend:

Results:

The Flask framework served as the backbone of the AI Therapist platform, handling user authentication, API requests, and interaction processing. The system was robust, ensuring smooth communication between the frontend, Firestore, and NLP models.

Discussions:

- Security: Flask, along with security tools like Flask-Security and Flask-WTF, kept user data safe from common threats like SQL injection and CSRF attacks. This gave us peace of mind knowing that user privacy and security were well-protected.
- Scalability: The backend performed well under moderate traffic loads, but as the user base grows, it will require further optimization to handle a larger number of concurrent interactions. Implementing load balancing and caching mechanisms could address this challenge.

9.7 Conversational Model:

Results:

The conversational model, built on top of the NLP architecture, demonstrated the ability to engage in empathetic, human-like dialogues. It successfully handled various conversation types, from casual check-ins to more serious emotional discussions, and responded appropriately based on user input.

Discussions:

- Engagement: The conversational model was engaging and supportive, fostering a sense of trust and openness. It was particularly effective at helping users express their emotions and receive immediate feedback.
- Emotional Depth: While the AI showed strong performance in handling general emotions, its responses could lack depth in certain situations, especially when dealing with deep, unresolved issues. This highlights the importance of continuous improvement in AI models for mental health contexts.

9.8 Conversational model Implementation:

Results:

The chatbot, powered by the conversational model, provided real-time interactions and emotional support. Users were able to talk to the chatbot about their feelings, and it responded with compassion and relevant resources, offering personalized coping mechanisms based on detected emotions.

Discussions:

- 24/7 Availability: The chatbot's round-the-clock availability was highly appreciated by users, particularly those who may not have access to in-person support.
- Limitations: While the chatbot was capable of empathetic conversations, it still faced challenges in managing sensitive issues like severe anxiety or suicidal thoughts. A more advanced escalation protocol, connecting users to human professionals or emergency services, would further enhance its effectiveness.

CHAPTER-10

CONCLUSION

AI Therapist project aimed to bridge the gap between technology and empathy. By weaving together cutting-edge technologies like Natural Language Processing (NLP), sentiment analysis, and a hybrid recommendation system, the platform brought a compassionate and tailored approach to supporting individuals.

10.1. Natural Language Processing (NLP):

Conversations are at the heart of emotional support. To make interactions meaningful, the AI Therapist relied on advanced NLP models capable of understanding not just words but the intent and context behind them. This allowed the system to respond with human-like sensitivity, ensuring users felt heard and validated. Whether users shared their worries, joys, or uncertainties, the AI engaged with them in a way that felt authentic and supportive.

10.2. Sentiment Analysis:

Imagine having someone who can sense your emotions as you speak—whether you're feeling sad, anxious, or happy. The AI Therapist accomplished this through real-time sentiment analysis, picking up on subtle emotional cues in user messages. This capability empowered the system to adapt its tone and suggestions, making the conversation feel empathetic and responsive.

.

10.3. Hybrid Recommendation System:

Support goes beyond words, and that's where the hybrid recommendation system shone. By blending content-based and collaborative filtering techniques, the AI offered personalized guidance. For instance, if a user was feeling overwhelmed, it might suggest calming exercises, uplifting articles, or tailored relaxation techniques. These recommendations were not just generic but crafted to suit individual emotional states and preferences.

10.4. Handling Complex Emotions:

Human emotions are rarely simple. They are layered, sometimes contradictory—like feeling frustrated yet hopeful or anxious yet excited. While the AI Therapist effectively managed basic emotions, navigating these complex emotional landscapes remained a challenge.

10.5. Scalability:

As more people turn to the AI Therapist for support, the system needs to scale seamlessly. Handling an influx of users without compromising response time or quality is no small feat. Behind the scenes, this means optimizing databases, improving real-time data processing, and ensuring that the platform can grow without losing its human touch.

10.6. Improving Emotional Understanding:

Understanding complex emotions isn't just about better algorithms—it's about creating a system that genuinely resonates with users. Future developments will focus on fine-tuning NLP models to recognize and address these emotional intricacies. This could involve training the AI to detect subtle shifts in tone or context, enabling it to respond with greater depth and empathy.

10.7. Enhancing Recommendation System Accuracy:

Personalization is at the core of meaningful support. By incorporating advanced techniques like reinforcement learning, the AI can learn and adapt in real-time. Imagine a system that evolves with you, offering increasingly precise and relevant suggestions as it understands your needs and preferences better with each interaction.

The AI Therapist project has taken significant steps toward making mental health support more accessible and personalized. However, the journey doesn't end here. By addressing challenges like emotional complexity and scalability, and continuing to enhance its capabilities, the platform can become an even more effective ally in users' emotional well-being. Ultimately, the goal is not just to provide solutions but to foster connections that make every user feel seen, understood, and supported.

REFERENCES

[1] Ameen Abdullah Qaid Aqlan, B. Manjula,R, Lakshman Naik,"A Study of Sentiment Analysis: Concepts, Techniques, and Challenges" , ResearchGate Publications,January 2019, 10.1007/978-981-13-6459-4_16 ,In book: Plant Long Non-Coding RNAs (pp.147-162).

[2] Eshan Shinde, Mahesh Shendage, Rahul Patil ,"ARTIFICIAL INTELLIGENCE THERAPIST",Research Gate Publications,August 2022.

[3] David B. Olawade, Ojima Z. Wada , Aderonke Odetayo , Aanuoluwapo Clement David- Olawade, Fiyinfoluwa Asaolu , Judith Eberhardt ,"ARTIFICIAL INTELLIGENCE THERAPIST",Journal of Medicine, Surgery, and Public Health Volume 3, August 2024, 100099.

[4] Erion C, Maurizio Morisio ,"Hybrid Recommender Systems: A Systematic Literature Review" ,ResearchGate Publications ,November 2017 Intelligent Data Analysis 21(6),1487- 1524,10.3233/IDA-163209.

[5] Pigi Kouki, James Schaffer, Jay Pujara, John O'Donovan, Lise Getoor ,"Hybrid Recommender Systems: A Systematic Literature Review" ,IUI '19: Proceedings of the 24th International Conference on Intelligent User Interfaces, March 2019.

[6] Mehdi Elahi , Danial Khosh Kholgh , Mohammad Sina Kiarostami , Mourad Oussalah , Sorush Saghari," Hybrid recommendation by incorporating the sentiment of product reviews",Information Sciences Volume 625, May 2023, Pages 738-756.

[7] P. Chinnasamy Ajmeera Kiran , Wing-Keung Wong , J. Chinna Babu , Ambeth Raja , Osamah Ibrahim Khalaf," Health Recommendation System using Deep Learning-based Collaborative Filtering", Published by Elsevier Ltd,Volume 9, Issue 12e22844December 2023.

[8] M.V.Patil , Subhawna, Priya Shree, Puneeth Singh," AI BASED HEALTHCARE CHAT BOT SYSTEM",ResearchGate Publications, July 2021 International Journal of Scientific and Engineering Research 12(7):668.

[9] Lalith Abualigah, Hamza Essam Alfar, Mohammed Shehab, Alhareth Mohammed Abu Hussein, "SENTIMENT ANALYSIS IN HEALTH CARE : A Brief Review", ResearchGate Publications, January 2020Studies in Computational Intelligence ,In book: Recent Advances in NLP: The Case of Arabic Language (pp.129-141) ,10.1007/978-3-030-34614-0_7.

[10] Siddharth Biju, Dolly Goplani, Aditya Khatavkar, Noopur Lade, and Pratush Jadoun," THERAPISTGPT:AN AI POWERED THERAPIST",International journal of creative research thoughts ,2023 IJCRT ,Volume 11, Issue 5 May 2023 ,ISSN: 2320-2882.

[11] Riddhi Shetty, Ankita Bhosale, Pankaj Verma, Ashwini Phalke," AI-BASED HEALTHCARE CHATBOT", SSRN ,Proceedings of the 7th International Conference on Innovations and Research in Technology and Engineering (ICIRTE-2022), organized by VPPCOE & VA, Mumbai-22, INDIA ,May 2022.

[12] Mark Lawrence, MD Istiyak , Mohd Aman," HEALTHCARE CHATBOT SYSTEM", International journal of novel research and development(IJNRD), 2024 IJNRD Volume 9, Issue 3 March 2024 ISSN: 2456-4184 ,March 2024.

[13] Ayain John, Abhigna G, Adithi K V, Harusha R, Kavya A S ,"HEALTHCARE CHATBOT" , ResearchGate Publications,International Research Journal of Computer Science 9(8):297- 303,August 2022, 10.26562/irjcs.2022.v0908.28.

[14] Idayati Mazlan, Noraswaliza Abdullah, Norashikin Ahmad,"Exploring the Impact of Hybrid Recommender Systems on Personalized Mental Health Recommendations" , ResearchGate Publications,International Journal of Advanced Computer Science and Applications 14(6), January 2023 ,License :CC BY 4.0.

[15] Eshan Shinde, Mahesh Shendage, Rahul Patil," Artificial Intelligence Therapist",International Research Journal of Modernization in Engineering Technology and Science Volume:03,April 2021.

[16] Prof. Mamta Madan, Ms.Rishima Madan, Dr Praveen Thakur," Analysing the patient sentiments in healthcare domain using Machine learning",Procedia Computer Science Volume 238, 2024, Pages 683-690 ,The 7th International Conference on Emerging Data and Industry (EDI40) April 23-25, 2024, Hasselt, Belgium.

APPENDIX-A

PSEUDOCODE

```
from sklearn.ensemble import VotingClassifier
estimators=[
    ("RFC", pipeline),
    ("Logistics Regression", logs_pipeline),
    ("Gradient Boosting",GB_pipeline),
    ("SVM", svm_pipeline)]



voting_classifier = VotingClassifier(estimators, voting='hard')
voting_classifier.fit(train_df['Text'], train_df['Emotion'])


train_pred = voting_classifier.predict(x_train_procceced)
train_accuracy = accuracy_score(train_df['Emotion'],
train_pred)
print("Train set accuracy:", train_accuracy)


# Validation set accuracy
val_pred = voting_classifier.predict(x_val_procceced)
val_accuracy = accuracy_score(val_df['Emotion'], val_pred)
print("Validation set accuracy:", val_accuracy)


# Test set accuracy
test_pred = voting_classifier.predict(x_test_procceced)
test_accuracy = accuracy_score(test_df['Emotion'], test_pred)
print("Test set accuracy:", test_accuracy)


model_accuracy = test_accuracy
```

```
print('---------------------- Model Accuracy -----------------------')

print('Our Machine Learning model has an accuracy of
{:.2f}%'.format(model_accuracy * 100))

custom_text = "it feels like my career is slipping away from
me and i don’t know how to fix it"
predicted_emotion = voting_classifier.predict([custom_text])
print("Predicted Emotion:", predicted_emotion[0])

custom_text = "I dont want this life"
predicted_emotion = voting_classifier.predict([custom_text])
print("Predicted Emotion:", predicted_emotion[0])

# Save the trained collaborative filtering model, TF-IDF
vectorizer, and the entire hybrid model
with open("collaborative_filtering_model.pkl", "wb") as
cf_model_file:
    pickle.dump(algo, cf_model_file)

with open("tfidf_vectorizer.pkl", "wb") as vectorizer_file:
    pickle.dump(tfidf_vectorizer, vectorizer_file)

with open("hybrid_recommender_model.pkl", "wb") as
hybrid_model_file:
    pickle.dump({
        "collaborative_filtering_model": algo,
        "tfidf_vectorizer": tfidf_vectorizer,
        "content_df": content_df
    }, hybrid_model_file)
```

```
def update_dataset_with_feedback():
    feedback_file = "feedback.csv"
    if os.path.exists(feedback_file):
        feedback_df = pd.read_csv(feedback_file)
        global data
        new_ratings = feedback_df[['User ID', 'Solution ID',
'Rating']]
        data = pd.concat([data, new_ratings],
ignore_index=True).drop_duplicates()
        print("Dataset updated with feedback.")
    else:
        print("No feedback data found to update the dataset.")


def retrain_collaborative_filtering():
    global algo, trainset
    reader = Reader(rating_scale=(1, 5))
    updated_cf_data = Dataset.load_from_df(data[['User ID',
'Solution ID', 'Rating']], reader)
    trainset = updated_cf_data.build_full_trainset()
    algo.fit(trainset)
    print("Collaborative filtering model retrained with updated
data.")


    # Save the updated collaborative filtering model
    with open("collaborative_filtering_model.pkl", "wb") as
cf_model_file:
        pickle.dump(algo, cf_model_file)


def refresh_content_based_data():
```

```
    global content_df, content_matrix
    content_df = data[['Solution ID', 'Solution Name', 'Type',
'Description', 'Targeted Issue']].copy()
    content_df['Content'] = content_df.apply(lambda row: '
'.join(row.dropna().astype(str)), axis=1)
    content_matrix =
tfidf_vectorizer.fit_transform(content_df['Content'])
    print("Content-based data and TF-IDF matrix refreshed.")

    # Save the updated TF-IDF vectorizer
    with open("tfidf_vectorizer.pkl", "wb") as vectorizer_file:
        pickle.dump(tfidf_vectorizer, vectorizer_file)

def save_feedback(user_id, solution_id, user_response,
rating):
    feedback_file = "feedback.csv"
    if os.path.exists(feedback_file):
        feedback_df = pd.read_csv(feedback_file)
    else:
        feedback_df = pd.DataFrame(columns=["Interaction ID",
"User ID", "Solution ID", "User Response", "Rating",
"Timestamp"])

    interaction_id = len(feedback_df) + 1
    timestamp = datetime.datetime.now().strftime("%Y-%m-
%d %H:%M:%S")
    new_feedback = pd.DataFrame([[interaction_id, user_id,
solution_id, user_response, rating, timestamp]],
                                columns=["Interaction ID", "User ID",
"Solution ID", "User Response", "Rating", "Timestamp"])
    feedback_df = pd.concat([feedback_df, new_feedback],
```

```
ignore_index=True)
    feedback_df.to_csv(feedback_file, index=False)
    print(f"Feedback for Solution ID {solution_id} saved with a
rating of {rating}.")


def is_first_user():
    feedback_file = "feedback.csv"
    return not os.path.exists(feedback_file) or
len(pd.read_csv(feedback_file)) == 0


# 5. Main Workflow for Subsequent Users
if __name__ == "__main__":
    user_id = 2  # Example User ID, this would be dynamically
determined in a real application
    text = "i want to die"
    predicted_emotion = voting_classifier.predict([text])[0]  #
Predict emotion from NLP model
    targeted_issue = predicted_emotion  # Use the predicted
emotion from NLP as the targeted issue
    top_n = 5  # Number of recommendations to generate


    print("---------------------------------------------------------------
-----------------------------")
    print("-------------------- EMOTION --------------------------")
    print("===================", predicted_emotion,
"========================")
    print(f"Subsequent user detected with emotion:
{targeted_issue}. Updating dataset and retraining models...")


    # Update dataset with feedback (assuming the feedback is
gathered elsewhere)
```

```
    update_dataset_with_feedback()

    # Retrain the collaborative filtering model
    retrain_collaborative_filtering()

    # Refresh content-based recommendation data
    refresh_content_based_data()

    # Get hybrid recommendations based on the emotion
predicted and the targeted issue
    recommendations = get_hybrid_recommendations(user_id,
targeted_issue, top_n)

    # Ensure Solution Name and ID mapping is correct
    recommendation_data = content_df[content_df['Solution
ID'].isin(recommendations)][['Solution ID', 'Solution Name']]
    solution_names = recommendation_data['Solution
Name'].tolist()
    solution_ids = recommendation_data['Solution ID'].tolist()

    # Display the valid recommendations to the user
    if solution_names:
        print(f"Updated recommendations for User {user_id}:
{solution_names}")
    else:
        print("No valid recommendations found.")

    # Collect feedback from the user with predefined options
    if solution_names:
        solution_name = solution_names[0]  # Pick the first one
```

as an example or allow user to choose.

```
    print("--------------------------------------- Recommendations
-------------------------------------------")
    print(f"Recommendation: {solution_name} (Solution ID:
{solution_ids[0]})")
    print("---------------------------------------------//--------------
-------------------------------------------")
    # Predefined options for feedback
    feedback_options = ["Very Satisfied", "Satisfied",
"Neutral", "Dissatisfied", "Very Dissatisfied"]
    print("Please provide feedback for the solution:")
    for i, option in enumerate(feedback_options, 1):
        print(f"{i}. {option}")
    print("------------------------------------//------------------------
--------------")
    # User selects feedback option
    feedback_choice = int(input("Enter the number
corresponding to your feedback: "))

    # Map the feedback choice to a text value
    user_response = feedback_options[feedback_choice - 1]

    # Ask for rating (1-5)
    rating = int(input("Please provide a rating between 1 and
5: "))

    # Find the Solution ID corresponding to the selected
Solution Name
    solution_id = solution_ids[0]  # Use the ID of the first
recommended solution
```

```
        # Save feedback with the correct Solution ID
        save_feedback(user_id, solution_id, user_response,
rating)


# Fetch the API key from the environment
api_key = os.getenv("GOOGLE_API_KEY")
if not api_key:
    raise EnvironmentError("GOOGLE_API_KEY is not set in
your .env file.")


def generate_response(user_input, emotion, recommendation,
feedback=None):
    """
    Generate an empathetic response using the AI model.
    """
    # Define the system prompt
    system_prompt = (
        "You are an AI therapist. Your goal is to respond
empathetically to users' concerns. "
        "Based on the emotional state they express, suggest
therapeutic solutions in a conversational manner."
        "\n\n"
        "The user is feeling: {emotion}\n"
        "Your recommendation: {recommendation}\n"
        "Empathetically respond with understanding and provide
more details or alternatives to the suggestion."
    )


    # Format the system prompt with inputs
    system_prompt = system_prompt.format(emotion=emotion,
```

```
recommendation=recommendation)

    # Combine the system prompt and user input into a single
string
    full_prompt = f"{system_prompt}\n\nUser Input:
{user_input}"

    # Initialize the AI model
    llm = ChatGoogleGenerativeAI(model="gemini-1.5-flash",
temperature=0, max_tokens=None, timeout=None)

    # Generate the response from the AI model
    response = llm.invoke(full_prompt)

    # Safely try common attributes
    if hasattr(response, 'content'):
        return response.content
    elif hasattr(response, 'text'):
        return response.text
    elif hasattr(response, 'message'):
        return response.message
    else:
        raise AttributeError("Unable to extract content from the
response object.")

from combine import (
    get_hybrid_recommendations,
    update_dataset_with_feedback,
    retrain_collaborative_filtering,
    refresh_content_based_data,
```

```
    save_feedback,
    is_first_user,
)
from ai_model import generate_response  # Function to
generate empathetic responses
from combine import voting_classifier  # Import sentiment
analysis classifier
import random

# Title and Introduction
st.title("AI Therapist")
st.write("Welcome! Share your concerns, and let me provide
personalized recommendations to support you.")

# Simulate or obtain user ID (this should be dynamic in
production)
user_id = random.randint(1, 10)  # Example user ID for
simulation

# Collect user input for free-text concerns
user_input = st.text_area("What's on your mind today?")

if user_input:
    # Predict the user's emotion using sentiment analysis
    predicted_emotion =
voting_classifier.predict([user_input])[0]
    st.write(f"Detected Emotion: **{predicted_emotion}**")
    targeted_issue = predicted_emotion  # Use predicted
emotion as the targeted issue
```

```
    # Update dataset and retrain models
    update_dataset_with_feedback()
    retrain_collaborative_filtering()
    refresh_content_based_data()

    # Generate hybrid recommendations based on the targeted
issue
    top_n = 5  # Number of recommendations to generate
    recommendations = get_hybrid_recommendations(user_id,
targeted_issue, top_n)

    if recommendations:
        st.write("Here are some recommendations tailored for
you:")
        recommendation_names = []

        # Loop through recommendations and map IDs to
solution names
        for rec in recommendations:
            # Find the solution name by mapping the ID to the
solution dataset
            solution_name = next(
                (solution["Solution Name"] for solution in solutions
if solution["Solution ID"] == rec),
                f"Solution {rec}"  # Default if not found
            )
            recommendation_names.append(solution_name)
            st.write(f"- {solution_name}")

        # Use the first recommendation (most relevant solution)
```

for the empathetic AI response

first_recommendation = recommendations[0] # First recommendation ID

first_recommendation_name = recommendation_names[0] # Name of the first solution

Generate empathetic AI response

empathetic_response = generate_response(user_input, targeted_issue, first_recommendation, feedback=None) # Pass the name

st.write(f"**AI Response:** {empathetic_response}")

Collect feedback from the user

feedback_options = ["Very Satisfied", "Satisfied", "Neutral", "Dissatisfied", "Very Dissatisfied"]

feedback_choice = st.radio("How satisfied are you with this solution?", feedback_options)

if feedback_choice:

Collect a rating (1-5) from the user

rating = st.slider("Rate the recommendation (1-5):", 1, 5)

save_feedback(user_id, first_recommendation, feedback_choice, rating)

st.success(f"Thank you for your feedback! You rated '{first_recommendation_name}' as '{feedback_choice}' with a rating of {rating}.")

else:

st.warning("No recommendations available for the given issue. Please try another input.")

APPENDIX-B

SCREENSHOTS

Figure 1.3

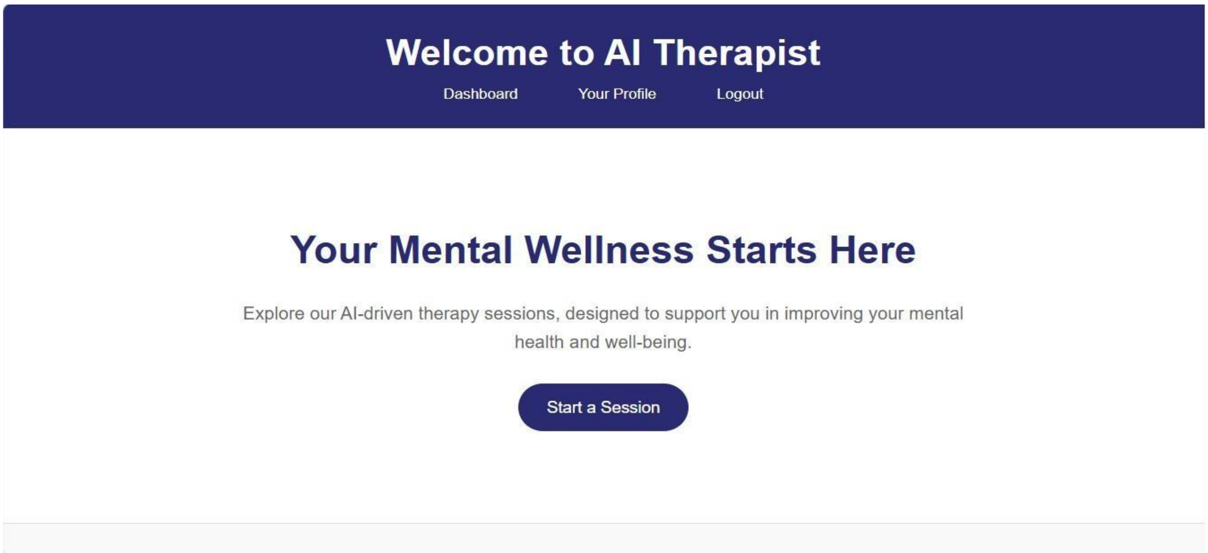


Figure 1.4

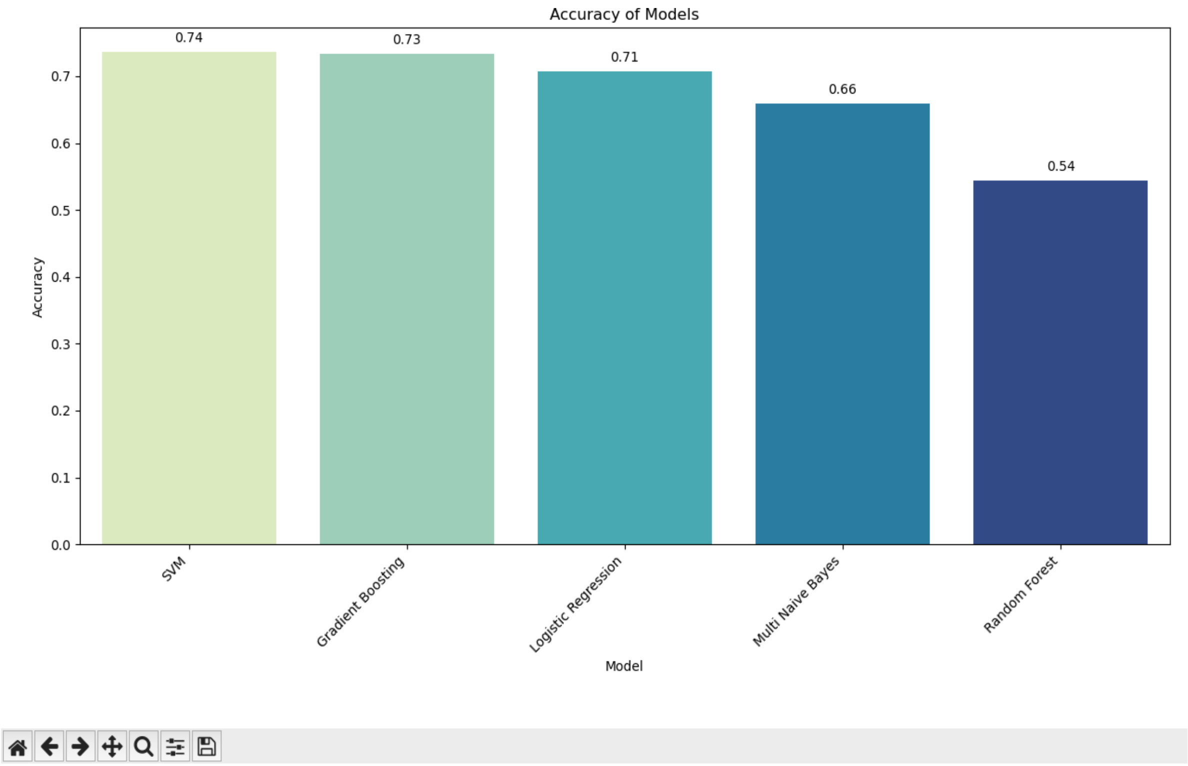


Figure 1.5

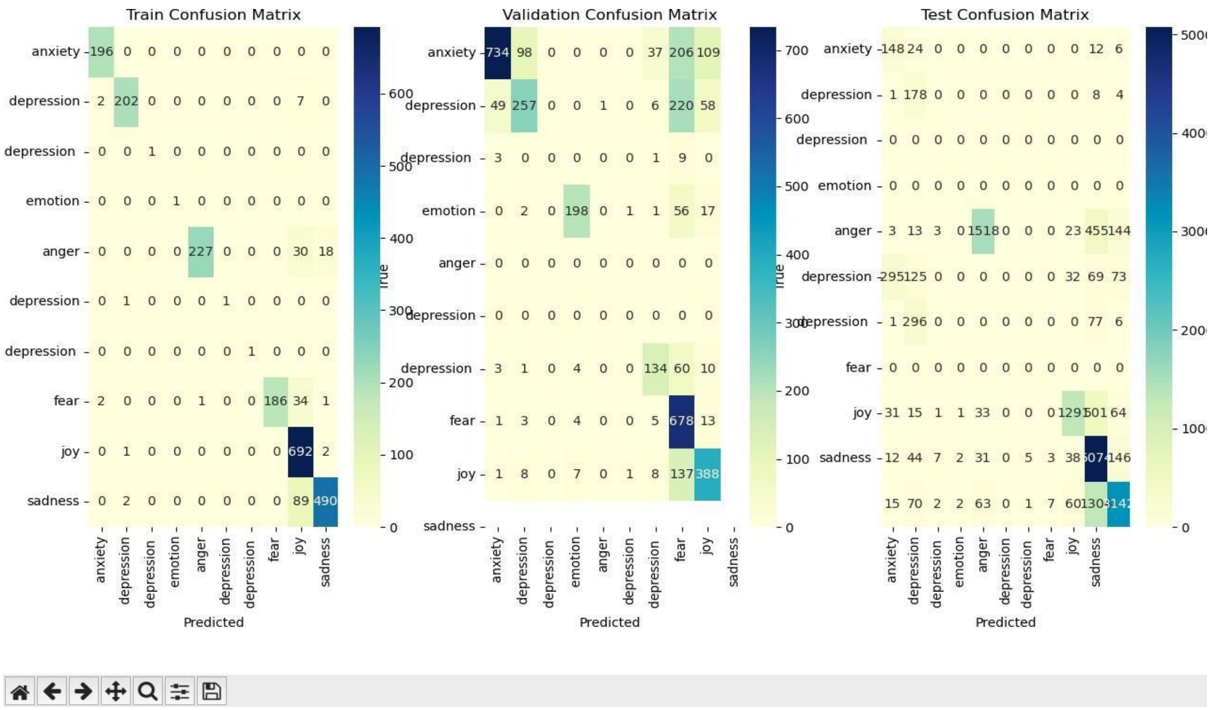


Figure 1.6

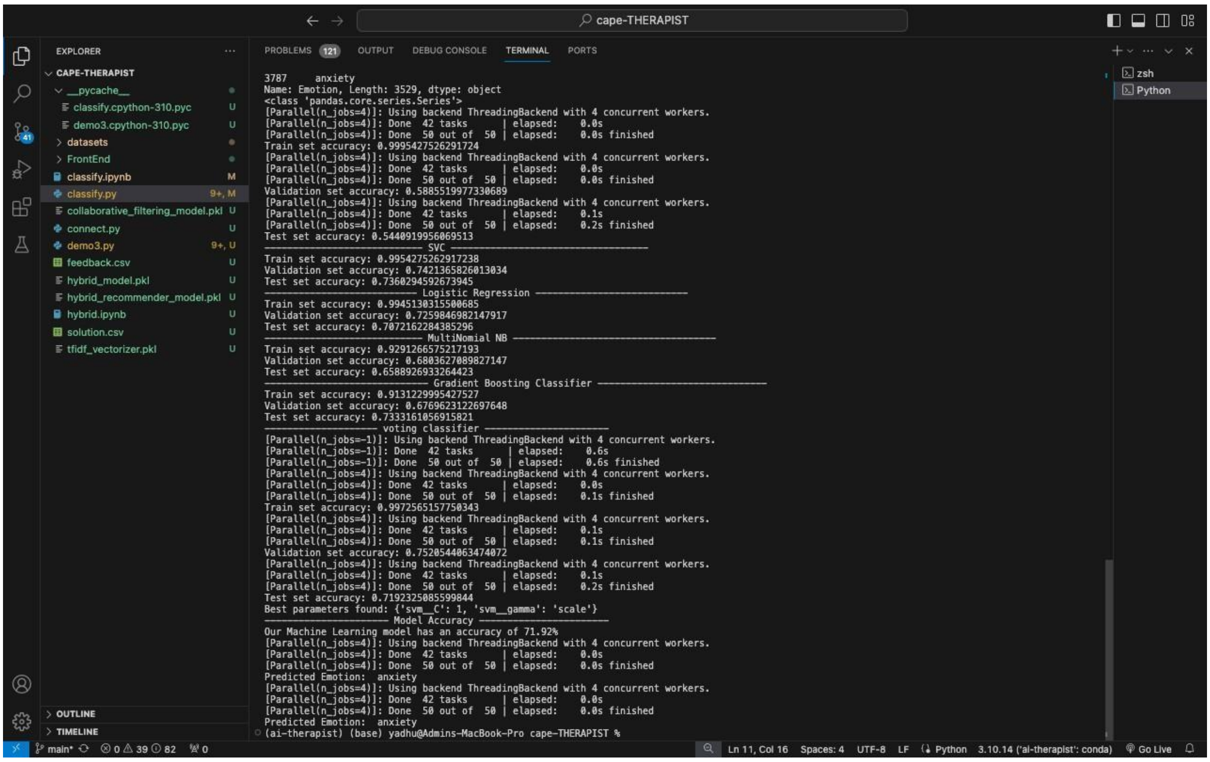
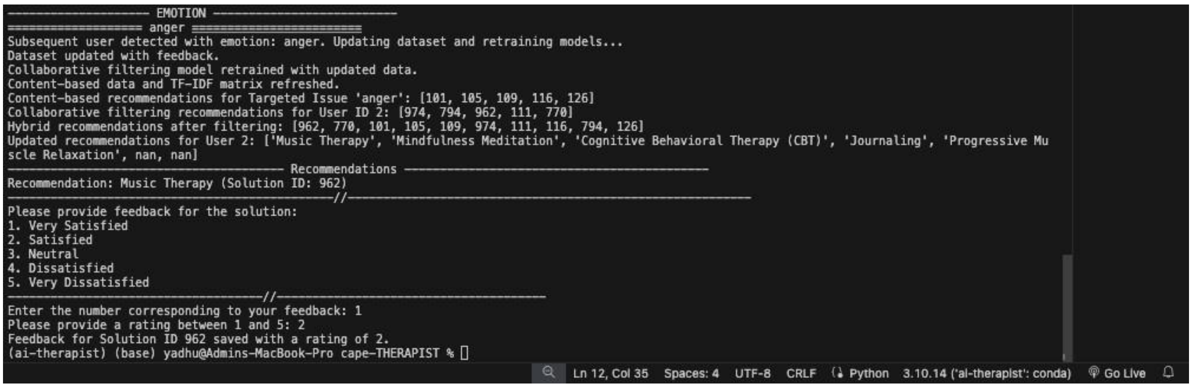


Figure 1.7


EMOTION
anger
Subsequent user detected with emotion: anger. Updating dataset and retraining models...
Dataset updated with feedback.
Collaborative filtering model retrained with updated data.
Content-based data and TF-IDF matrix refreshed.
Content-based recommendations for Targeted Issue 'anger': [101, 105, 109, 116, 126]
Collaborative filtering recommendations for User ID 2: [974, 794, 962, 111, 770]
Hybrid recommendations after filtering: [962, 770, 101, 105, 109, 974, 111, 116, 794, 126]
Updated recommendations for User 2: ['Music Therapy', 'Mindfulness Meditation', 'Cognitive Behavioral Therapy (CBT)', 'Journaling', 'Progressive Mu
scle Relaxation', nan, nan]
Recommendations
Recommendation: Music Therapy (Solution ID: 962)
//
Please provide feedback for the solution:
1. Very Satisfied
2. Satisfied
3. Neutral
4. Dissatisfied
5. Very Dissatisfied
//
Enter the number corresponding to your feedback: 1
Please provide a rating between 1 and 5: 2
Feedback for Solution ID 962 saved with a rating of 2.
(ai-therapist) (base) yadhu@Admins-MacBook-Pro cape-THERAPIST %
Ln 12, Col 35
Spaces: 4
UTF-8
CRLF
Python
3.10.14 ('ai-therapist': conda)
Go Live

SUSTAINABLE DEVELOPMENT GOAL

The Project work carried out here is mapped to SDG-3 Good Health and Well Being.

The project emphasizes the assurance of healthy lives and the welfare of all at all ages. Focusing on mental health, the project addresses one of the most significant aspects of global health by targeting conditions such as depression and anxiety that affect quality of life profoundly. Through an innovative approach, the AI therapist engages technology to deliver on-demand mental health support, enabling empathetic suggestions and verbal reasoning. This will not only help address the unmet need for mental health support, most of which is available only in areas with good access to professionals, but will also empower users to become proactive agents for creating and maintaining well-being for themselves. In addition, the feedback mechanism helps upgrade and improve the system over time, ensuring its flexibility and effectiveness. This project promotes awareness and good mental health support to empower people across the world.