EMPOWERING MENTAL HEALTH CARE WITH ARTIFICIAL INTELLIGENCE TOOL

BY

KOEY HOR YEE

A REPORT
SUBMITTED TO
Universiti Tunku Abdul Rahman
in partial fulfillment of the requirements
for the degree of
BACHELOR OF COMPUTER SCIENCE (HONOURS)
Faculty of Information and Communication Technology
(Kampar Campus)

JUNE 2024

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Empowering Mental Health Care with Artificial Intelligence Tool

Academic Session: June 2024

I KOEY HOR YEE

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

Siti Nurlaili

(Author's signature) (Supervisor's signature)

Address:
1, Persiaran Bercham Utara 7,
Taman Ramai, 31400, Ipoh,
Perak.

Dr. Siti Nurlaili Binti Karim
Supervisor's name

Date: 10 September 2024

Date: 10 September 2024

Universiti Tunku Abdul Rahman			
Form Title : **Sample of Submission Sheet for FYP/Dissertation/Thesis**			
Form Number: **FM-IAD-004**	Rev No.: **0**	Effective Date: **21 JUNE 2011**	Page No.: **1 of 1**

FACULTY/INSTITUTE* OF INFORMATION AND COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 10 September 2024

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that KOEY HOR YEE (ID No: 2006028) has completed this final year project/ dissertation/ thesis* entitled "Empowering Mental Health Care with Artificial Intelligence Tool" under the supervision of, Dr. Siti Nurlaili Binti Karim (Supervisor) from the Department of Computer Science, Faculty/Institute* of Information and Communication.

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

Koey Hor Yee

*Delete whichever not applicable

DECLARATION OF ORIGINALITY

I declare that this report entitled "**Empowering Mental Health Care with Artificial Intelligence Tool**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : __________

Name : Koey Hor Yee

Date : 10 September 2024

ACKNOWLEDGEMENTS

I would like to express thanks and appreciation to my supervisor Dr. Siti Nurlaili Binti Karim and my moderator, Dr Tse Siu Hong Savio who have given me a golden opportunity to involve in the Internet of Things field study. Besides that, they have given me a lot of guidance in order to complete this project. When I was facing problems in this project, the advice from them always assists me in overcoming the problems. Again, a million thanks to my supervisor and moderator.

Finally, I must say thanks to my parents and my family for their love, support, and continuous encouragement throughout the course.

ABSTRACT

This project aims to develop an AI-driven mobile application specifically designed to address the pervasive mental health challenges faced by university students, particularly those at UTAR Kampar. It highlights the widespread prevalence of mental health issues within this demographic and explores the transformative potential of Artificial Intelligence (AI) in providing immediate, cost-effective, and non-judgmental support. Through an extensive literature review, the study investigates how AI can help reduce mental health issues and evaluates existing mental health applications, assessing the efficacy of AI technologies in mental health care. The core objective is to create a user-centric, AI-powered mobile application that functions as a virtual companion. This companion offers students a confidential platform to engage in conversations and practice communication skills without the fear of judgment. The application includes features such as sentiment analysis, mental health assessments, and access to helpline and contact information, offering a comprehensive approach to mental health support. These features help users gain insights into their emotional states, assess their mental well-being, and access immediate assistance when needed. By leveraging AI's capabilities to provide personalized and accessible support, this project aims to enhance the emotional well-being and overall quality of life for university students, while also preparing them for more effective interpersonal interactions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
NLP	Natural Language Processing
CNN	Convolutional Neural Networks
CLBP	Central Local Binary Pattern
SER	Speech Emotion Recognition
MLR	Multivariate linear regression
SVM	Support Vector Machines
HMM	Hidden Markov Model

Chapter 1
Introduction

This chapter outlines the motivation behind developing an AI-driven mental health support system, specifically targeting UTAR Kampar students. It begins with the problem statement and the significance of addressing mental health concerns in university settings. Key objectives of the project are defined, including the development of a mobile application, implementation of AI-based response generation and sentiment analysis, and incorporation of mental health assessment tools. The chapter also discusses the project’s scope, contributions, and future directions, emphasizing its potential impact on mental health care for students.

1.1 Problem Statement and Motivation

The prominence of mental health concerns, often overshadowed by physical health, demands greater attention due to their significant impact on overall well-being. Globally, nearly 970 million people are affected by mental illness and substance abuse [1], with one in four likely to face a mental health challenge in their lifetime. The COVID-19 pandemic exacerbated these issues, dramatically increasing cases of depression and anxiety due to prolonged social isolation and restricted interpersonal interactions [2].

Moreover, the universal nature of mental health issues is evident across diverse demographics, including university students, who are particularly vulnerable due to academic pressures and significant life transitions. Research indicates that the university environment can be a critical juncture for mental health struggles [3]. At UTAR Kampar, students face stressors that can detrimentally affect their mental health, impacting both their academic performance and their susceptibility to severe psychological distress. Mental health issues not only affect students' academic achievements but also heighten their vulnerability to suicidal thoughts [4]. Prioritizing mental well-being during this pivotal phase is crucial, as it directly affects focus and effective learning, thereby fostering emotional resilience [5].

The advent of Artificial Intelligence (AI) in healthcare presents a pivotal opportunity for innovative mental health solutions. AI technology has the capability to process data, recognize patterns, and interact using natural language, opening new avenues for empathetic and accessible care. For example, Marr states that AI aids in the diagnosis and early detection of conditions such as depression and suicidal tendencies before they become more severe [6]. This project leverages a combination of AI technologies, including natural language processing, deep learning, and machine learning, along with mobile app development technologies, encompassing backend and frontend development, to achieve its objectives.

Limited access to traditional mental health resources in the Kampar region further complicates these challenges, underscoring the need for an innovative approach to mental health care. This project proposes an AI-driven mental health care system that is not only immediate and cost-effective but also sensitive to the user’s privacy and convenience. Integrating advanced AI capabilities within a mobile app aims to provide a solution that addresses the specific needs of UTAR students, enhancing their mental well-being and thus, their overall quality of life during their university years.

1.2 Objectives

The overarching goal of this project is to develop an AI-driven mental health care support mobile app for UTAR Kampar students, focusing on enhancing emotional well-being and providing mental health support through an innovative mobile application. This initiative aims to leverage advanced technology, such as AI models, web frameworks, UI toolkits, and libraries, to offer real-time, personalized mental health support, catering specifically to the needs of the student population.

Objective 1 – Development of the Mobile Application:

The first objective is to develop a comprehensive mobile application that will serve as the central platform for mental health support. This application will provide 24/7 access to mental health resources and tools. It is designed to be engaging and user-friendly, ensuring that help is easily accessible whenever students need it.

Objective 2 – Implementation of AI-driven Response Generation and Sentiment Analysis:

This objective involves implementing an AI-driven function capable of generating responsive dialogues and performing sentiment analysis. This functionality will enable the system to assess users' emotional states through their text inputs and generate appropriate responses. It offers immediate and effective assistance, setting a strong foundation for integrating this functionality into the mobile app.

Objective 3 – Incorporation of Mental Health Assessments:

The third objective is to incorporate comprehensive mental health assessment tools into the application. These tools will enable users to regularly evaluate their mental well-being and assist in identifying early signs of mental health issues. By providing users with valuable insights into their emotional health status, the app will play a crucial role in proactive mental health management.

The third objective is to incorporate comprehensive mental health assessment tools into the application. These tools will enable users to regularly evaluate their mental well-being and assist in identifying early signs of mental health issues. By providing users with valuable insights into their emotional health status, the app will play a crucial role in proactive mental health management.

This semester, the project successfully achieved all set objectives. Additionally, an extra objective was met, **providing users with information and contact details for mental health helplines**. The successful implementation of these features has laid a solid foundation for the subsequent development phases of the mobile application and its additional functionalities.

1.3 Project Scope and Direction

This project, titled "Empowering Mental Health Care with Artificial Intelligence," is focused on developing and deploying a mobile application tailored to address mental health challenges among UTAR Kampar students. This initiative aims to create an intuitive mobile app that integrates advanced technologies, including Natural Language Processing (NLP) and Machine Learning (ML). The application will feature a user-friendly interface, ensuring seamless access to resources at any time.

A robust backend infrastructure will support efficient data handling and security, which is crucial for maintaining user privacy and ensuring smooth app functionality. The primary goal of this project is to enhance emotional resilience by providing a chatbot, assisting in tracking students' mental health status, and offering helpline information. The app also aims to foster a supportive campus environment and reduce the stigma associated with mental health issues.

The application will be continuously updated based on user feedback and emerging mental health trends. This approach ensures that the app not only meets the immediate needs of UTAR Kampar students but also adapts to future demands. Future developments may include more personalized mental health tracking and support tools, with the potential to expand the app’s reach to other educational institutions, thereby broadening its impact.

1.4 Contributions

The development and implementation of this AI-driven mental health care mobile application for UTAR Kampar students exemplify the practical and effective use of AI in mental health interventions, making a significant contribution to the field.

Firstly, the project pioneers the innovative application of Natural Language Processing (NLP) and Machine Learning (ML) to analyze sentiment and emotional states in real-time. This demonstrates how AI can be tailored to address specific psychological needs, providing virtual buddy support that responds to users' emotional cues.

Secondly, the project greatly enhances accessibility and user engagement through the development of a user-friendly mobile platform that operates 24/7. This continuous accessibility lowers barriers to mental health support and encourages a proactive approach to mental health care among university students, ensuring support is available whenever needed.

Thirdly, the application incorporates a link to mental health assessments provided by Malaysia Mentari through an in-app browser. Additionally, information and contact details for mental health helplines are made available using an in-app browser to access the UTAR Hospital website link. These features help users identify potential mental health issues, understand their mental health status, and find support when necessary. Although these assessments and helpline information are not AI-driven, they are integral to the app’s holistic support framework.

Additionally, the project addresses the stigma associated with mental health issues by integrating empathetic AI interactions into students' daily lives. This fosters a more supportive and understanding campus environment, enhancing overall mental health awareness.

Lastly, the project establishes a foundation for continuous improvement, with future developments planned based on user feedback and testing. This approach ensures the application remains effective and responsive to the evolving needs of its users. The contributions made in this phase of the project underscore its potential to not only improve mental health management at UTAR Kampar but also to serve as a model for integrating AI technology in mental health care across other institutions. The planned enhancements and iterative refinements in the next phase will build on the solid foundation established in this work.

Lastly, the project lays the groundwork for establishing a framework for continuous improvement, which will be further developed in the future based on feedback and testing. This approach ensures that the application remains effective and responsive to the evolving needs of its users. The contributions made in this phase of the project highlight its potential not only to improve mental health management at UTAR Kampar but also to serve as a model for broader applications in mental health care and AI technology integration across other institutions and settings. The planned enhancements and iterative refinements, scheduled for the next phase, will continue to build on the solid foundation laid in the current work.

1.5 Report Organization

This report is organized into seven chapters. Chapter 1 introduces the project, outlining the problem statement, objectives, scope, and contributions. Chapter 2 reviews the relevant technologies and existing systems that informed the project's development. Chapter 3 explains the system design methodology, including key diagrams that illustrate the architecture and functionality. Chapter 4 focuses on the system's design, detailing the components and interactions. Chapter 5 discusses the implementation process, highlighting the setup and challenges faced. Chapter 6 evaluates the system's performance through testing, while Chapter 7 concludes the report with key findings and recommendations for future improvements.

Chapter 2
Literature Review

This literature review explores the role of artificial intelligence (AI) in addressing mental health issues and provides a comprehensive evaluation of existing digital products designed for mental health support. The review begins with an examination of how AI technologies are applied to mental health care, followed by a detailed analysis of popular products in the market: Happify, TalkSpace, and Calm. Each product is evaluated for its strengths and weaknesses, offering insights into their effectiveness and user experience. The review also includes an assessment of the chosen AI models used in the development of our system, specifically DistilBERT for sentiment analysis and DialoGPT for response generation. Additionally, the review covers the backend technology (Flask) and frontend technology (Flutter) utilized in our application. By comparing existing solutions and evaluating the selected technologies, this review aims to provide a foundation for understanding the current landscape of digital mental health tools and the rationale behind the technology choices in our system.

2.1 Artificial Intelligence in Addressing Mental Health Issue

Mental health awareness is a critical concern in educational settings, particularly in universities. The stigma surrounding mental health issues can serve as a significant barrier to seeking help. Numerous studies have underscored the detrimental effects of this stigma on students' mental well-being. For example, research conducted by Hofstra et al. [7] emphasized that while the number of students with mental health issues has increased, only a portion of them seek help due to the fear of discrimination, such as concerns about potential career opportunities in the future. Hence, the significance of employing AI to reduce this stigma cannot be overstated, as it contributes to the creation of a supportive campus environment.

The role of mobile applications in addressing mental health concerns has gained prominence in recent years. Several mobile applications have been developed to provide support for individuals experiencing mental health issues. For example, Khoo [8] highlighted five mental health apps in Malaysia, including Thoughfull Chat, PlusVibes, The Help Talk, Naluri, and Intellect, which allow users to connect with mental health professionals anywhere. These apps have demonstrated the potential to improve users' moods and reduce stress, and this type of app is believed to be a significant trend in the future market.

Artificial intelligence (AI) has found applications in mental health support, particularly as an assessment tool for mental health issues, and it has demonstrated remarkably accurate results. In an article by Marr [9], research conducted at Vanderbilt University Medical Center achieved an 80 percent accuracy rate in predicting the likelihood of an individual taking their own life using Machine Learning. This achievement has fostered the belief that AI holds the capacity to create a favourable influence in the realm of mental health care.

Furthermore, AI technologies extend beyond Machine Learning (ML) and encompass Natural Language Processing (NLP), which is used to understand human language in the context of mental health care. Research conducted by Banerjee [10] emphasizes the effectiveness of various AI technologies in healthcare clinical practice, including Machine Learning (ML), Deep Learning (DL), Computer Vision, and Natural Language Processing (NLP). These AI-powered technologies can provide valuable insights and suggestions to users, thereby enhancing their emotional well-being.

Recent studies have shown that AI-powered mental health applications can significantly benefit users in various ways. An overview by Haque and Rubya [11] investigated the AI chatbox Woebot is decreased depressive symptoms for the college student with depression. The research found that AI-driven apps not only improved users' emotional well-being but also provided continuous monitoring and early intervention. The AI is trained to do Facial Expression Analysis, Speech and Voice Analysis, Text and Sentiment Analysis infer human emotions instead of the traditional method which is rely on the self-report by the user Jbaily [12]. By analysing patterns in users' responses and behaviours, AI could detect subtle changes in emotional states over time.

Facial expressions refer to the range of emotional and communicative signals conveyed by humans through their faces. In real life, there are many facial expressions, but researchers have identified seven universal facial expressions: anger, contempt, disgust, fear, happiness, sadness, and surprise [13]. These seven facial expressions are crucial for AI systems designed to recognize human emotions.

AI processes human faces as objects using machine learning and deep learning techniques, analysing features like brows, eyes, and mouth [14]. A common example of deep learning models used for facial expression analysis is Convolutional Neural Networks (CNNs). Additionally, when combined with CNNs, the Central Local Binary Pattern (CLBP) algorithm has been shown to achieve an average facial expression recognition rate of 88.16% [15].

Speech emotion recognition (SER) is a domain within artificial intelligence and natural language processing (NLP) dedicated to automatically identifying and examining human emotions conveyed through spoken language. The primary goal of SER is to identify the emotional or affective state of a speaker based on their speech patterns, tone, and prosody. Multivariate linear regression (MLR) and support vector machines (SVM), where SVM techniques are commonly used for emotion recognition in spoken audio signals [16].

SER works by analysing audio signals to extract emotional features from speech, and various data transformation methods are used to represent this information numerically. Additionally, direct speech recognition techniques enable the conversion of spoken words into text, facilitating tasks like subtitling and transcription [17]. Furthermore, SER algorithms leverage statistical models like HMMs, linguistic understanding through NLP, and predictive models like N-grams, often enhanced by AI and machine learning, to convert spoken language into text or meaningful commands [18].

Text analysis involves systematically examining unstructured text data to extract valuable information and patterns [19], while sentiment analysis focuses on determining the emotional tone expressed in text, such as positive, negative, or neutral sentiments [20]. The steps of sentiment analysis work by collecting data, cleaning the data, extracting features, selecting an ML model, and performing sentiment classification [21]. Both techniques can be suitable for AI mental health applications, as they enable the analysis of text-based mental health assessments to gauge users' emotional states, providing valuable insights and support.

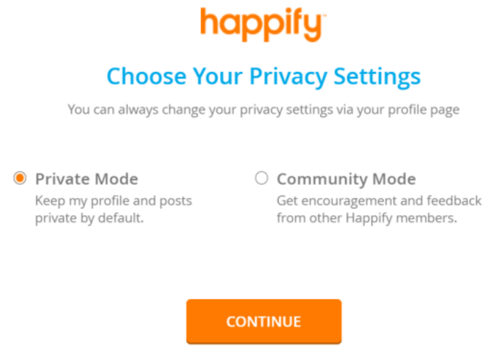
Although facial expressions and speech emotion recognition are valuable for developing AI mental health care applications, they may involve privacy issues due to the collection of biometric data and potential intrusion into individuals' emotional and personal space. In contrast, text analysis and sentiment analysis typically involve fewer privacy concerns because they analyse less intrusive text data, allow for user control over generated content, offer better anonymization potential, and are subject to distinct privacy regulations. In summary, AI-powered mental health applications offer not only accurate assessments but also the advantages of continuous monitoring, early intervention, and accessibility to mental health care services. These apps have the potential to transform the way students manage their mental well-being, providing them with personalized support and resources when they need it most.

2.2 Review of Existing Product: Happify

Happify, established in 2012 in New York City by Tomer Ben-Kiki, Ofer Leidner, and Andy Parsons, emerged from a shared belief among these three serial entrepreneurs that technology should be harnessed to improve people's lives. Happify's mission centres on adopting a 21st-century approach to helping individuals lead fulfilling lives. Recognizing that happiness is a skill that can be cultivated, Happify aims to make the fruits of scientific discovery accessible and practical through interactive means. In a world marked by increasing stress and complexity, the platform seeks to offer enjoyable, personalized, and science-based pathways to greater happiness. Above all, Happify's goal is to empower individuals to take charge of their emotional well-being, equipping them with the necessary tools to do so.

2.2.1 Pros of Happify

Happify offers a user-friendly experience right from the start, providing clear guidelines on how the platform can benefit users immediately after registration. While it offers a free version, a premium option is available, catering to different preferences. With over 110 tracks at users' disposal, Happify allows individuals to assess their well-being and assigns corresponding activities. These tracks, divided into four parts, guide users through happiness-boosting activities designed by experts. Happify demonstrates its commitment to user communication by offering modes that let individuals choose between privacy and community interaction. Additionally, the platform offers an abundance of instant-play options based on the S.T.A.G.E.R™ framework, representing the six essential well-being skills. Each activity is thoughtfully backed by scientific research, providing users with insights into why it works and how it benefits them.



happify

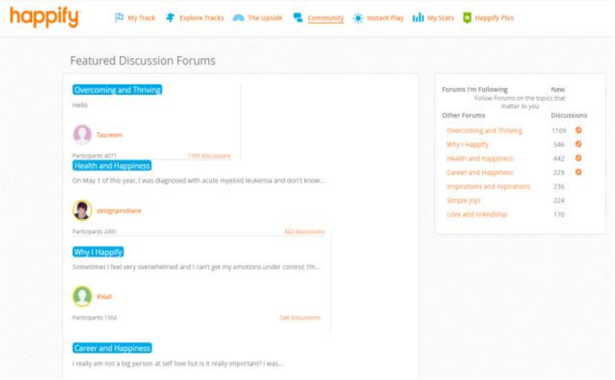
Choose Your Privacy Settings

You can always change your privacy settings via your profile page

● Private Mode
Keep my profile and posts private by default.

○ Community Mode
Get encouragement and feedback from other Happify members.

Figure 2.1: Community section of Happify (1)



Featured Discussion Forums

Overcoming and Thriving
Hello
Tasneem
Participants 4071 1109 discussions

Health and Happiness
On May 1 of this year, I was diagnosed with acute myeloid leukemia and don't know...
designprodiane
Participants 2491 442 discussions

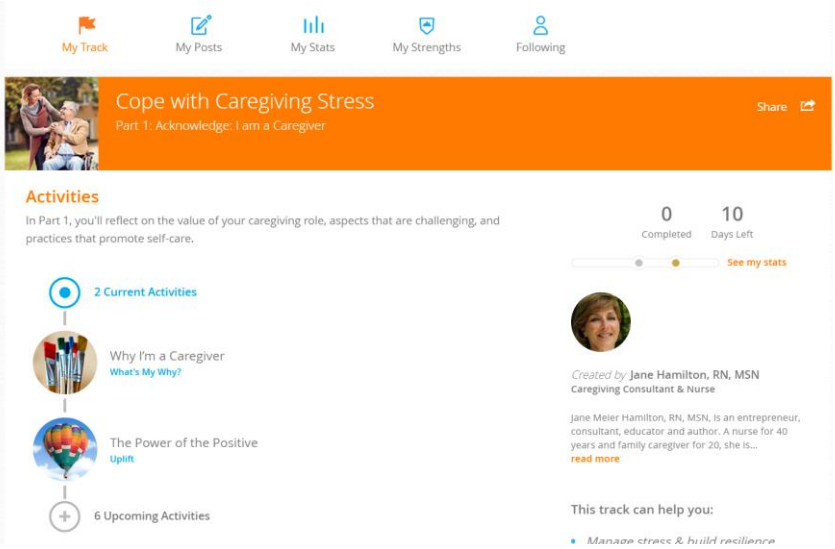
Why I Happify
Sometimes I feel very overwhelmed and I can't get my emotions under control; I'm...
KHall
Participants 1564 546 discussions

Career and Happiness
I really am not a big person at self love but is it really important? I was...

Forums I'm Following New
Follow Forums on the topics that matter to you

Other Forums	Discussions
Overcoming and Thriving	1109
Why I Happify	546
Health and Happiness	442
Career and Happiness	229
Inspirations and Aspirations	236
Simple Joys	224
Love and Friendship	170

Figure 2.2: Community section of Happify (2)



Cope with Caregiving Stress

Part 1: Acknowledge: I am a Caregiver

Activities

In Part 1, you'll reflect on the value of your caregiving role, aspects that are challenging, and practices that promote self-care.

Created by Jane Hamilton, RN, MSN
Caregiving Consultant & Nurse

Jane Meier Hamilton, RN, MSN, is an entrepreneur, consultant, educator and author. A nurse for 40 years and family caregiver for 20, she is...

This track can help you:

- *Manage stress & build resilience*

Figure 2.3: Tracks of Happify (1)

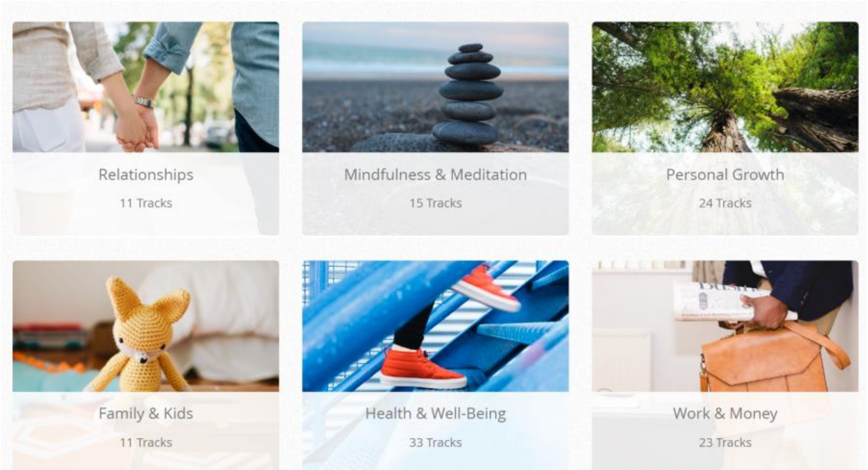
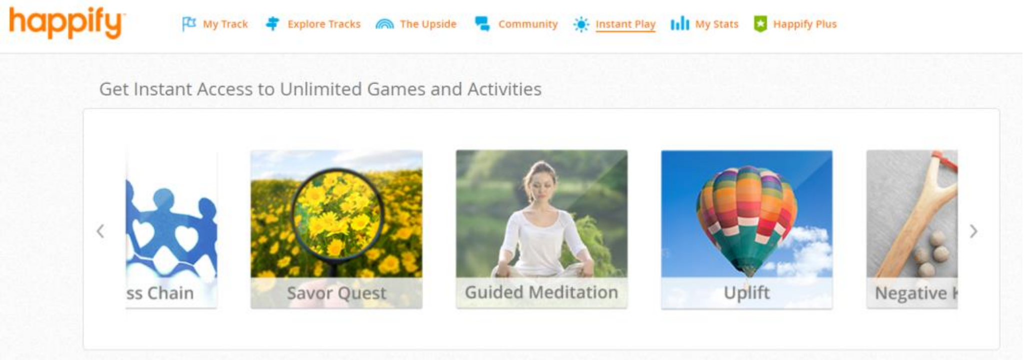


Figure 2.4: Tracks of Happify (2)



Get Instant Access to Unlimited Games and Activities

Figure 2.5: Instant Game of Happify

2.2.2 Cons of Happify

Happify's premium account requirement to unlock many of its features, such as the majority of tracks and instant-play options, may limit access for users who opt for the free version. Additionally, the platform lacks an AI monitoring system to address discussions related to suicide or self-harm within its community. While it provides a support email address and phone number, a more explicit approach to handling such situations could enhance user safety and well-being.

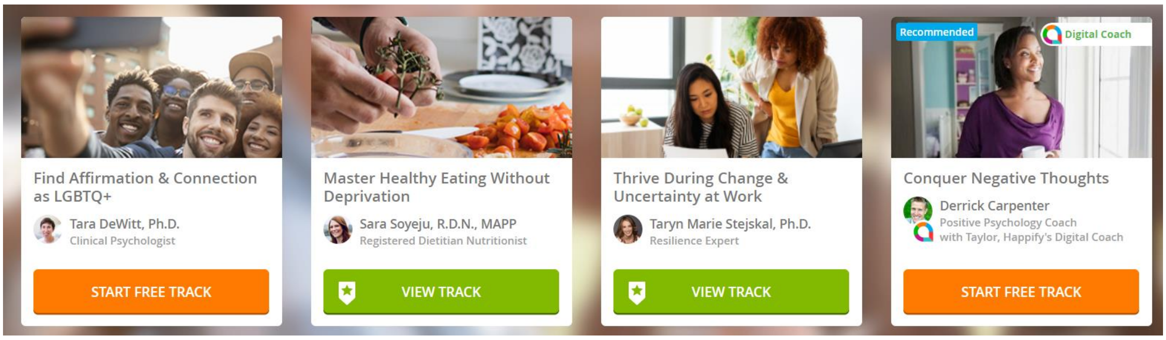

Find Affirmation & Connection as LGBTQ+
Tara DeWitt, Ph.D.
Clinical Psychologist
START FREE TRACK
Master Healthy Eating Without Deprivation
Sara Soyeju, R.D.N., MAPP
Registered Dietitian Nutritionist
VIEW TRACK
Thrive During Change & Uncertainty at Work
Taryn Marie Stejskal, Ph.D.
Resilience Expert
VIEW TRACK
Recommended
Digital Coach
Conquer Negative Thoughts
Derrick Carpenter
Positive Psychology Coach
with Taylor, Happify's Digital Coach
START FREE TRACK

Figure 2.6: Lock features of Happify (1)

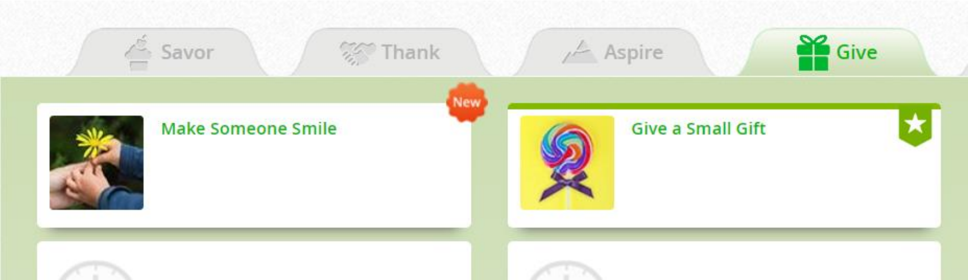


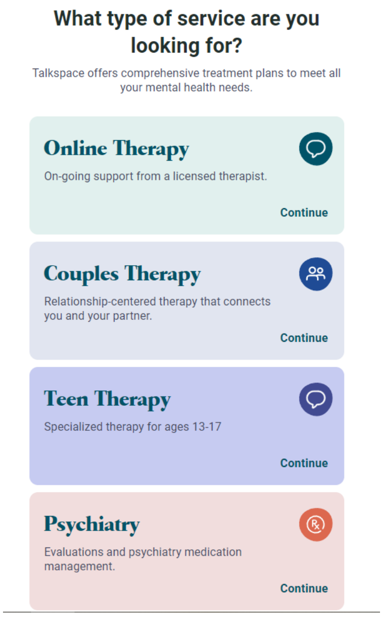
Figure 2.7: Lock features of Happify (2)

2.3 Review of Existing Product: TalkSpace

Talkspace is a prominent online therapy platform that facilitates connections between individuals seeking mental health support and licensed therapists and mental health professionals. Through a range of communication methods such as text, audio, and video messaging, Talkspace offers convenient and accessible therapy services. Users have the flexibility to engage in therapy sessions with their chosen therapist via the Talkspace app or website, making it a preferred choice for those seeking remote mental health support. Founded in 2012 and headquartered in New York City, Talkspace provides a wide array of therapy services, including individual therapy, couples therapy, family therapy, and group therapy, catering to diverse needs and challenges individuals and families may face.

2.3.1 Pro of TalkSpace

Talkspace offers a comprehensive range of therapy services tailored to individual needs, making it a versatile platform for mental health support. Users can access various types of therapy, including individual, couple, teen, and psychiatry sessions, allowing them to find the specific support they require. One of Talkspace's standout features is its round-the-clock accessibility, ensuring that users can connect with their therapists whenever they need assistance. The inclusion of live video sessions further enhances the therapy experience, providing real-time, interactive support that bridges the gap between traditional in-person therapy and online convenience.



What type of service are you looking for?

Talkspace offers comprehensive treatment plans to meet all your mental health needs.

Online Therapy

On-going support from a licensed therapist.

Couples Therapy

Relationship-centered therapy that connects you and your partner.

Teen Therapy

Specialized therapy for ages 13-17

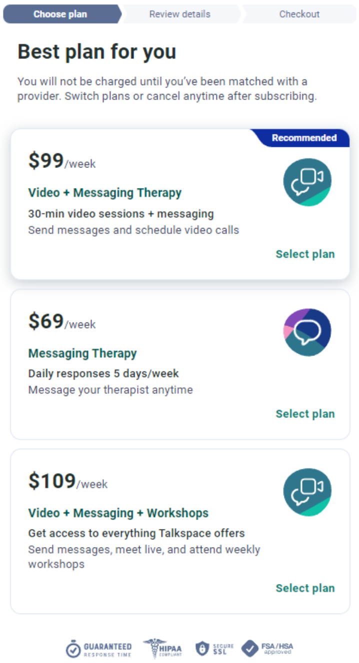
Psychiatry

Evaluations and psychiatry medication management.

Figure 2.8: Range of Therapy services of Talkspace

2.3.2 Cons of Talkspace

While Talkspace excels in providing live therapy sessions, it may not be as comprehensive when it comes to self-help resources. Unlike platforms like Happify, which offer a wide array of assessments and activities to help users manage stress and emotional well-being independently, Talkspace primarily focuses on therapy sessions. Another consideration is the cost, as Talkspace operates on a subscription model starting at $69 per week. While the platform provides access to live sessions, each session is limited to 30 minutes, which might not be sufficient for individuals seeking more extended therapy sessions. These factors should be weighed against the platform's benefits when considering its suitability for one's mental health needs.



Best plan for you

You will not be charged until you've been matched with a provider. Switch plans or cancel anytime after subscribing.

$99/week

Video + Messaging Therapy

30-min video sessions + messaging

Send messages and schedule video calls

$69/week

Messaging Therapy

Daily responses 5 days/week

Message your therapist anytime

$109/week

Video + Messaging + Workshops

Get access to everything Talkspace offers

Send messages, meet live, and attend weekly workshops

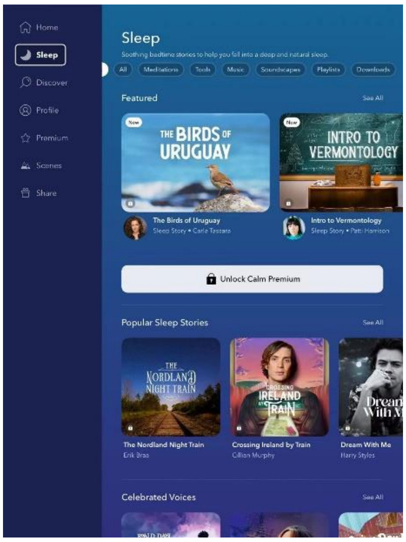
Figure 2.9: Subscription cost of Talkspace

2.4 Review of Existing Product: Calm

Calm is a meditation and wellness app that was established in 2012 and is headquartered in San Francisco, California. Founded by Michael Acton Smith and Alex Tew, the app has garnered recognition and acclaim, including being named Apple's App of the Year in 2017 and 2020. With over 100 million users worldwide and features highlighted in notable publications like The New York Times and Forbes, Calm is dedicated to enhancing the health and happiness of individuals. The app offers a diverse range of tools to promote relaxation, focus, and improved sleep, including guided meditations, soothing sleep stories, calming.

2.4.1 Pros of Calm

Calm is celebrated for its multifaceted approach to promoting mental well-being. One of its significant advantages lies in the variety of meditation and relaxation content it offers. With over 100 guided meditations, 70 sleep stories, and diverse selections of calming music and nature sounds, Calm caters to a wide range of preferences and needs, making it a versatile tool for stress reduction.

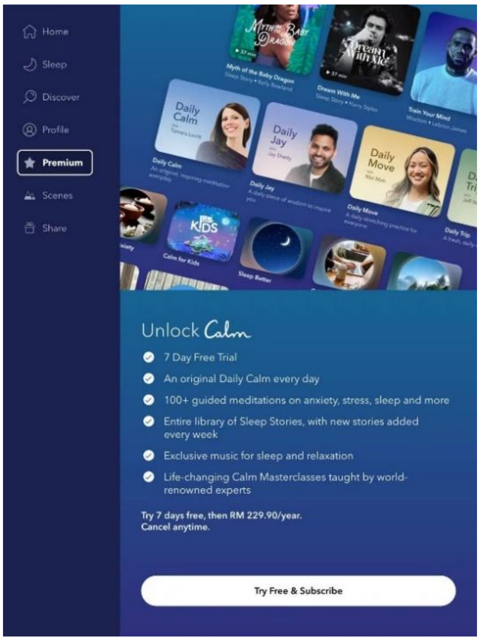


Soothing bedtime stories to help you fall into a deep and natural sleep.

Figure 2.10: Variety of meditation and relaxation content of Calm

2.4.2 Cons of Calms

However, a notable disadvantage of Calm is its subscription cost. Although the app does offer a free version, the premium option can be relatively expensive. This pricing structure might hinder accessibility for users who could greatly benefit from the app's full array of features but are discouraged by the associated costs. Another drawback of Calm is its limited functionality. While it excels in relaxation and meditation offerings, it does not provide therapy services akin to Talkspace or the tracking and assessment features available in Happify.



Unlock Calm

- 7 Day Free Trial
- An original Daily Calm every day
- 100+ guided meditations on anxiety, stress, sleep and more
- Entire library of Sleep Stories, with new stories added every week
- Exclusive music for sleep and relaxation
- Life-changing Calm Masterclasses taught by world-renowned experts

Try 7 days free, then RM 229.90/year.
Cancel anytime.

Figure 2.11: Subscription cost of Calm

2.5 Comparison of Existing Product

Each of the existing products has its own advantages and disadvantages, as well as features.

	Happify	Talkspace	Calm
Community section	YES	NO	NO
Mood Track	YES	NO	NO
Instant game	YES	NO	NO
Various Therapy	NO	YES	NO
24/7 Accessibility Therapist	NO	YES	NO
Subscription cost	YES	YES	YES
Relaxation content	NO	NO	YES

Table 2.1 Comparison of Existing Products

2.6 Review of Chosen AI Models

2.6.1 Review of DistilBERT for sentiment analysis

DistilBERT, short for 'Distilled-BERT,' was developed by the team at Hugging Face as a streamlined version of the groundbreaking BERT (Bidirectional Encoder Representations from Transformers) model introduced by Google. The term 'Distilled' refers to the reductions in both size and computational demands compared to the original BERT model. According to an overview by Holt-Nguyen, DistilBERT has 40% fewer parameters and operates 60% faster than its BERT predecessor while maintaining similar quality [22]. While DistilBERT serves as a structured, more compact version derived from BERT, understanding the foundational model upon which DistilBERT is built is crucial. BERT, a pre-trained language representation model, revolutionized natural language processing. An article by Horev explained that BERT employs the Transformer's encoder to process text non-directionally, allowing it to understand each word in the context of all surrounding words [23].

The architectural and operational enhancements of DistilBERT clearly demonstrate why this model is exceptionally suited for sentiment analysis, particularly within the framework of a mental health support chatbot. Such chatbots require accurate and real-time analysis of textual data to understand the emotional states of users, which is critical in providing effective mental health support. DistilBERT's capabilities are pivotal in these scenarios, where detecting subtle nuances in sentiment can guide the chatbot’s interactions and interventions.

The efficiency and speed of DistilBERT are vital attributes for processing extensive textual data from user interactions. This allows the chatbot to offer immediate insights into user sentiment, essential for identifying distress signals or changes in emotional states that may require urgent attention. The streamlined architecture of DistilBERT, which retains 60% of BERT’s parameters while achieving up to 95% of its performance as mentioned by its developer, Hugging Face [24], ensures that it can operate within resource-constrained environments without significant loss in accuracy. This makes it particularly suitable for deployment in digital health applications where real-time processing is crucial.

Furthermore, the reduced computational demand of DistilBERT not only enables faster processing times but also decreases the overall resource requirements. This aspect makes advanced sentiment analysis more accessible to organizations and platforms that may not have extensive computational power [25]. The high degree of linguistic understanding maintained by DistilBERT allows for precise interpretation of complex emotional expressions often found in mental health-related communications.

In this project, DistilBERT is employed specifically to analyze sentiment within user communications, not to generate responses. This focused application enables users to accurately and consistently understand their own emotional tones, forming the basis for any subsequent human intervention or guidance provided by mental health professionals. The next section of this review will explore specific instances and empirical evidence demonstrating the effectiveness of DistilBERT in capturing and analyzing sentiment in the mental health context, thereby underscoring its valuable contributions to enhancing digital mental health support systems.

2.6.2 Review of DialoGPT for response generation

DialoGPT, short for 'Dialogue Generative Pre-trained Transformer,' is a model developed by Microsoft, building on the architecture of OpenAI's GPT-2. It is specifically designed to generate human-like conversational responses. This adaptation utilizes the transformer architecture, which is renowned for its effectiveness in processing and generating text through its use of stacked decoder blocks. These blocks are specially engineered to produce coherent and contextually relevant outputs, making DialoGPT highly adept at managing dialogue sequences. As a result, DialoGPT excels in both understanding and generating responses that are contextually appropriate, enhancing its utility in conversational applications.

The design of DialoGPT makes it exceptionally suitable for applications such as mental health support chatbots, where generating empathetic and accurate responses is crucial. These chatbots require the ability to engage in meaningful dialogue with users, understanding their concerns and responding in a manner that supports their emotional well-being. DialoGPT’s training on diverse conversational data allows it to grasp subtle nuances in dialogue, making it an ideal candidate for such sensitive interactions. According to Greyling, Microsoft's evaluation demonstrated that sentences generated by DialoGPT are diverse and contextually relevant to the source prompts, showcasing the model's ability to accurately interpret and respond to varied conversational inputs [26].

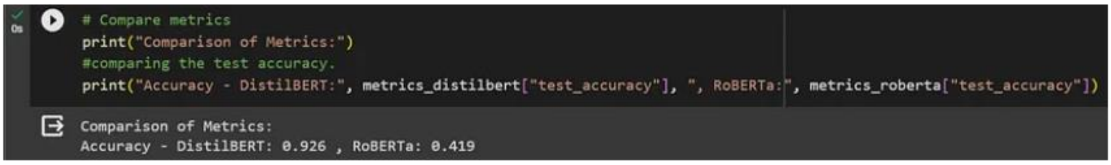
The performance of DialoGPT in generating dialogue is enhanced by its deep learning model, which can handle extensive conversational datasets without significant losses in speed or efficiency. The reason is because it trained by a massive dataset which in a range of 147 million conversational example [27]. This makes it capable of offering real-time interactions, essential for maintaining the flow of conversation in mental health support scenarios. The model’s robustness, stemming from its extensive pre-training on internet-scale datasets [25], ensures that it can adapt to the varied linguistic styles and emotional tones encountered in mental health discussions.

Additionally, DialoGPT's adaptability allows it to be fine-tuned for specific tasks within the mental health domain. By training on specialized datasets that include therapeutic dialogue or crisis intervention exchanges, DialoGPT can further refine its response generation to be sensitive and contextually appropriate for mental health support. This focused application significantly enhances the chatbot's utility, providing users with interactions that feel genuine and supportive. In this project, DialoGPT is specifically utilized to generate responses within a mental health support chatbot, highlighting the model's capability to produce responses that are not only contextually aware but also empathetically aligned with users' emotional states. The following section will delve into case studies and empirical data illustrating the effectiveness of DialoGPT in real-world applications, thereby demonstrating its pivotal role in advancing digital mental health solutions.

2.6.3 Evaluation of DistilBert

This section explores empirical evaluations that demonstrate the practical efficacy of DistilBERT in analyzing sentiment within mental health communications.

Reviewing various case studies and research findings illustrates how DistilBERT not only meets theoretical expectations but also excels in real-world applications DistilBERT has been fine-tuned using specific datasets designed for sentiment analysis, making it particularly adept at recognizing a wide range of emotions that reflect real-world scenarios. For instance, an article by Jethani discusses the model's fine-tuning with an emotion dataset, highlighting that DistilBERT achieved higher accuracy than RoBERTa [28]. Moreover, Eshun reported in an article on Medium that the fine-tuned DistilBERT model achieved an accuracy and an F1 score of approximately 0.78 [29]. The versatility of DistilBERT is further demonstrated by its adaptability to various tasks, which can be achieved by fine-tuning it with more specific datasets as needed. Oiro, a technical writer on LinkedIn, mentioned that DistilBERT's sentiment analysis could serve multiple purposes, such as tracking customer sentiment feedback for a shop, identifying harmful content on social media for removal, and more [30]. These applications underscore DistilBERT's value to society.

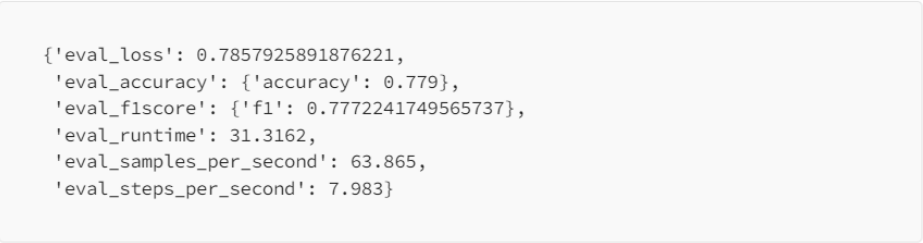


```
# Compare metrics
print("Comparison of Metrics:")
#comparing the test accuracy.
print("Accuracy - DistilBERT:", metrics_distilbert["test_accuracy"], ", RoBERTa:", metrics_roberta["test_accuracy"])
```

```
Comparison of Metrics:
Accuracy - DistilBERT: 0.926 , RoBERTa: 0.419
```

Comparison between test accuracy

Figure 2.12: Sample output from Jethani's article

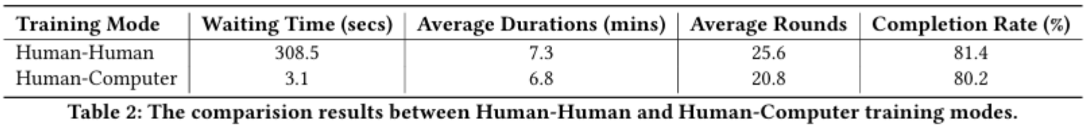


```
{'eval_loss': 0.7857925891876221,
 'eval_accuracy': {'accuracy': 0.779},
 'eval_f1score': {'f1': 0.7772241749565737},
 'eval_runtime': 31.3162,
 'eval_samples_per_second': 63.865,
 'eval_steps_per_second': 7.983}
```

Figure 2.13: Model Evaluating output from Eshun's article

2.6.4 Evaluation of DialoGPT

This section highlights empirical evaluations that underscore the effectiveness of DialoGPT in facilitating human-like conversations across various platforms, with a particular focus on its utility in customer service and mental health support. Case studies and research findings demonstrate how DialoGPT not only meets but often surpasses theoretical expectations, excelling in practical applications. DialoGPT has been extensively fine-tuned using large-scale conversational datasets from sources such as Reddit, which encompass a broad spectrum of human interactions, enhancing its ability to manage complex and nuanced dialogues. For example, a study by Zhang et al. illustrates that DialoGPT's responses are more contextually relevant and conversationally appropriate compared to earlier models like GPT-2 [31]. Additionally, Lubelski's article highlighted DialoGPT's real-world capabilities, noting its effective use in fields such as customer service, mental health, and education [32]. The adaptability of DialoGPT is showcased in its diverse applications. For instance, AdaCoach, a virtual coaching system developed by the Ant Group using DialoGPT, demonstrated superior performance to traditional human-to-human training modes in metrics such as waiting time, training duration, and average rounds per dialogue [33]. These real-world applications highlight DialoGPT's transformative impact in reshaping conversational AI. By leveraging its advanced generative capabilities, DialoGPT not only enhances the quality of user interactions but also extends the reach of AI's benefits across various sectors, improving how businesses and health services engage with their clients and patients, respectively.



Training Mode	Waiting Time (secs)	Average Durations (mins)	Average Rounds	Completion Rate (%)
Human-Human	308.5	7.3	25.6	81.4
Human-Computer	3.1	6.8	20.8	80.2

Table 2: The comparision results between Human-Human and Human-Computer training modes.

Figure 2.14: Evaluation Metrics of AdaCoac

2.7 Review on Chosen Backend Technology: Flask

Flask is a web framework for Python designed to simplify the process of building applications, APIs, and back-end services. It is well-suited for this project due to its ability to handle RESTful requests in Python, especially for AI models. Common uses of Flask include developing web applications, APIs, prototypes, and small to medium-sized websites, offering an interactive and user-friendly experience [34]. While there are several frameworks available for Python developers—such as Django, FastAPI, and Ruby—Flask stands out due to its flexibility and simplicity.

Compared to Django, Flask is ideal for smaller projects because of its minimalistic design. Django, though powerful, tends to be more complex, and the codebase is often larger, which can lead to compatibility issues with other technologies [35]. In this project, where the chatbot feature requires quick responses, an efficient framework is crucial. Flask's lightweight nature ensures rapid load times for simple applications, making it an excellent choice [36].

Additionally, Flask integrates well with Python libraries, which is important for deep learning models that frequently interact with Python-based tools. Its modular design allows developers to extend its functionality with plugins and third-party libraries, enabling multiple features without overwhelming the project [36].

2.8 Review on Chosen Front-end Technology: Flutter

Flutter is an open-source software development toolkit from Google designed for creating user interfaces across various platforms, including mobile, web, and desktop. Flutter is an excellent choice for mobile application development due to its strong emphasis on delivering a smooth user interface and a superior user experience. It supports extensive customization even with complex UI designs and provides a variety of libraries to help developers create high-fidelity user interfaces [37]. So that the UI and UX design can be fulfill when using Flutter as the front-end tools.

Additionally, Flutter leverages the Dart programming language, which streamlines the management of asynchronous tasks. Compared to languages like JavaScript, Dart typically offers higher performance and faster execution [38]. Using Dart with Flutter can accelerate development due to its ease of learning and efficient handling of asynchronous operations. Furthermore, Flutter provides a range of features including an extensive library, versatile compilation options, and strong security features [39]. The large and active community also contributes to its robustness. These features work together to ensure that Flutter-built applications deliver high performance.

Moreover, Flutter enhances development efficiency for both Android and iOS by using a unified codebase. Features such as "hot reload" and "hot restart" further accelerate the development process, making it faster compared to other cross-platform tools like React Native and Xamarin [40].

Chapter 3
System Methodology/Approach OR System Model

This chapter explains the system design of the mental health mobile application.

3.1 System Design Diagram/Equation

The system integrates chatbot interactions, sentiment analysis, self-screening, and helpline functionalities. It consists of several components, including a Flutter-based frontend, Flask API backend, AI models (DistilBERT for sentiment analysis and DialoGPT for chatbot interactions), and a Dockerized deployment on Google Cloud Platform (GCP). This chapter provides diagrams and descriptions of the system's architecture, use cases, and user interaction flows, illustrating how different components communicate and work together to deliver real-time mental health support.

3.1.1 System Architecture Diagram

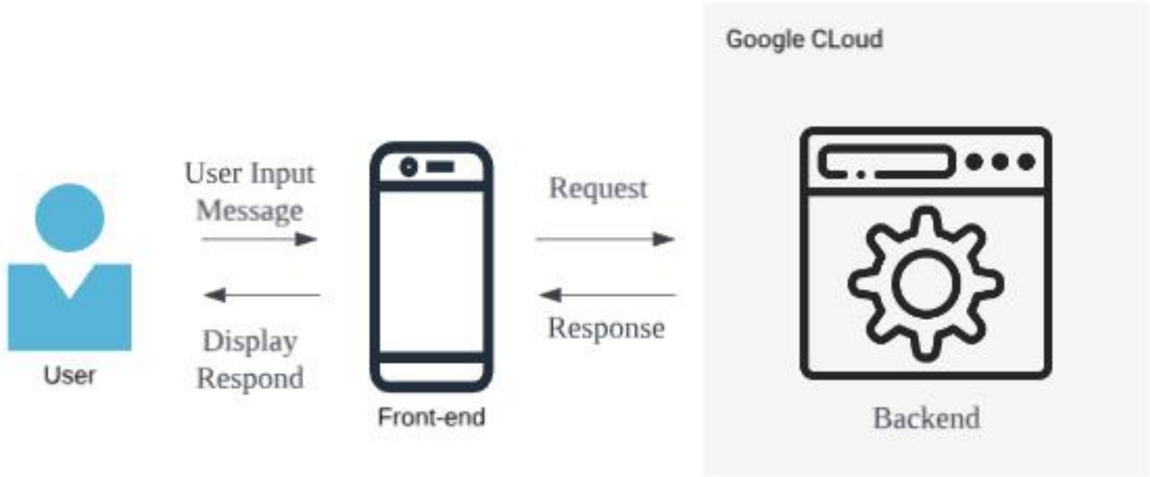


Figure 3.1: System Architecture Diagram

The architecture diagram illustrates the flow of communication between the user, the front-end, and the backend. The mobile app, developed using **Flutter**, serves as the user interface, allowing users to interact with the chatbot, access self-screening tests, and utilize helpline services.

The **Flask backend** processes user requests by routing them to the appropriate AI model: **DialoGPT** for chatbot responses and **DistilBERT** for sentiment analysis. The backend has designated endpoints for handling these tasks and manages the flow of data between the frontend and the models.

All backend services are containerized using **Docker** and deployed on **Google Cloud Run** to ensure scalability and reliability. This cloud-based setup provides high availability, enabling the system to handle multiple user requests efficiently.

The request-response cycle ensures that user inputs sent from the app are processed in real-time and returned to the frontend for display. This architecture allows for seamless interaction between the app and the AI models, ensuring robust performance for all functionalities.

3.1.2 Use Case Diagram and Description

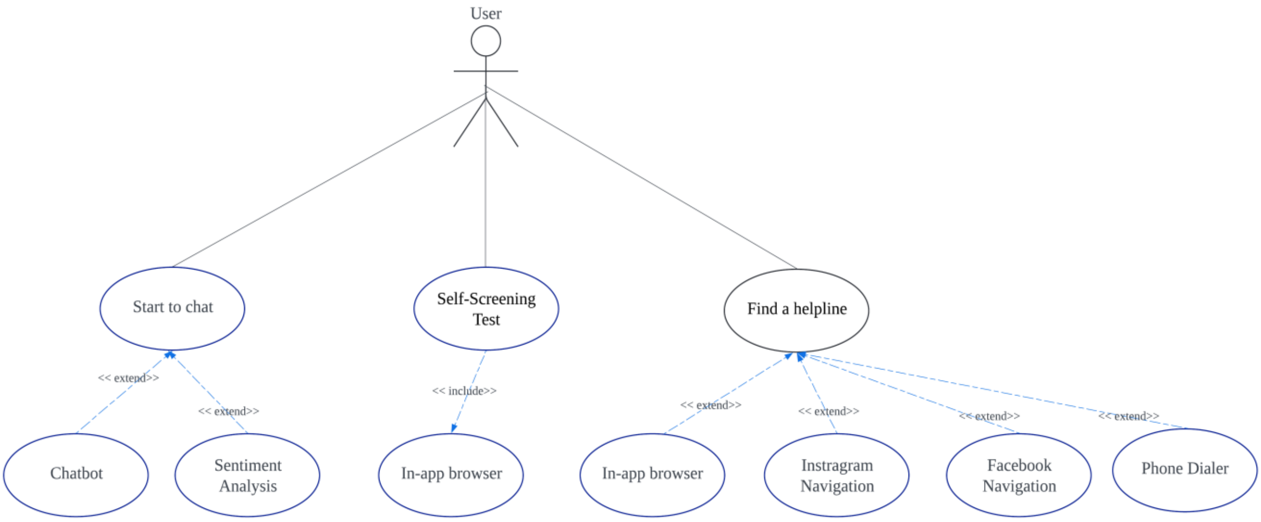


Figure 3.2: Use case diagram

The use case diagram captures the interaction between the user and the mobile application, highlighting key functionalities such as chatbot interaction, sentiment analysis, self-screening tests, and helpline navigation.

In the **Chatbot Interaction** use case, the user submits a message via the chatbox, which is forwarded to the Flask backend. The backend performs two actions: generating a chatbot response using **DialoGPT** and conducting **sentiment analysis** using **DistilBERT** to classify the emotional sentiment of the message (e.g., sadness, joy). The chatbot's response and the detected sentiment are both displayed to the user.

For the **Self-Screening Test** use case, users can access a mental health self-screening test hosted externally, such as the **Malaysia Mentari** website, through an in-app browser. The user receives feedback based on the test results.

In the **Helpline Navigation** use case, the app provides access to mental health resources, including websites, social media pages (e.g., Instagram and Facebook), and a phone call feature to contact **UTAR Hospital's helpline** directly. This comprehensive design ensures users can engage with both automated and human resources for mental health support.

3.1.3 Activity Diagram

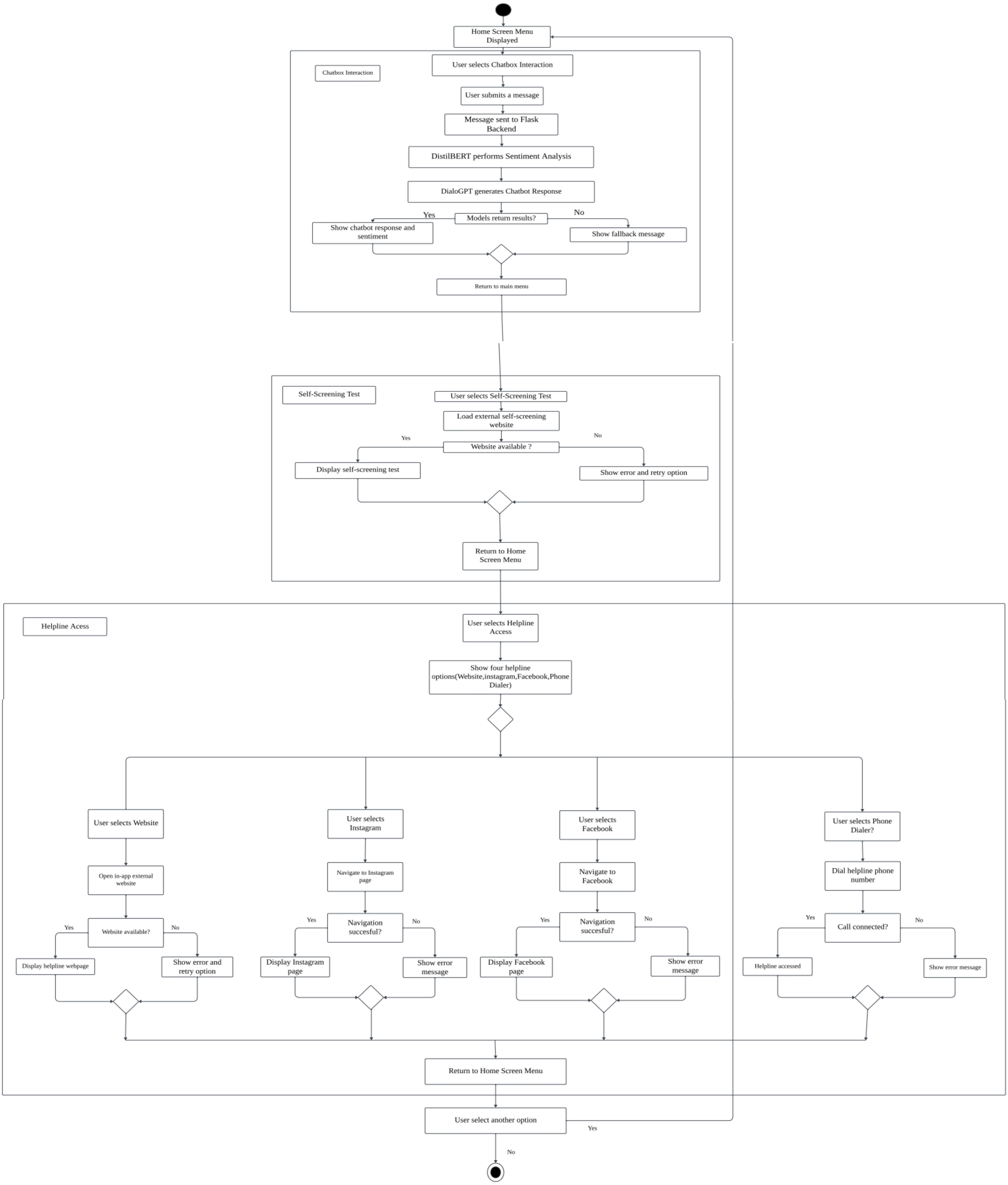

Home Screen Menu Displayed
Chatbox Interaction
User selects Chatbox Interaction
User submits a message
Message sent to Flask Backend
DistilBERT performs Sentiment Analysis
DialoGPT generates Chatbot Response
Yes
Models return results?
No
Show chatbot response and sentiment
Show fallback message
Return to main menu
Self-Screening Test
User selects Self-Screening Test
Load external self-screening website
Yes
Website available ?
No
Display self-screening test
Show error and retry option
Return to Home Screen Menu
Helpline Acess
User selects Helpline Access
Show four helpline options(Website,instagram,Facebook,Phone Dialer)
User selects Website
Open in-app external website
Yes
Website available?
No
Display helpline webpage
Show error and retry option
User selects Instagram
Navigate to Instagram page
Yes
Navigation succesful?
No
Display Instagram page
Show error message
User selects Facebook
Navigate to Facebook
Yes
Navigation succesful?
No
Display Facebook page
Show error message
User selects Phone Dialer?
Dial helpline phone number
Yes
Call connected?
No
Helpline accessed
Show error message
Return to Home Screen Menu
User select another option
Yes
No

Figure 3.3: Activity Diagram

This activity diagram represents the user interaction flow for the mental health app, highlighting three primary functionalities: **Chatbox Interaction**, **Self-Screening Test**, and **Helpline Access**. The diagram outlines the following processes:

1. **Chatbox Interaction**:
   - The user submits a message via the chatbox interface.
   - The message is sent to the Flask backend, where two key processes occur:
     - **DialoGPT** generates the chatbot's conversational response.
     - **DistilBERT** performs sentiment analysis on the user's message to assess their emotional state.
   - The app either displays the chatbot response and the sentiment analysis or, in case of failure, shows a fallback message.
   - After interacting with the chatbot, the user can return to the main menu.

2. **Self-Screening Test**:
   - The user selects the self-screening test option, which loads an external website for conducting the test.
   - If the website is available, it is displayed within the app.
   - In case the website is unavailable, the app shows an error message and provides an option to retry.
   - Once the test is complete, the user can return to the home screen.

3. **Helpline Access**:
   - The user can access helpline resources through four options: Website, Instagram, Facebook, or Phone Dialer.
   - Each selection triggers a navigation or call attempt:
     - For the website, Instagram, and Facebook options, the app checks if the page loads successfully. If successful, the user can interact with the page; otherwise, an error message and retry option are displayed.
     - For the phone dialer, the app initiates a call to the helpline. If the call connects, the helpline is accessed. Otherwise, an error message is shown.

- After accessing the helpline, the user can return to the home screen or choose another option.

Each activity ensures proper error handling, providing a seamless user experience with fallback options if a resource is unavailable.

Chapter 4

System Design

In this chapter, the system design and architecture are explained. The System Block Diagram shows data flow between the frontend, backend, and AI models. The Sequence Diagram details how user interactions with the chatbot are processed. The System Components Specifications outline the main tools, models, and infrastructure used. Finally, the Circuits and Components Design describes the communication between the mobile app, Flask backend, and Dockerized AI models, ensuring a reliable system.

4.1 System Block Diagram

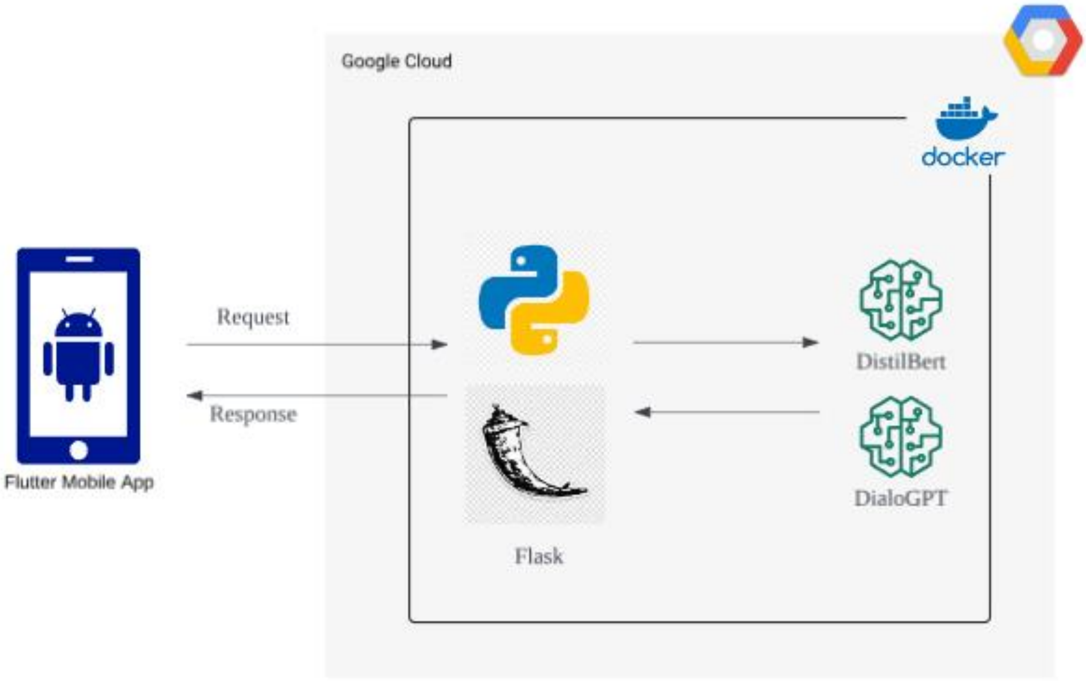


Figure 4.1: System Block Diagram

The system block diagram accurately represents the flow of data between the Flutter frontend, the Flask backend, and the AI models (DistilBERT and DialoGPT). The Flutter mobile app serves as the user interface, allowing users to interact with the system by sending requests, such as messages for the chatbot or text for sentiment analysis. The Flask backend processes these requests from the app and routes them to the appropriate AI model. Depending on the request, Flask either sends the input to DistilBERT for sentiment analysis, which classifies the emotional tone of the input, or to DialoGPT, which generates conversational responses for the chatbot. The backend, along with the AI models, is containerized using Docker and deployed on Google Cloud Run, providing scalability, reliability, and fault tolerance to handle multiple user requests efficiently. In the request-response cycle, the Flutter app sends the user's request to the Flask backend, which processes it and forwards the relevant data to the AI models. Once the data is processed, the response is sent back to the Flutter app and displayed to the user. This architecture enables real-time chatbot interactions, sentiment analysis, and access to other mental health support services through a robust cloud-based infrastructure.

4.2 Sequence Diagram for the Chatbot

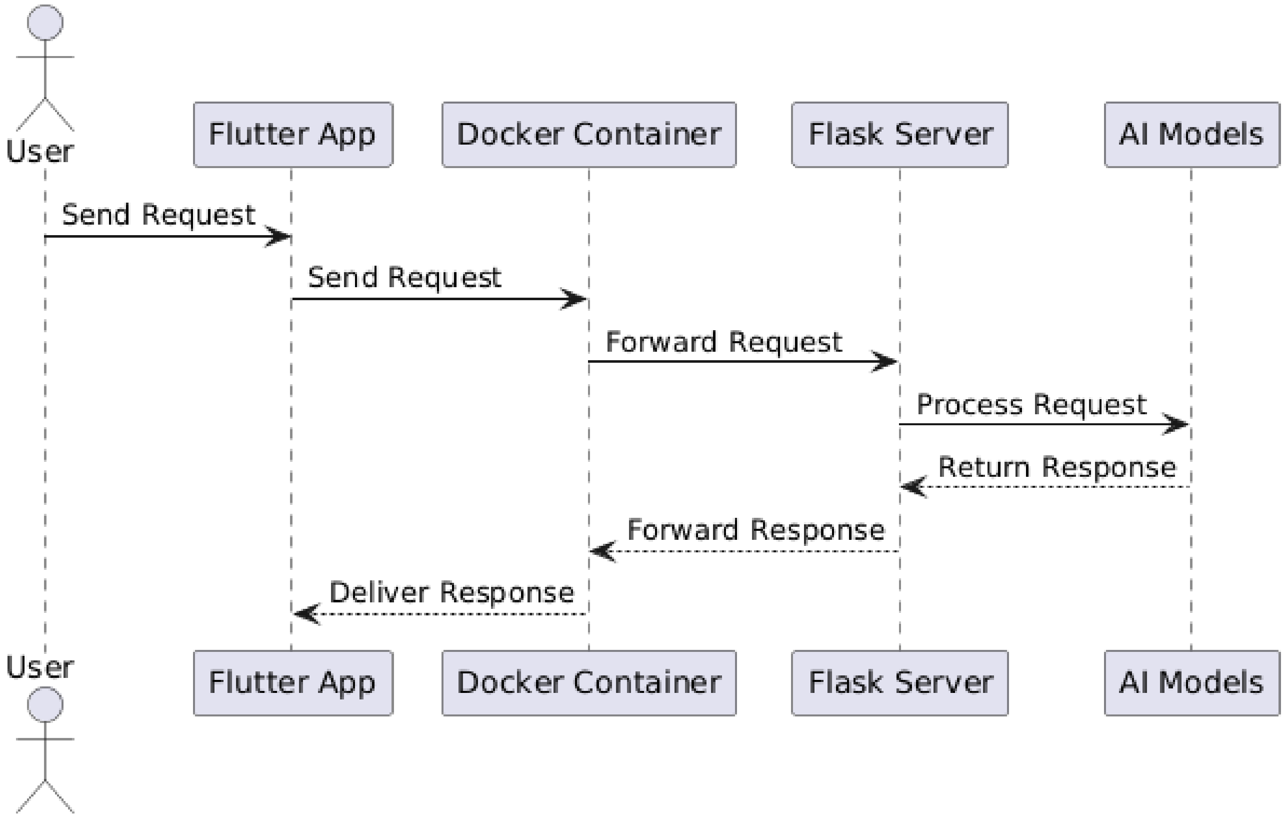


Figure 4.2: Sequence Diagram of Chatbox

The sequence diagram illustrates the interaction between the Flutter app, Docker container, Flask server, and AI models during chatbot usage. When a user submits a message, the request is sent to the Flask backend, which routes it to DialoGPT for a chatbot response or DistilBERT for sentiment analysis. The Flask server receives the processed data and sends it back to the app for display. This flow enables real-time interactions for chatbot and sentiment functionalities.

4.3 System Components Specifications

This section outlines the key components that make up the system:

- **Main AI Models**:
    - **DialoGPT**: Fine-tuned on mental health conversation data, this model generates empathetic and relevant chatbot responses.
    - **DistilBERT**: Focuses on sentiment analysis, detecting emotions like Sadness, Joy, Love, and Anger.
- **Frontend (Mobile App)**:
    - Built using Flutter, providing a simple and user-friendly interface. It allows interaction with the chatbot, access to self-screening tests, and helpline services.
- **Backend (Flask)**:
    - The Flask server manages communication between the frontend and the AI models, processing user inputs and returning results in real-time.
- **Cloud Deployment**:
    - Deployed on Google Cloud Run using Docker containers, the system is scalable and ensures fault tolerance, with monitoring for optimal performance.

4.4 Circuits and Components Design

This system is entirely software-based, with a focus on seamless interaction between its components. The frontend Flutter app communicates with the Flask backend, which routes user requests to the AI models for sentiment analysis (DistilBERT) and chatbot responses (DialoGPT). Docker is used to containerize the backend and AI models, ensuring consistent deployment and scalability. The design enables real-time interactions between the app and backend services without requiring any physical circuit designs.

4.5 System Components Interaction Operations

The operation of the system hinges on smooth interactions between its key components:

- **Chatbot and Sentiment Analysis**: When a user sends a message, the Flask backend routes it to DialoGPT for response generation and DistilBERT for sentiment analysis. Results are then returned to the user through the Flutter interface. If any part of the process fails, fallback messages are displayed to maintain user experience.
- **Self-Screening and Helpline Access**: For self-screening tests, the user can access external websites via the in-app browser. Helpline access allows navigation to social media platforms or direct phone calls, ensuring easy access to mental health resources.
- **Monitoring and Performance Tracking**: Google Cloud Monitoring is employed to track system performance and error logging, ensuring issues are identified and resolved quickly.

This design ensures a seamless user experience by integrating multiple components into a cohesive system that provides real-time mental health support and resource access.

Chapter 5
System Implementation

In this chapter, the implementation and configuration of the mental health support system are detailed. It covers the hardware and software setups, including the mobile app and Flask backend. Key aspects like AI model fine-tuning, Docker containerization, and cloud deployment are explained. The chapter also describes system operations, including chatbot interactions, sentiment analysis, and in-app browser functionalities. Challenges faced during implementation are discussed, and the chapter concludes with a summary of the system's successful deployment.

5.1 Hardware Setup

Since this project focuses on software development and machine learning model deployment, the hardware chosen includes a personal computer for development and testing devices for app compatibility. The specifications are as follows:

Description	**Specifications**
Model	HP Laptop 15s-eq1180AU
Processor	AMD Ryzen 5 4500U
Operating System	Windows 10
Graphic	Radeon Graphics 2.38 GHz
Memory	8.00 GB RAM
Storage	512GB SSD

Table 5.1: Laptop Specifications

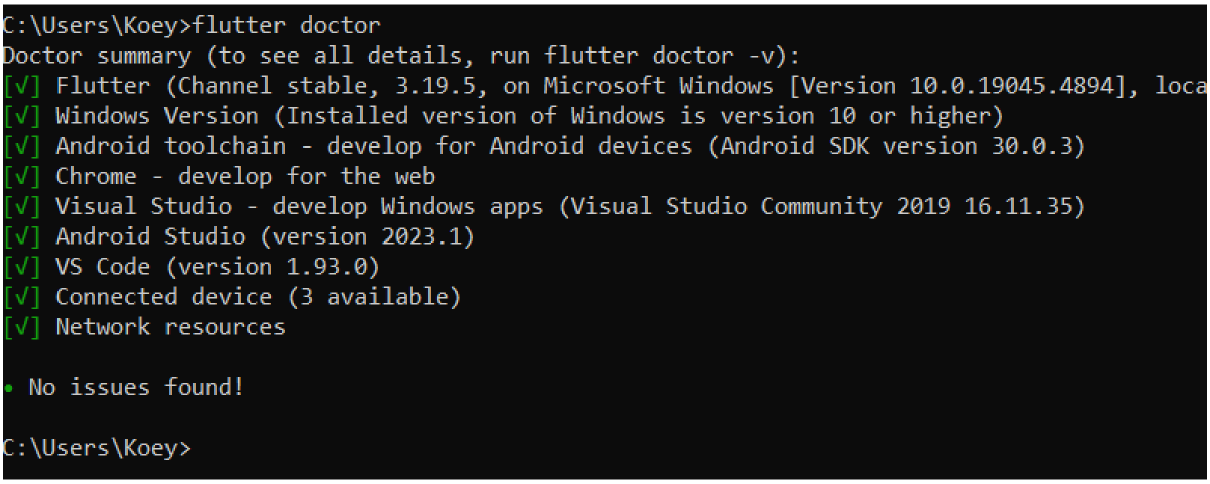
Description	**Specifications**
Model	Vivo Y77 5G
Processor	2.4 GHz Dimensity 810 Octa-core
Android version	14
Memory	8 + 8 GB RAM
Storage	256 GB

Table 5.2: Mobile Device Specifications

5.2 Software Setup

The software setup for the mental health support system involves multiple tools, frameworks, and platforms that work together to ensure seamless functionality across the mobile app, backend, AI models, and deployment. The following are the key steps and components involved in the setup:

1. **Development Environment:**
   - **Flutter:** The mobile application is developed using Flutter, a cross-platform UI toolkit that enables smooth and responsive interfaces for both Android and iOS. To set up Flutter:
     - Install the Flutter SDK from *https://docs.flutter.dev/get-started/install*.
     - Use the *flutter doctor* command to verify the installation and resolve any setup issues.

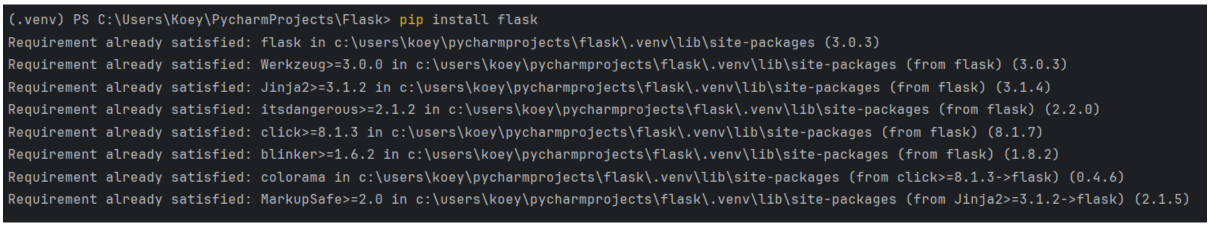


```
C:\Users\Koey>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[√] Flutter (Channel stable, 3.19.5, on Microsoft Windows [Version 10.0.19045.4894], loca
[√] Windows Version (Installed version of Windows is version 10 or higher)
[√] Android toolchain - develop for Android devices (Android SDK version 30.0.3)
[√] Chrome - develop for the web
[√] Visual Studio - develop Windows apps (Visual Studio Community 2019 16.11.35)
[√] Android Studio (version 2023.1)
[√] VS Code (version 1.93.0)
[√] Connected device (3 available)
[√] Network resources

• No issues found!

C:\Users\Koey>
```

Figure 5.1: Flutter Doctor Summary

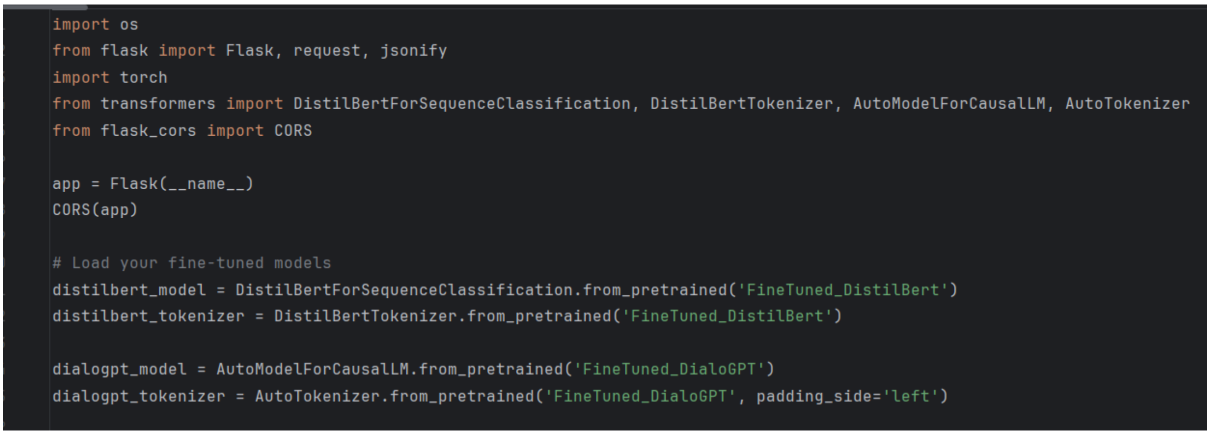
2. **Backend Setup:**
   - **Flask:** The backend is built using Flask, a lightweight Python web framework that handles user requests and communicates with the AI models. To set up Flask and run it in PyCharm:
     - Install Flask using pip with the command *pip install flask*.
     - Create an API with endpoints for chatbot interaction and sentiment analysis.
     - Utilize Flask’s routing capabilities to connect the app to the AI models (DialoGPT and DistilBERT).

- Open PyCharm, set up a new project or use an existing one, and configure the Flask environment. Run your Flask code directly within PyCharm to manage and test the backend functionality.



```
(.venv) PS C:\Users\Koey\PycharmProjects\Flask> pip install flask
Requirement already satisfied: flask in c:\users\koey\pycharmprojects\flask\.venv\lib\site-packages (3.0.3)
Requirement already satisfied: Werkzeug>=3.0.0 in c:\users\koey\pycharmprojects\flask\.venv\lib\site-packages (from flask) (3.0.3)
Requirement already satisfied: Jinja2>=3.1.2 in c:\users\koey\pycharmprojects\flask\.venv\lib\site-packages (from flask) (3.1.4)
Requirement already satisfied: itsdangerous>=2.1.2 in c:\users\koey\pycharmprojects\flask\.venv\lib\site-packages (from flask) (2.2.0)
Requirement already satisfied: click>=8.1.3 in c:\users\koey\pycharmprojects\flask\.venv\lib\site-packages (from flask) (8.1.7)
Requirement already satisfied: blinker>=1.6.2 in c:\users\koey\pycharmprojects\flask\.venv\lib\site-packages (from flask) (1.8.2)
Requirement already satisfied: colorama in c:\users\koey\pycharmprojects\flask\.venv\lib\site-packages (from click>=8.1.3->flask) (0.4.6)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\koey\pycharmprojects\flask\.venv\lib\site-packages (from Jinja2>=3.1.2->flask) (2.1.5)
```

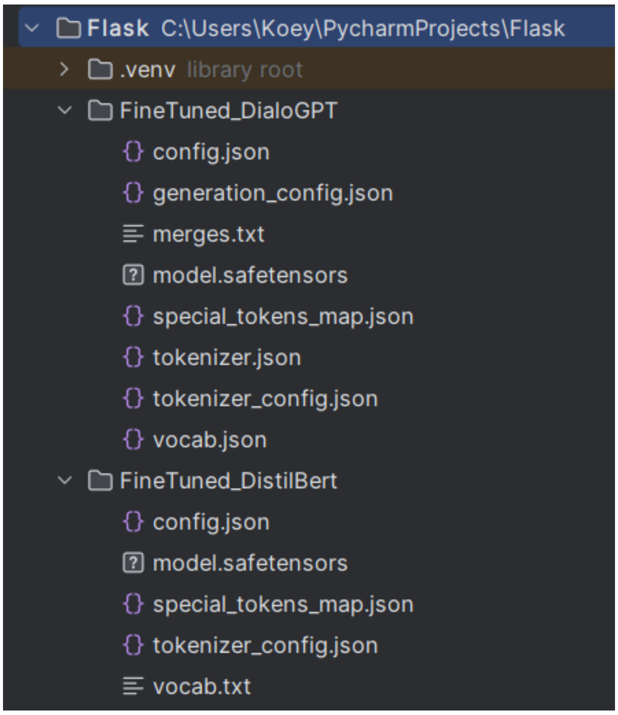
Figure 5.2: Screenshot of the Flask Installation



```python
import os
from flask import Flask, request, jsonify
import torch
from transformers import DistilBertForSequenceClassification, DistilBertTokenizer, AutoModelForCausalLM, AutoTokenizer
from flask_cors import CORS

app = Flask(__name__)
CORS(app)

# Load your fine-tuned models
distilbert_model = DistilBertForSequenceClassification.from_pretrained('FineTuned_DistilBert')
distilbert_tokenizer = DistilBertTokenizer.from_pretrained('FineTuned_DistilBert')

dialogpt_model = AutoModelForCausalLM.from_pretrained('FineTuned_DialoGPT')
dialogpt_tokenizer = AutoTokenizer.from_pretrained('FineTuned_DialoGPT', padding_side='left')
```

Figure 5.3: Screenshot of the Flask code

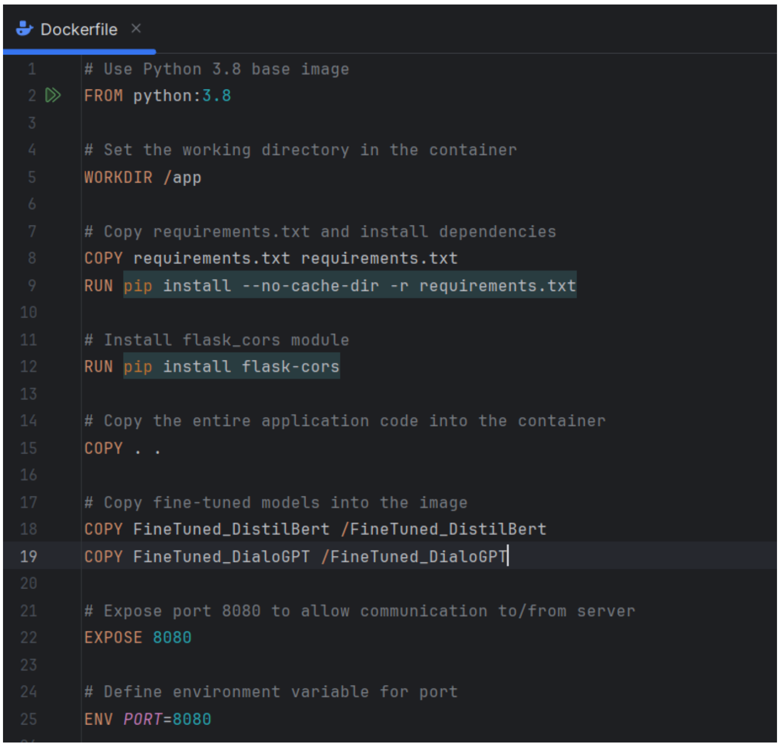
3. **AI Model Integration:**
   - **DialoGPT and DistilBERT:** These models are integrated into the Flask backend to handle chatbot responses and sentiment analysis.
     - Install the Hugging Face Transformers library using pip install transformers.
     - Load pre-trained models for DialoGPT and DistilBERT and fine-tuned them on mental health conversation datasets.
     - Ensure that the Flask backend can communicate with these models for real-time processing of user inputs.



```
Flask C:\Users\Koey\PycharmProjects\Flask
  .venv library root
  FineTuned_DialoGPT
    config.json
    generation_config.json
    merges.txt
    model.safetensors
    special_tokens_map.json
    tokenizer.json
    tokenizer_config.json
    vocab.json
  FineTuned_DistilBert
    config.json
    model.safetensors
    special_tokens_map.json
    tokenizer_config.json
    vocab.txt
```

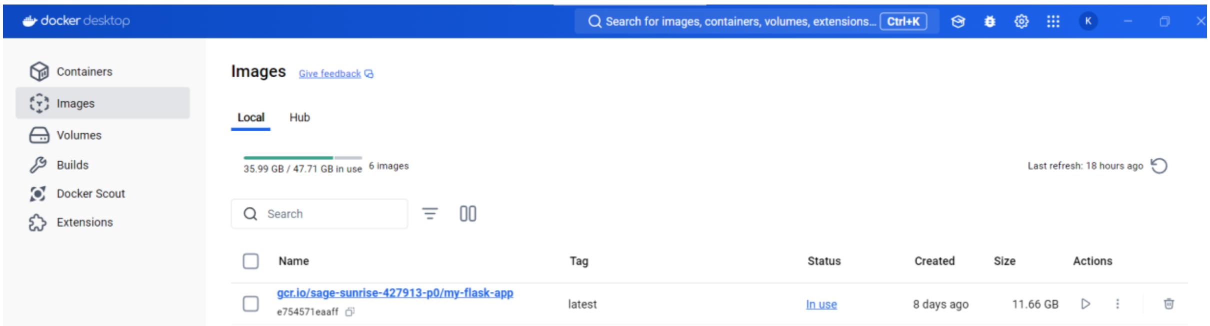
Figure 5.4: Saving the fine-tuned models in the same folder as the Flask code

4. **Containerization:**
   - **Docker:** The entire backend, including the Flask server and AI models, is containerized using Docker for easy deployment.
     - Install Docker from *https://www.docker.com/products/docker-desktop/*.
     - Create a Dockerfile to define the environment for the Flask app and AI models, including necessary libraries and dependencies.
     - Use docker build to create an image of the backend and docker run to test it locally.



```dockerfile
# Use Python 3.8 base image
FROM python:3.8

# Set the working directory in the container
WORKDIR /app

# Copy requirements.txt and install dependencies
COPY requirements.txt requirements.txt
RUN pip install --no-cache-dir -r requirements.txt

# Install flask_cors module
RUN pip install flask-cors

# Copy the entire application code into the container
COPY . .

# Copy fine-tuned models into the image
COPY FineTuned_DistilBert /FineTuned_DistilBert
COPY FineTuned_DialoGPT /FineTuned_DialoGPT

# Expose port 8080 to allow communication to/from server
EXPOSE 8080

# Define environment variable for port
ENV PORT=8080
```

Figure 5.5: DockerFile



35.99 GB / 47.71 GB in use 6 images

Last refresh: 18 hours ago

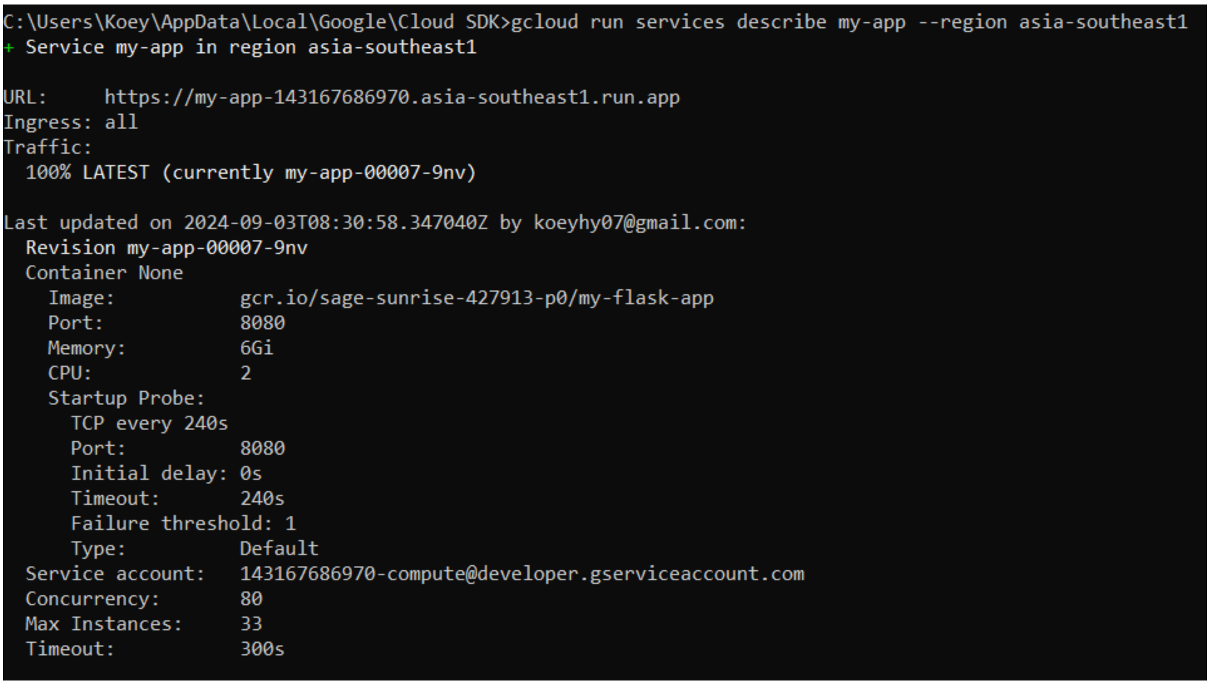
□	Name	Tag	Status	Created	Size	Actions
□	gcr.io/sage-sunrise-427913-p0/my-flask-app e754571eaaff	latest	In use	8 days ago	11.66 GB	▷ ⋮ 🗑

Figure 5.6: Docker Imgae

5. **Cloud Deployment:**

    ○ **Google Cloud Run:** The Dockerized backend is deployed on Google Cloud Run to ensure scalability and high availability.

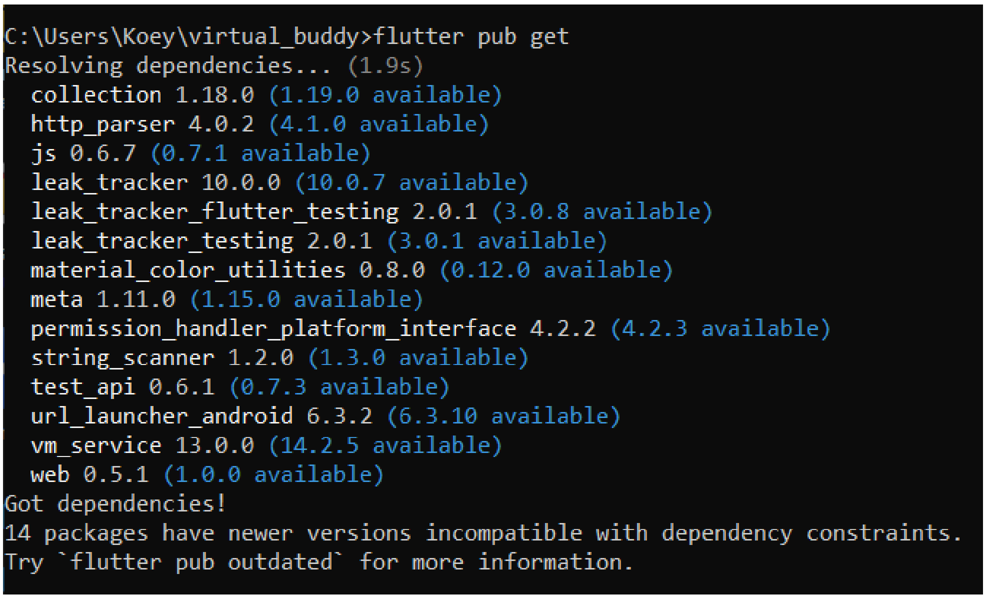
    - Set up a Google Cloud account and enable Cloud Run.
    - Push the Docker image to Google Container Registry using docker push.
    - Deploy the image on Cloud Run with *gcloud run deploy*, ensuring that the app is scalable and accessible via HTTP endpoints.



```
C:\Users\Koey\AppData\Local\Google\Cloud SDK>gcloud run services describe my-app --region asia-southeast1
+ Service my-app in region asia-southeast1

URL:     https://my-app-143167686970.asia-southeast1.run.app
Ingress: all
Traffic:
  100% LATEST (currently my-app-00007-9nv)

Last updated on 2024-09-03T08:30:58.347040Z by koeyhy07@gmail.com:
  Revision my-app-00007-9nv
  Container None
    Image:           gcr.io/sage-sunrise-427913-p0/my-flask-app
    Port:            8080
    Memory:          6Gi
    CPU:             2
    Startup Probe:
      TCP every 240s
      Port:          8080
      Initial delay: 0s
      Timeout:       240s
      Failure threshold: 1
      Type:          Default
  Service account:   143167686970-compute@developer.gserviceaccount.com
  Concurrency:       80
  Max Instances:     33
  Timeout:           300s
```

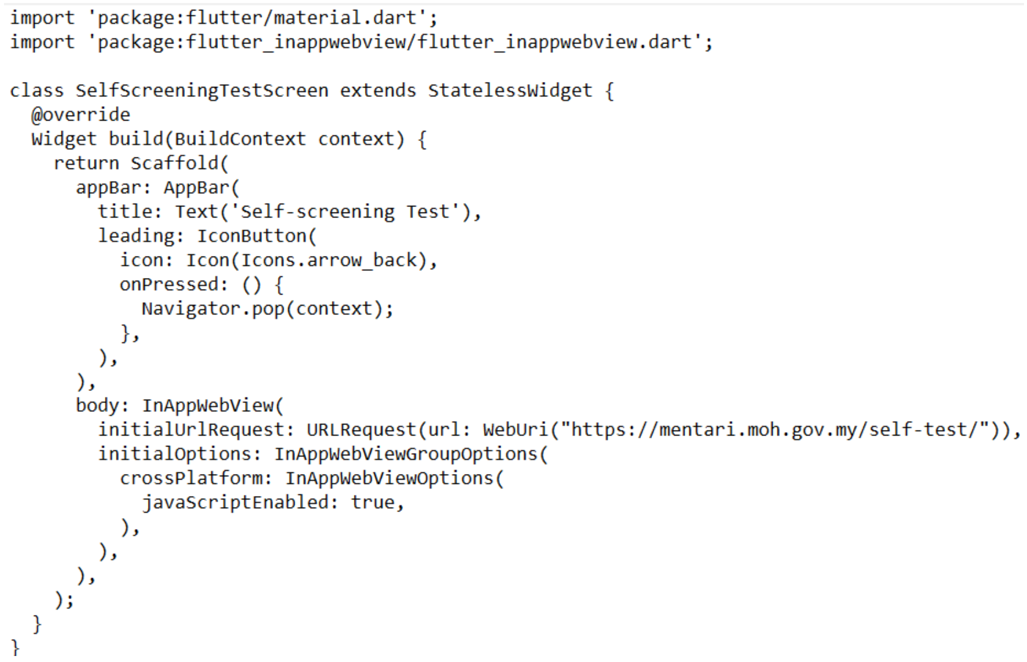
Figure 5.7: Google Cloud Service Information

6. **In-App Browser and External Links:**
   - **In-App Browser:** The mobile app includes an in-app browser to allow users to access self-screening tests and helpline resources.
     - Use the *url_launcher* package in Flutter to handle external navigation within the app.
     - Ensure links to external resources like websites and social media platforms are properly configured.



```
C:\Users\Koey\virtual_buddy>flutter pub get
Resolving dependencies... (1.9s)
  collection 1.18.0 (1.19.0 available)
  http_parser 4.0.2 (4.1.0 available)
  js 0.6.7 (0.7.1 available)
  leak_tracker 10.0.0 (10.0.7 available)
  leak_tracker_flutter_testing 2.0.1 (3.0.8 available)
  leak_tracker_testing 2.0.1 (3.0.1 available)
  material_color_utilities 0.8.0 (0.12.0 available)
  meta 1.11.0 (1.15.0 available)
  permission_handler_platform_interface 4.2.2 (4.2.3 available)
  string_scanner 1.2.0 (1.3.0 available)
  test_api 0.6.1 (0.7.3 available)
  url_launcher_android 6.3.2 (6.3.10 available)
  vm_service 13.0.0 (14.2.5 available)
  web 0.5.1 (1.0.0 available)
Got dependencies!
14 packages have newer versions incompatible with dependency constraints.
Try `flutter pub outdated` for more information.
```

Figure 5.8: Flutter Dependencies

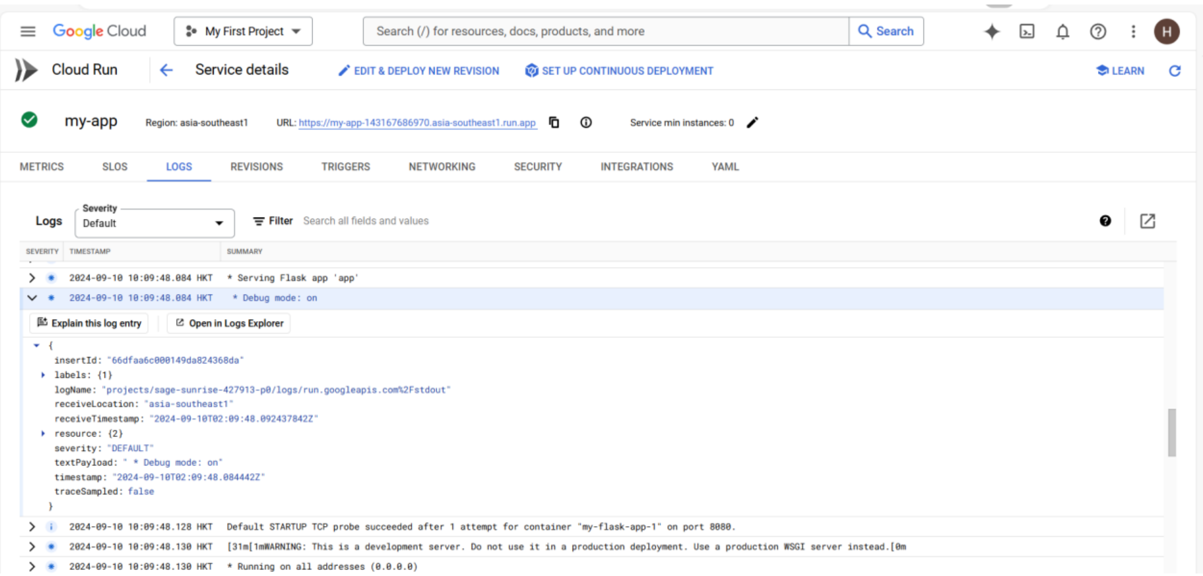


```
import 'package:flutter/material.dart';
import 'package:flutter_inappwebview/flutter_inappwebview.dart';

class SelfScreeningTestScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Self-screening Test'),
        leading: IconButton(
          icon: Icon(Icons.arrow_back),
          onPressed: () {
            Navigator.pop(context);
          },
        ),
      ),
      body: InAppWebView(
        initialUrlRequest: URLRequest(url: WebUri("https://mentari.moh.gov.my/self-test/")),
        initialOptions: InAppWebViewGroupOptions(
          crossPlatform: InAppWebViewOptions(
            javaScriptEnabled: true,
          ),
        ),
      ),
    );
  }
}
```

Figure 5.9: Dart code defining external links

7. **Monitoring and Error Tracking:**

   ○ **Google Cloud Run:** Use Cloud Logging for detailed error reports and troubleshooting.



my-app Region: asia-southeast1 URL: https://my-app-143167686970.asia-southeast1.run.app Service min instances: 0

Severity: Default

```
2024-09-10 10:09:48.084 HKT  * Serving Flask app 'app'
2024-09-10 10:09:48.084 HKT  * Debug mode: on
{
  insertId: "66dfaa6c000149da824368da"
  labels: {1}
  logName: "projects/sage-sunrise-427913-p0/logs/run.googleapis.com%2Fstdout"
  receiveLocation: "asia-southeast1"
  receiveTimestamp: "2024-09-10T02:09:48.092437842Z"
  resource: {2}
  severity: "DEFAULT"
  textPayload: " * Debug mode: on"
  timestamp: "2024-09-10T02:09:48.084442Z"
  traceSampled: false
}
2024-09-10 10:09:48.128 HKT  Default STARTUP TCP probe succeeded after 1 attempt for container "my-flask-app-1" on port 8080.
2024-09-10 10:09:48.130 HKT  [31m[1mWARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.[0m
2024-09-10 10:09:48.130 HKT  * Running on all addresses (0.0.0.0)
```

Figure 5.10: Google cloud app log

5.3 Setting and Configuration

The key configurations involved are as follows:

- **DistilBERT Training and Fine-tuning:** In FYP1, the DistilBERT model was fine-tuned on emotional classification datasets to enhance its ability to accurately identify user sentiment. This process involved adjusting the learning rate and hyperparameters to optimize classification accuracy.

- **DialoGPT Training and Fine-tuning:** In FYP1, the the DialoGPT model was fine-tuned on extensive conversational datasets to improve its capability to generate contextually relevant and coherent responses. Special emphasis was placed on tuning for emotional intelligence and coherence in responses.

- **Integration:** Both AI models are integrated into the Flask backend through REST APIs. This integration ensures real-time availability of sentiment analysis and chatbot functionalities during user interactions. The backend code, developed using PyCharm, includes:
  - **Sentiment Analysis Endpoint:** Processes text input to return sentiment results.
  - **Chatbot Endpoint:** Generates conversational responses based on user input.

- **Docker and Deployment:** Docker containers are utilized to package the Flask backend and AI models, facilitating deployment on Google Cloud Run. This setup supports scalable and efficient management of the system.

- **Libraries and Frameworks:**
  - **PyTorch:** Used for model implementation and training.
  - **Hugging Face Transformers:** Employed for working with pre-trained models and tokenizers.

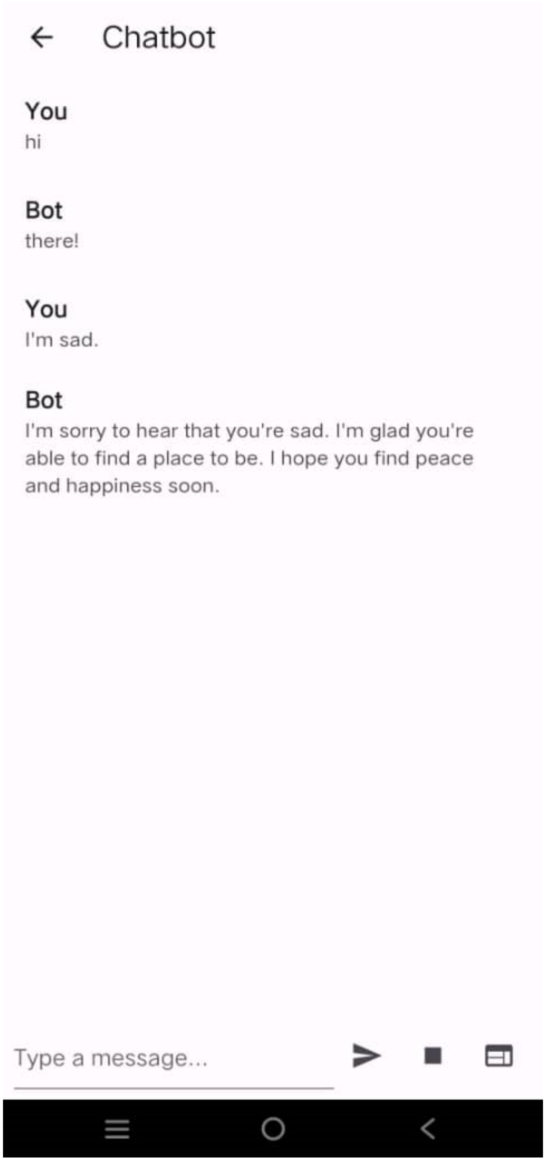
5.4 System Operation (with Screenshots)

5.4.1 Chatbot Interaction

When the user engages with the chatbot on the app, they are interacting with DialoGPT, which generates responses based on their input. For example:

User Input: "I’m sad."

Chatbot Response: " I'm sorry that you’re sad. I’m glad you’re able to find a place to be. I hope you find peace and happiness soon."



You
hi

Bot
there!

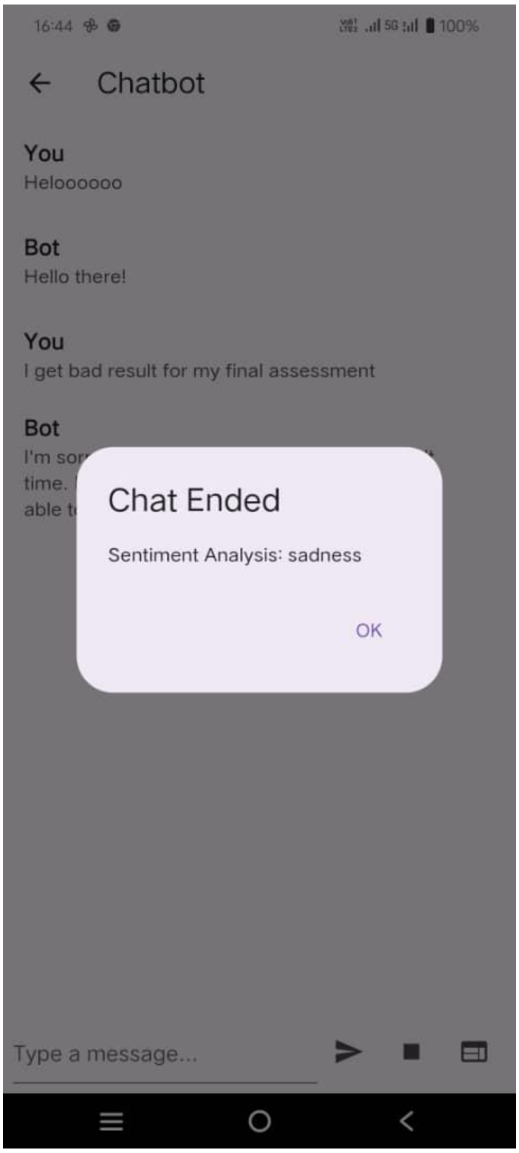
You
I'm sad.

Bot
I'm sorry to hear that you're sad. I'm glad you're able to find a place to be. I hope you find peace and happiness soon.

Figure 5.11: Screenshot of the chatbot

5.4.2 Sentiment Analysis Feedback

Then, DistilBERT processes the user input and identifies the sentiment when the user clicks on the “end chat” button. In this case, the result would be “Sentiment Analysis: sadness.”



You
Helooooo

Bot
Hello there!

You
I get bad result for my final assessment

Bot
I'm sor
time.
able t

Chat Ended

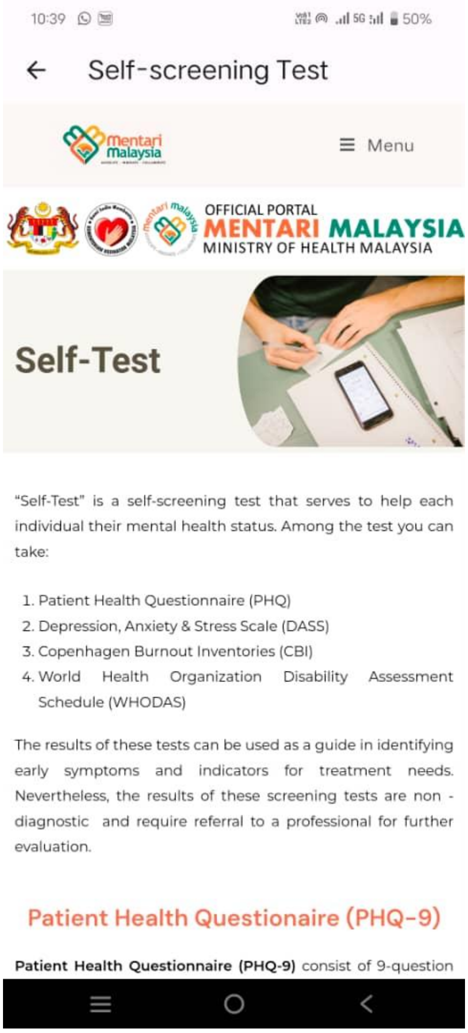
Sentiment Analysis: sadness

5.12: Screenshot of the sentiment result

5.4.3 In-App Browser for External Websites

When a user selects the self-screening test option or other external resources within the app, the in-app browser is activated. This browser view loads an external website, such as a self-screening mental health test from a specified URL. For example:

- User Action: The user taps on the self-screening test option.
- In-App Browser Action: The browser within the app opens and navigates to the external website for the test.
- External Website: Displays the self-screening questionnaire or relevant information, allowing users to complete the test without leaving the app.



Self-Test

"Self-Test" is a self-screening test that serves to help each individual their mental health status. Among the test you can take:

1. Patient Health Questionnaire (PHQ)
2. Depression, Anxiety & Stress Scale (DASS)
3. Copenhagen Burnout Inventories (CBI)
4. World Health Organization Disability Assessment Schedule (WHODAS)

The results of these tests can be used as a guide in identifying early symptoms and indicators for treatment needs. Nevertheless, the results of these screening tests are non - diagnostic and require referral to a professional for further evaluation.

Patient Health Questionaire (PHQ-9)

Patient Health Questionnaire (PHQ-9) consist of 9-question

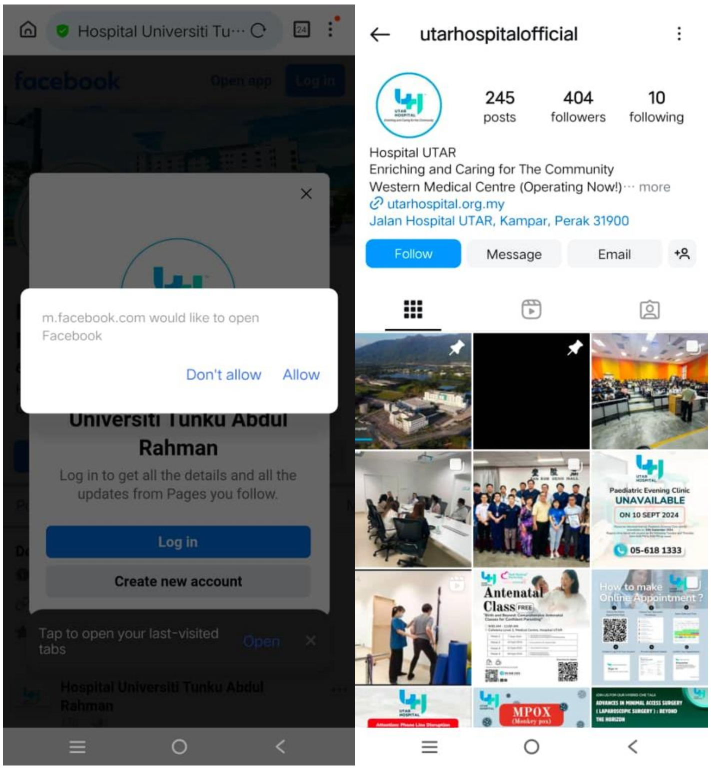
5.13: Screenshot of In-app Browser

5.4.4 Navigation to Other Apps

The app provides seamless navigation to various external resources through integrated buttons. These buttons enable users to:

• Navigate to social media: Tapping a button directs the user to social media platforms such as Facebook or Instagram. This is achieved by opening the respective app or website in the device's default browser.

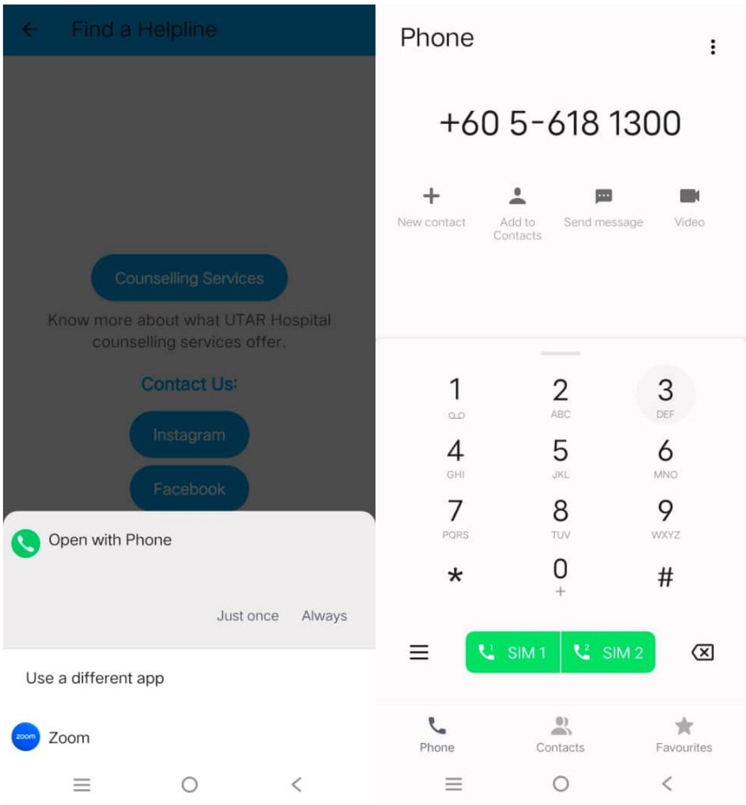
• Initiate Phone Calls: A dedicated button allows users to initiate phone calls directly to helplines or support services.



m.facebook.com would like to open Facebook

Hospital Universiti Tu···
facebook
Open app
Log in
Universiti Tunku Abdul Rahman
Log in to get all the details and all the updates from Pages you follow.
Don't allow
Allow
Log in
Create new account
Tap to open your last-visited tabs
Open
Hospital Universiti Tunku Abdul Rahman
utarhospitalofficial
245 posts
404 followers
10 following
Hospital UTAR
Enriching and Caring for The Community
Western Medical Centre (Operating Now!)··· more
utarhospital.org.my
Jalan Hospital UTAR, Kampar, Perak 31900
Follow
Message
Email
Paediatric Evening Clinic UNAVAILABLE ON 10 SEPT 2024
05-618 1333
Antenatal Class
How to make Online Appointment ?
MPOX

5.14 Screenshot of the Social Media Navigation



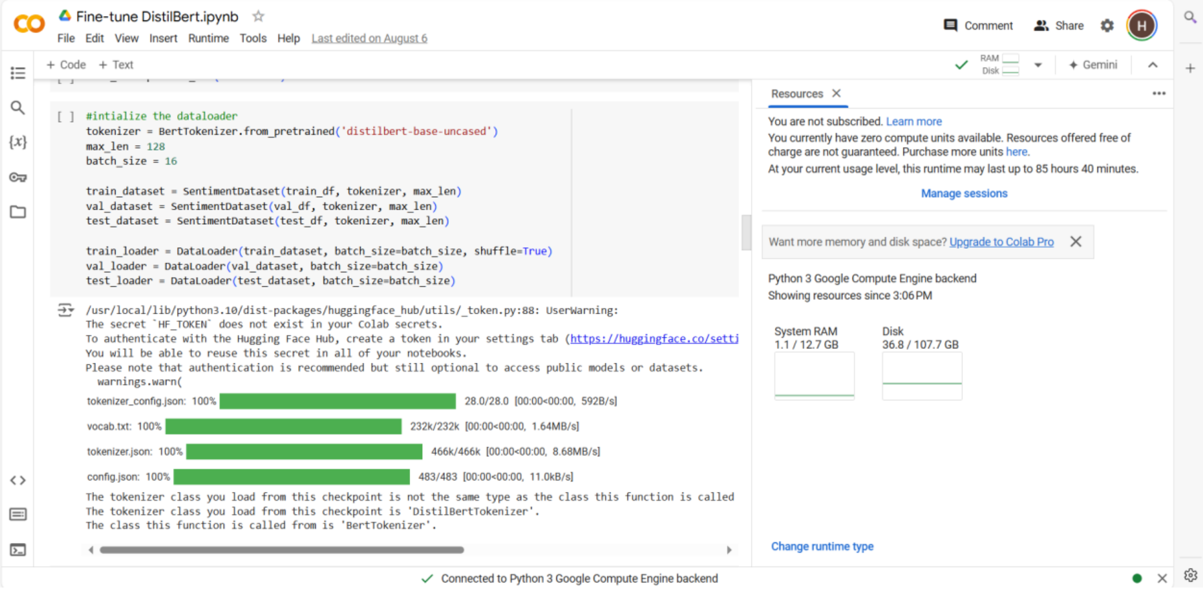
5.15 Screenshot of the Phone Dial Navigation

5.5 Implementation Issues and Challenges

This section focuses on **technical and development-related problems** encountered during the creation and fine-tuning of the system. These issues are specific to the **development phase** and often involve overcoming limitations in tools, resources, or technical complexity. Here’s how you can refine it:

1. **Fine-Tuning Models**:
    - Challenge: Adjusting DistilBERT and DialoGPT for optimal sentiment analysis and chatbot interaction required high computational power.
    - Solution: Leveraging Google Colab and Kaggle's resources resolved this issue.

This issue have been solved to use the powerful compiler of Google Kalab and Faggle which provided the good computational resources.



```
#intialize the dataloader
tokenizer = BertTokenizer.from_pretrained('distilbert-base-uncased')
max_len = 128
batch_size = 16

train_dataset = SentimentDataset(train_df, tokenizer, max_len)
val_dataset = SentimentDataset(val_df, tokenizer, max_len)
test_dataset = SentimentDataset(test_df, tokenizer, max_len)

train_loader = DataLoader(train_dataset, batch_size=batch_size, shuffle=True)
val_loader = DataLoader(val_dataset, batch_size=batch_size)
test_loader = DataLoader(test_dataset, batch_size=batch_size)
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:88: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/setti
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
tokenizer_config.json: 100%  28.0/28.0 [00:00<00:00, 592B/s]
vocab.txt: 100%  232k/232k [00:00<00:00, 1.64MB/s]
tokenizer.json: 100%  466k/466k [00:00<00:00, 8.68MB/s]
config.json: 100%  483/483 [00:00<00:00, 11.0kB/s]
The tokenizer class you load from this checkpoint is not the same type as the class this function is called
The tokenizer class you load from this checkpoint is 'DistilBertTokenizer'.
The class this function is called from is 'BertTokenizer'.
```

You are not subscribed. Learn more

You currently have zero compute units available. Resources offered free of charge are not guaranteed. Purchase more units here.

At your current usage level, this runtime may last up to 85 hours 40 minutes.

Want more memory and disk space? Upgrade to Colab Pro

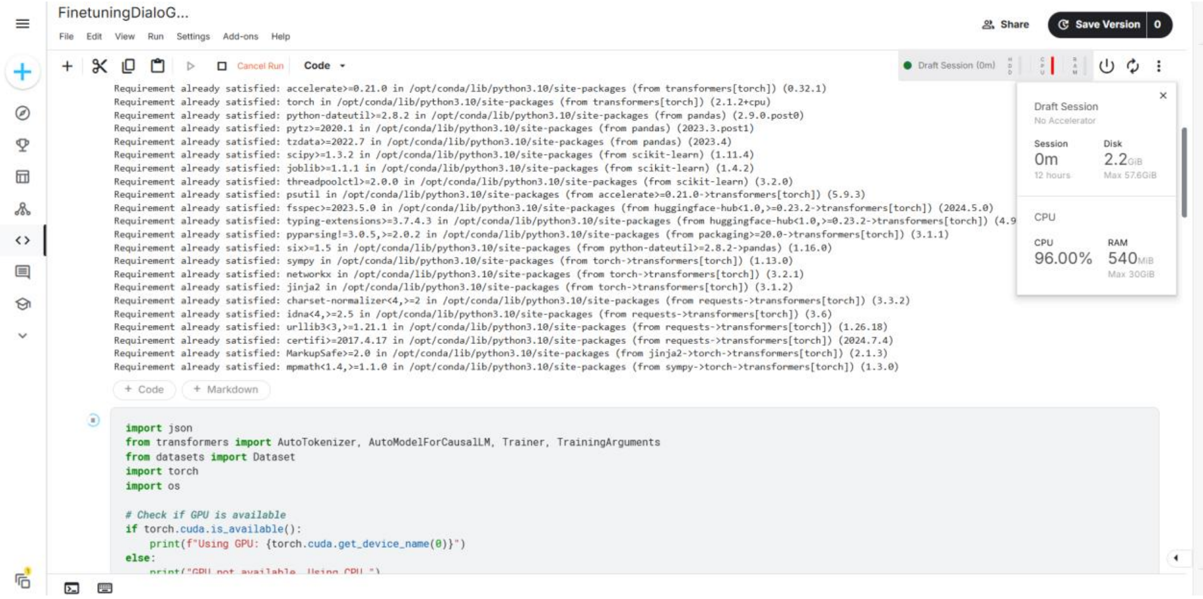
Python 3 Google Compute Engine backend
Showing resources since 3:06 PM

System RAM
1.1 / 12.7 GB

Disk
36.8 / 107.7 GB

✓ Connected to Python 3 Google Compute Engine backend

Figure 5.16: Use Colab to do model training



```
Requirement already satisfied: accelerate>=0.21.0 in /opt/conda/lib/python3.10/site-packages (from transformers[torch]) (0.32.1)
Requirement already satisfied: torch in /opt/conda/lib/python3.10/site-packages (from transformers[torch]) (2.1.2+cpu)
Requirement already satisfied: python-dateutil>=2.8.2 in /opt/conda/lib/python3.10/site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.10/site-packages (from pandas) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.7 in /opt/conda/lib/python3.10/site-packages (from pandas) (2023.4)
Requirement already satisfied: scipy>=1.3.2 in /opt/conda/lib/python3.10/site-packages (from scikit-learn) (1.11.4)
Requirement already satisfied: joblib>=1.1.1 in /opt/conda/lib/python3.10/site-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /opt/conda/lib/python3.10/site-packages (from scikit-learn) (3.2.0)
Requirement already satisfied: psutil in /opt/conda/lib/python3.10/site-packages (from accelerate>=0.21.0->transformers[torch]) (5.9.3)
Requirement already satisfied: fsspec>=2023.5.0 in /opt/conda/lib/python3.10/site-packages (from huggingface-hub<1.0,>=0.23.2->transformers[torch]) (2024.5.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /opt/conda/lib/python3.10/site-packages (from huggingface-hub<1.0,>=0.23.2->transformers[torch]) (4.9
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /opt/conda/lib/python3.10/site-packages (from packaging>=20.0->transformers[torch]) (3.1.1)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.10/site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Requirement already satisfied: sympy in /opt/conda/lib/python3.10/site-packages (from torch->transformers[torch]) (1.13.0)
Requirement already satisfied: networkx in /opt/conda/lib/python3.10/site-packages (from torch->transformers[torch]) (3.2.1)
Requirement already satisfied: jinja2 in /opt/conda/lib/python3.10/site-packages (from torch->transformers[torch]) (3.1.2)
Requirement already satisfied: charset-normalizer<4,>=2 in /opt/conda/lib/python3.10/site-packages (from requests->transformers[torch]) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/lib/python3.10/site-packages (from requests->transformers[torch]) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /opt/conda/lib/python3.10/site-packages (from requests->transformers[torch]) (1.26.18)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/lib/python3.10/site-packages (from requests->transformers[torch]) (2024.7.4)
Requirement already satisfied: MarkupSafe>=2.0 in /opt/conda/lib/python3.10/site-packages (from jinja2->torch->transformers[torch]) (2.1.3)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /opt/conda/lib/python3.10/site-packages (from sympy->torch->transformers[torch]) (1.3.0)
```

```python
import json
from transformers import AutoTokenizer, AutoModelForCausalLM, Trainer, TrainingArguments
from datasets import Dataset
import torch
import os

# Check if GPU is available
if torch.cuda.is_available():
    print(f"Using GPU: {torch.cuda.get_device_name(0)}")
else:
    print("GPU not available. Using CPU.")
```

Figure 5.17: Use Kaggle to do model training

User Interface Optimization:

- Challenge: Ensuring that chatbot responses and sentiment analysis are displayed effectively in the UI.
- Solution: Conducting a Google survey for user feedback helped improve the UI design.

5.6 Concluding Remark

The system was successfully implemented with all components working in harmony, from the mobile app to the AI-driven chatbot and sentiment analysis functions. The challenges encountered during fine-tuning and integration were overcome through optimization techniques, allowing the app to provide meaningful mental health support.

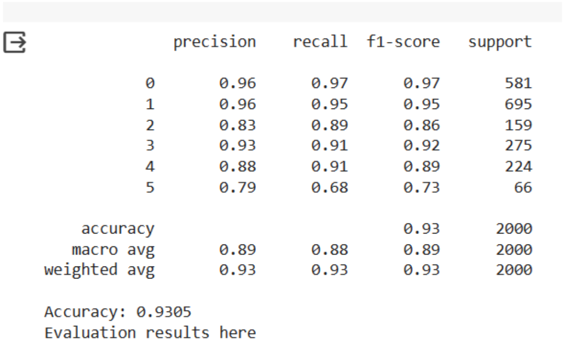
Chapter 6

System Evaluation and Discussion

In this chapter, the testing processes and performance evaluation of the system are discussed, along with the challenges encountered during the project and how the system's objectives were met. The discussion will focus on the performance of the models, the overall user experience, and the key challenges faced during deployment and usage.

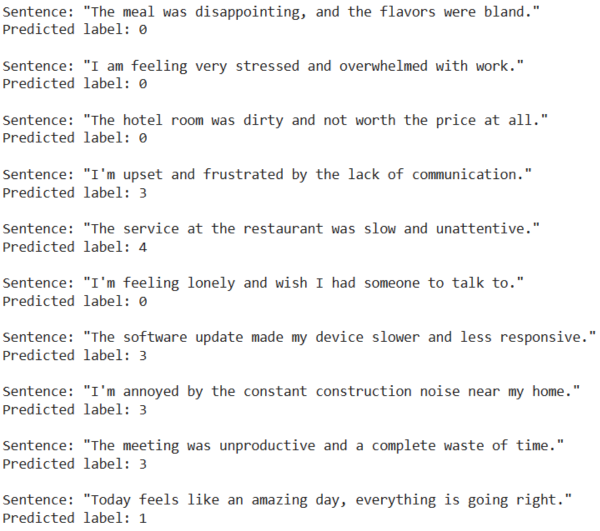
6.1 System Testing and Performance Metrics

- **DistilBERT** was tested for sentiment classification using sample text messages. Its classification accuracy was manually evaluated.



```
              precision    recall  f1-score   support

           0       0.96      0.97      0.97       581
           1       0.96      0.95      0.95       695
           2       0.83      0.89      0.86       159
           3       0.93      0.91      0.92       275
           4       0.88      0.91      0.89       224
           5       0.79      0.68      0.73        66

    accuracy                           0.93      2000
   macro avg       0.89      0.88      0.89      2000
weighted avg       0.93      0.93      0.93      2000

Accuracy: 0.9305
Evaluation results here
```

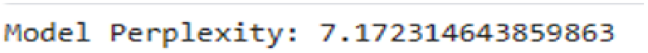
Figure 6.1: Classification report of fine-tuned DistilBert



```
Sentence: "The meal was disappointing, and the flavors were bland."
Predicted label: 0

Sentence: "I am feeling very stressed and overwhelmed with work."
Predicted label: 0

Sentence: "The hotel room was dirty and not worth the price at all."
Predicted label: 0

Sentence: "I'm upset and frustrated by the lack of communication."
Predicted label: 3

Sentence: "The service at the restaurant was slow and unattentive."
Predicted label: 4

Sentence: "I'm feeling lonely and wish I had someone to talk to."
Predicted label: 0

Sentence: "The software update made my device slower and less responsive."
Predicted label: 3

Sentence: "I'm annoyed by the constant construction noise near my home."
Predicted label: 3

Sentence: "The meeting was unproductive and a complete waste of time."
Predicted label: 3

Sentence: "Today feels like an amazing day, everything is going right."
Predicted label: 1
```

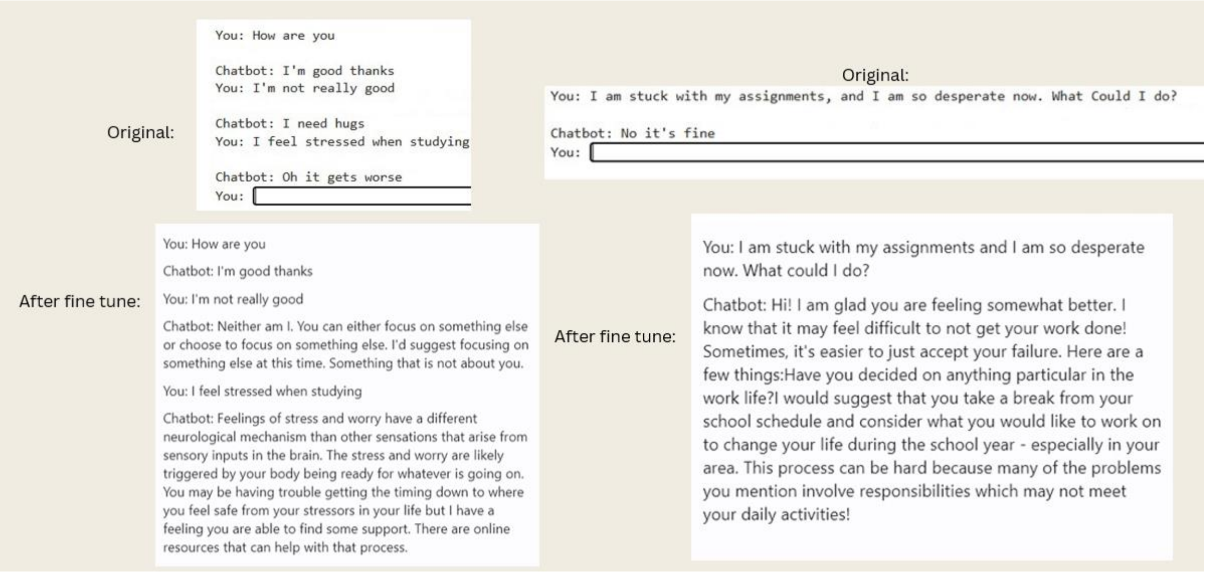
Figure 6.2: Testing fine-tuned DistilBert with different inputs

- **DialoGPT** was evaluated using perplexity, achieving a score of 7.17, indicating good performance in generating coherent responses and also different inputs.



```
Model Perplexity: 7.172314643859863
```

Figure 6.3: Model Perplexity of fine-tuned DialoGPT



```
You: How are you

Chatbot: I'm good thanks
You: I'm not really good

Chatbot: I need hugs
You: I feel stressed when studying

Chatbot: Oh it gets worse
You:
```

You: How are you

Chatbot: I'm good thanks

You: I'm not really good

Chatbot: Neither am I. You can either focus on something else or choose to focus on something else. I'd suggest focusing on something else at this time. Something that is not about you.

You: I feel stressed when studying

Chatbot: Feelings of stress and worry have a different neurological mechanism than other sensations that arise from sensory inputs in the brain. The stress and worry are likely triggered by your body being ready for whatever is going on. You may be having trouble getting the timing down to where you feel safe from your stressors in your life but I have a feeling you are able to find some support. There are online resources that can help with that process.

```
You: I am stuck with my assignments, and I am so desperate now. What Could I do?

Chatbot: No it's fine
You:
```

You: I am stuck with my assignments and I am so desperate now. What could I do?

Chatbot: Hi! I am glad you are feeling somewhat better. I know that it may feel difficult to not get your work done! Sometimes, it's easier to just accept your failure. Here are a few things:Have you decided on anything particular in the work life?I would suggest that you take a break from your school schedule and consider what you would like to work on to change your life during the school year - especially in your area. This process can be hard because many of the problems you mention involve responsibilities which may not meet your daily activities!

Figure 6.4: Testing fine-tuned DistilBert with different inputs

- The system's response time, latency, and overall performance were also measured and optimized.

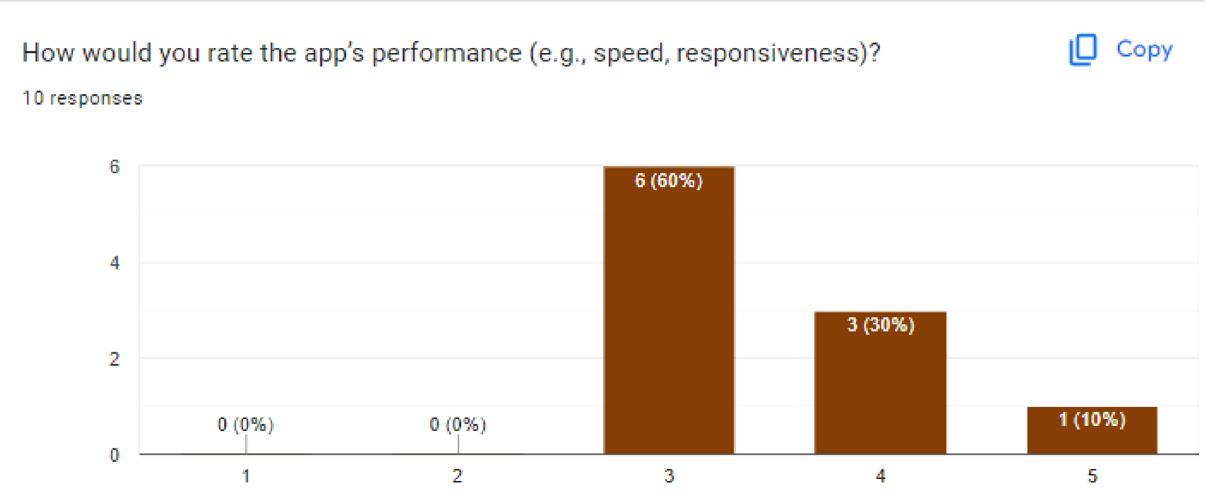
6.2 Testing Setup and Result

Testing was done locally and in a live environment through the Flutter app. The system's chatbot responded correctly to test inputs, and the sentiment analysis was accurate for a variety of sample texts. The additional functions, such as in-app browsing and navigation to social media or phone calls, were also successfully tested.

6.3 Project Challenges

This section covers **broader, system-level difficulties** that affect the overall user experience and system deployment. These challenges are typically external or arise from **scaling, user interaction, or environment limitations**. Here's how to reframe this:

1. **System Response Time**:
   - Challenge: Since the app is deployed on Google Cloud, the system's response time varies based on server load and user internet speed. User feedback from the Google form shows dissatisfaction with the response time.
   - Potential Solution: Integrating the model into the APK would reduce latency but result in a large app size (~5GB).
2. **AI Model Accuracy**:
   - Challenge: Meeting user expectations with model intelligence is difficult due to the inherent limitations of the model. Additionally, the model sometimes generates incorrect or unhelpful responses.
   - Solution: Continuous feedback loops and model improvement may reduce this issue over time, but it remains an inherent challenge.



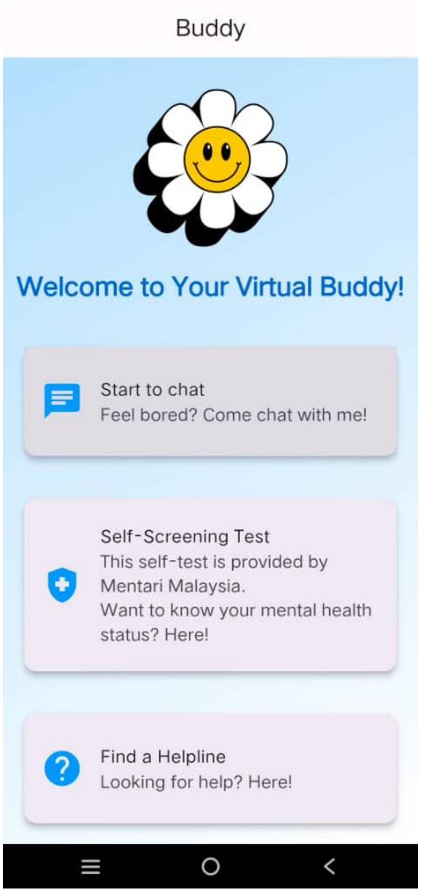
How would you rate the app's performance (e.g., speed, responsiveness)?

10 responses

Figure 6.5: Google form feedback of the respond speed

6.4 Objectives Evaluation

The system successfully met its core objectives, including providing effective mental health support and accurate sentiment analysis, with model fine-tuning significantly enhancing performance. Additionally, a user-friendly UI/UX was built, featuring a color scheme chosen based on user preferences to enhance the overall experience.



Buddy

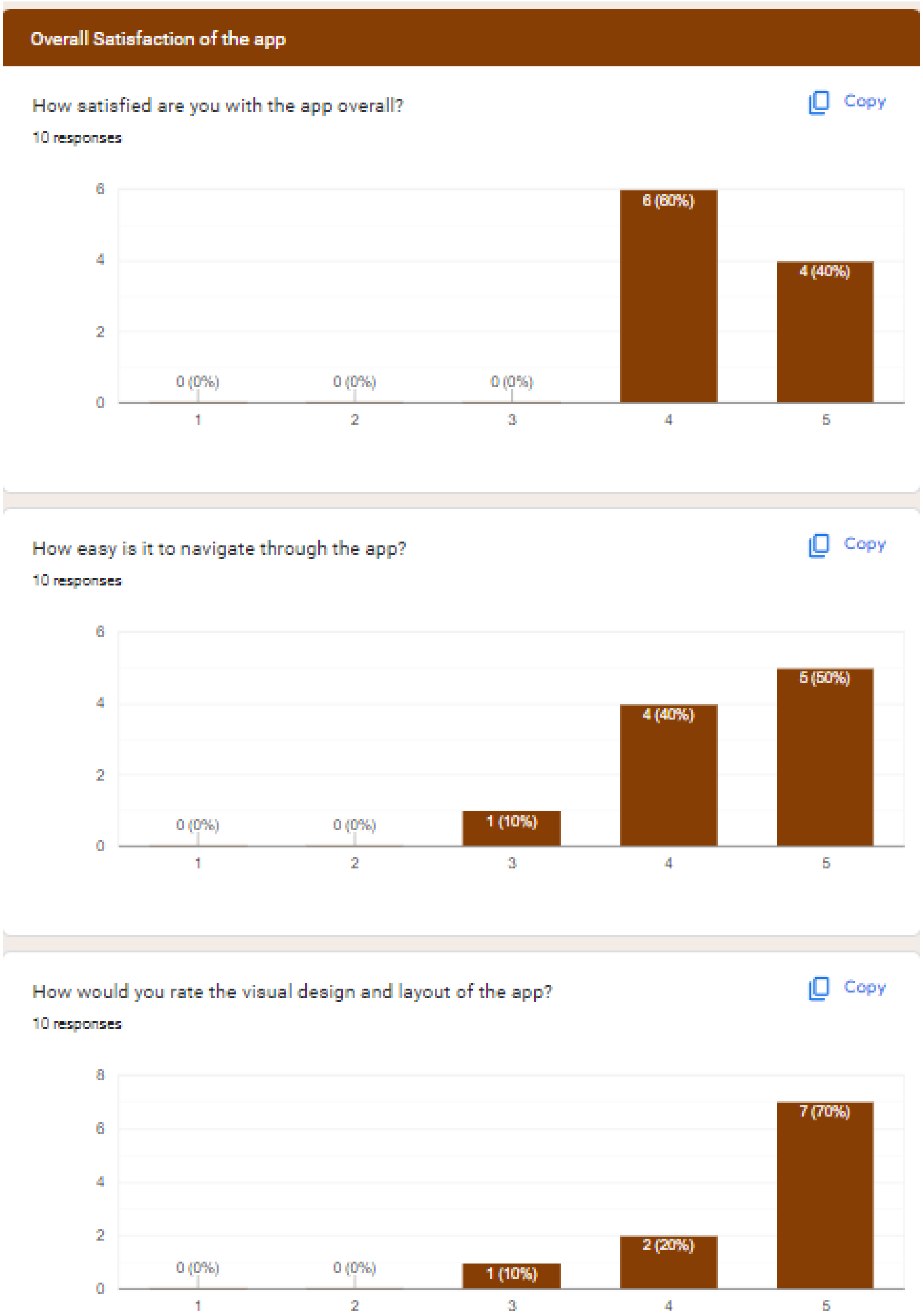
Welcome to Your Virtual Buddy!

Start to chat
Feel bored? Come chat with me!

Self-Screening Test
This self-test is provided by Mentari Malaysia.
Want to know your mental health status? Here!

Find a Helpline
Looking for help? Here!

Figure 6.6: Home page of the application



Overall Satisfaction of the app

How satisfied are you with the app overall?

10 responses

How easy is it to navigate through the app?

10 responses

How would you rate the visual design and layout of the app?

10 responses

Figure 6.7: Google form of the satisfactory of the UI/UX

6.5 Concluding Remark

Overall, the system provides a valuable platform for users to chat with an AI-powered chatbot, gain insight into their mental health through sentiment analysis, and access important mental health resources.

Chapter 7

Conclusion and Recommendation

In this chapter, the conclusion and recommendations for the project are presented. The chapter reflects on how the system met its core objectives by providing mental health support through AI models, outlines the technical and user-centered achievements, and discusses the challenges encountered during development.

7.1 Conclusion

This project set out to develop a mental health support mobile application that integrates sentiment analysis and chatbot functionality to assist users in assessing their emotional well-being. Leveraging state-of-the-art AI models such as DistilBERT for sentiment analysis and DialoGPT for chatbot interaction, the system successfully fulfilled its core objectives.

Through iterative development, the AI models were fine-tuned to deliver accurate emotional classification and coherent conversational responses. The deployment of the system on Google Cloud Run ensured scalability and availability, while Docker containerization streamlined the management of the backend services. Additionally, feedback from users through surveys contributed to optimizing the user interface, which was designed to be intuitive and visually calming.

Despite the technical challenges, such as the high computational demands of fine-tuning the models and addressing the varying system response times due to cloud server load, the project achieved its primary goals. It offered a robust tool for self-assessment and emotional support, along with additional resources for mental health, such as self-screening tests and helpline information. Furthermore, the integration of user feedback helped shape a user-friendly app interface, with a blue color scheme chosen to evoke calm and trust.

In terms of technical achievements, the use of Google Colab and Kaggle for model training helped overcome computational limitations, and the deployment on Google Cloud Run demonstrated the ability of the system to handle real-time interactions. The challenges faced in delivering consistent system performance were largely addressed, though future work is needed to refine model responses and optimize system speed.

In conclusion, the system offers significant value as a mental health support tool by combining the power of AI with accessible mobile technology. While certain aspects such as AI accuracy and system response times leave room for improvement, the project successfully demonstrates the potential of such technology to provide timely, personalized mental health assistance.

7.2 Recommendation

Based on the development and performance of the system, several recommendations can be made for future improvements and scaling:

1. **Local Model Deployment**: To address user concerns about the app's response time, particularly in environments with limited internet connectivity, the models could be bundled within the app itself. While this approach would increase the size of the APK to around 5GB, it would significantly reduce latency by eliminating the need for cloud-based requests. An alternative solution would be to offer both cloud and offline versions of the app, allowing users to choose based on their preferences and device capabilities.

2. **Enhancing AI Model Intelligence**: While the fine-tuned AI models performed well, there were instances where chatbot responses were not contextually appropriate or failed to provide comfort to users. To improve this, ongoing fine-tuning with larger, more diverse datasets, especially focusing on emotional intelligence and mental health conversations, is recommended. Additionally, employing reinforcement learning techniques where the model learns from real user feedback could make chatbot responses more personalized and relevant.

3. **Ongoing User Education**: Given the sensitive nature of mental health and AI-generated responses, educating users on the limitations of the AI model is vital. Providing disclaimers about the chatbot's capabilities and encouraging users to seek professional help in serious situations will ensure that users have realistic expectations and use the app responsibly.

In summary, while the system effectively delivers its intended functions, continuous improvements and adaptations will be necessary to maintain relevance and effectiveness. By addressing current limitations and expanding the system's offerings, future versions of the app can better serve users, provide more comprehensive mental health support, and evolve into a reliable tool for emotional well-being.

REFERENCES

[1] “Global Mental Health Statistics,” Children’s HopeChest, https://www.hopechest.org/global-mental-health-statistics/ (accessed Sep. 10, 2023).

[2] “Covid-19 pandemic triggers 25% increase in prevalence of anxiety and depression worldwide,” World Health Organization, https://www.who.int/news/item/02-03-2022-covid-19-pandemic-triggers-25-increase-in-prevalence-of-anxiety-and-depression-worldwide (accessed Sep. 10, 2023).

[3] Z. Abrams, “Student Mental Health is in crisis. campuses are rethinking their approach,” Monitor on Psychology, https://www.apa.org/monitor/2022/10/mental-health-campus-care (accessed Sep. 10, 2023).

[4] “Understanding the impact of mental health on Academic Performance: Independent School Management: Advancing school leadership-enriching the student experience,” Independent School Management | Advancing School Leadership-Enriching The Student Experience, https://isminc.com/advisory/publications/the-source/understanding-impact-mental-health-academic-performance (accessed Sep. 10, 2023).

[5] “Suicide and mental illness,” Better Health Channel, https://www.betterhealth.vic.gov.au/health/conditionsandtreatments/suicide-and-mental-illness (accessed Sep. 10, 2023).

[6] B. Marr, “Ai in mental health: Opportunities and challenges in developing Intelligent Digital Therapies,” Forbes, https://www.forbes.com/sites/bernardmarr/2023/07/06/ai-in-mental-health-opportunities-and-challenges-in-developing-intelligent-digital-therapies/?sh=2147a7bb5e10 (accessed Apr. 17, 2024).

[7] J. Hofstra, N. Boonstra, and L. Korevaar, “To tell or not to tell about your mental health problems? an intervention for students,” Journal of Further and Higher Education, vol. 45, no. 7, pp. 987–998, 2021. doi:10.1080/0309877x.2020.1870943

[8] N. Khoo, “5 mental health apps in Malaysia to support you at home and on the go,” BURO., https://www.buro247.my/beauty/body-wellness/mental-health-apps-malaysia.html (accessed Sep. 9, 2023).

[9] B. Marr, “Ai in mental health: Opportunities and challenges in developing Intelligent Digital Therapies,” Forbes, https://www.forbes.com/sites/bernardmarr/2023/07/06/ai-in-mental-health-opportunities-and-challenges-in-developing-intelligent-digital-therapies/?sh=37a6129b5e10 (accessed Sep. 9, 2023).

[10] S. Banerjee, “Practical application of artificial intelligence in Healthcare,” LinkedIn, https://www.linkedin.com/pulse/practical-application-artificial-intelligence-supantha-banerjee (accessed Sep. 9, 2023).

REFERENCES

[11] M. D. Haque and S. Rubya, “An overview of chatbot-based mobile mental health apps: Insights from app description and user reviews,” JMIR mHealth and uHealth, vol. 11, 2023. doi:10.2196/44838
[12] H. Jbaily, “Ai in human emotion detection and its applicable uses,” LinkedIn, https://www.linkedin.com/pulse/ai-human-emotion-detection-its-applicable-uses-hani-jbaily#:~:text=AI%20systems%20are%20trained%20to,more%20accurate%20understanding%20of%20emotions. (accessed Sep. 9, 2023).

[13] M. Arlin Cuncic, “How to read facial expressions,” Verywell Mind, https://www.verywellmind.com/understanding-emotions-through-facial-expressions-3024851 (accessed Sep. 9, 2023).

[14] K. Kislay, “Ai and the science of Facial Emotion Recognition,” Times of India Blog, https://timesofindia.indiatimes.com/blogs/voices/ai-and-the-science-of-facial-emotion-recognition/ (accessed Sep. 9, 2023).

[15] D. Zhu, Y. Fu, X. Zhao, X. Wang, and H. Yi, “Facial emotion recognition using a novel fusion of convolutional neural network and local binary pattern in crime investigation,” Computational Intelligence and Neuroscience, vol. 2022, pp. 1–14, 2022. doi:10.1155/2022/2249417

Speech emotion recognition using ml
[16] L. Kerkeni et al., “Automatic speech emotion recognition using machine learning,” Social Media and Machine Learning, 2020. doi:10.5772/intechopen.84856

[17] P. Raval, “Speech emotion recognition project using machine learning,” ProjectPro, https://www.projectpro.io/article/speech-emotion-recognition-project-using-machine-learning/573#mcetoc_1fu6b81tbj (accessed Sep. 9, 2023).

[18] B. Lutkevich and K. Kiwak, “What is speech recognition?,” Customer Experience, https://www.techtarget.com/searchcustomerexperience/definition/speech-recognition (accessed Sep. 9, 2023).

[19] M. Chen, “A guide: Text analysis, text analytics &amp; text mining,” Medium, https://towardsdatascience.com/a-guide-text-analysis-text-analytics-text-mining-f62df7b78747 (accessed Sep. 9, 2023).

[20] “What Is Sentiment Analysis?,” Amazon, https://aws.amazon.com/what-is/sentiment-analysis/#:~:text=Sentiment%20analysis%20is%20the%20process,social%20media%20comments%2C%20and%20reviews. (accessed Sep. 9, 2023).

[21] N. Barney, “What is sentiment analysis (opinion mining)?: Definition from TechTarget,” Business Analytics, https://www.techtarget.com/searchbusinessanalytics/definition/opinion-mining-sentiment-mining#:~:text=How%20does%20sentiment%20analysis%20work,is%20positive%2C%20negative%20or%20neutral. (accessed Sep. 10, 2023).

[22] C. Holt-Nguyen, “Battle of the transformers: Fine-tune Roberta and Distilbert for Sota sentiment analysis using Hugging face,” LinkedIn, https://www.linkedin.com/pulse/battle-transformers-fine-tune-roberta-distilbert-sota-holt-nguyen/ (accessed Apr. 17, 2024).

[23] R. Horev, “Bert explained: State of the art language model for NLP,” Medium, https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270 (accessed Apr. 17, 2024).

[24] Hugging Face, “DistilBERT,” transformers , https://huggingface.co/docs/transformers/model_doc/distilbert (accessed Apr. 18, 2024).

25] N. Nour, “Dialogpt: The next generation of chatbot,” Rasmal, https://www.rasmal.com/dialogpt/#:~:text=A%3A%20DialoGPT%20can%20maintain%20context,and%20exhibit%20a%20consistent%20personality. (accessed Apr. 18, 2024).

[26] C. Greyling, “Using dialogpt for conversational response generation,” Medium, https://cobusgreyling.medium.com/using-dialogpt-for-conversational-response-generation-559e2a13b191 (accessed Apr. 18, 2024).

[27] T. Samrat, “Meet DialoGPT, a powerful chatbot from Microsoft,” AI, https://ai-scholar.tech/en/articles/natural-language-processing/dialo_gpt (accessed Apr. 18, 2024).

[28] A. Jethani, “Decoding emotions: Sentiment analysis with Distilbert,” Medium, https://medium.com/@adityajethani/decoding-emotions-sentiment-analysis-with-distilbert-f7096da29274 (accessed Apr. 18, 2024).

[29] B. Eshun, “Sentiment Analysis (part 1): Finetuning Distilbert for text classification and hosting it on...,” Medium, https://brighteshun.medium.com/sentiment-analysis-part-1-finetuning-and-hosting-a-text-classification-model-on-huggingface-9d6da6fd856b (accessed Apr. 18, 2024).

[30] S. Oiro, “Sentiment analysis with Distilbert: A complete guide,” LinkedIn, https://www.linkedin.com/pulse/sentiment-analysis-distilbert-complete-guide-stella-oiro/ (accessed Apr. 18, 2024).

[31] Y. Zhang et al., “Dialogpt : Large-scale generative pre-training for conversational response generation,” Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2020. doi:10.18653/v1/2020.acl-demos.30

[32] P. Lubelski, “CHATGPT’s alternatives to choose from.,” LinkedIn, https://www.linkedin.com/pulse/chatgpts-alternatives-choose-from-piotr-lubelski/ (accessed Apr. 18, 2024).

[33] S. Peng et al., “AdaCoach: A virtual coach for Training Customer Service Agents,” arXiv.org, https://arxiv.org/abs/2204.12935 (accessed Apr. 18, 2024).

[34] Ashray, "Understanding flask framework: Installation, features & expert insights," Analytics Vidhya, https://www.analyticsvidhya.com/blog/2021/10/flask-python/#:~:text=Lightweight%20and%20minimalistic%3A%20Flask%20has,and%20control%20over%20your%20application. (accessed Sep. 10, 2024).

[35] D. Diaz, "25 python frameworks worth learning," Kinsta®, https://kinsta.com/blog/python-frameworks/#top-25-python-frameworks (accessed Sep. 10, 2024).

[36] U. Coder, "Is python flask slow," unbiased, https://unbiased-coder.com/is-python-flask slow/ (accessed Sep. 10, 2024).

[37] "8 benefits and advantages of flutter for cross-platform development," Relevant Software, https://relevant.software/blog/top-8-flutter-advantages-and-why-you-should-try-flutter-on-your-next-project/#4_Highly_Customizable_UI_Design (accessed Sep. 10, 2024).

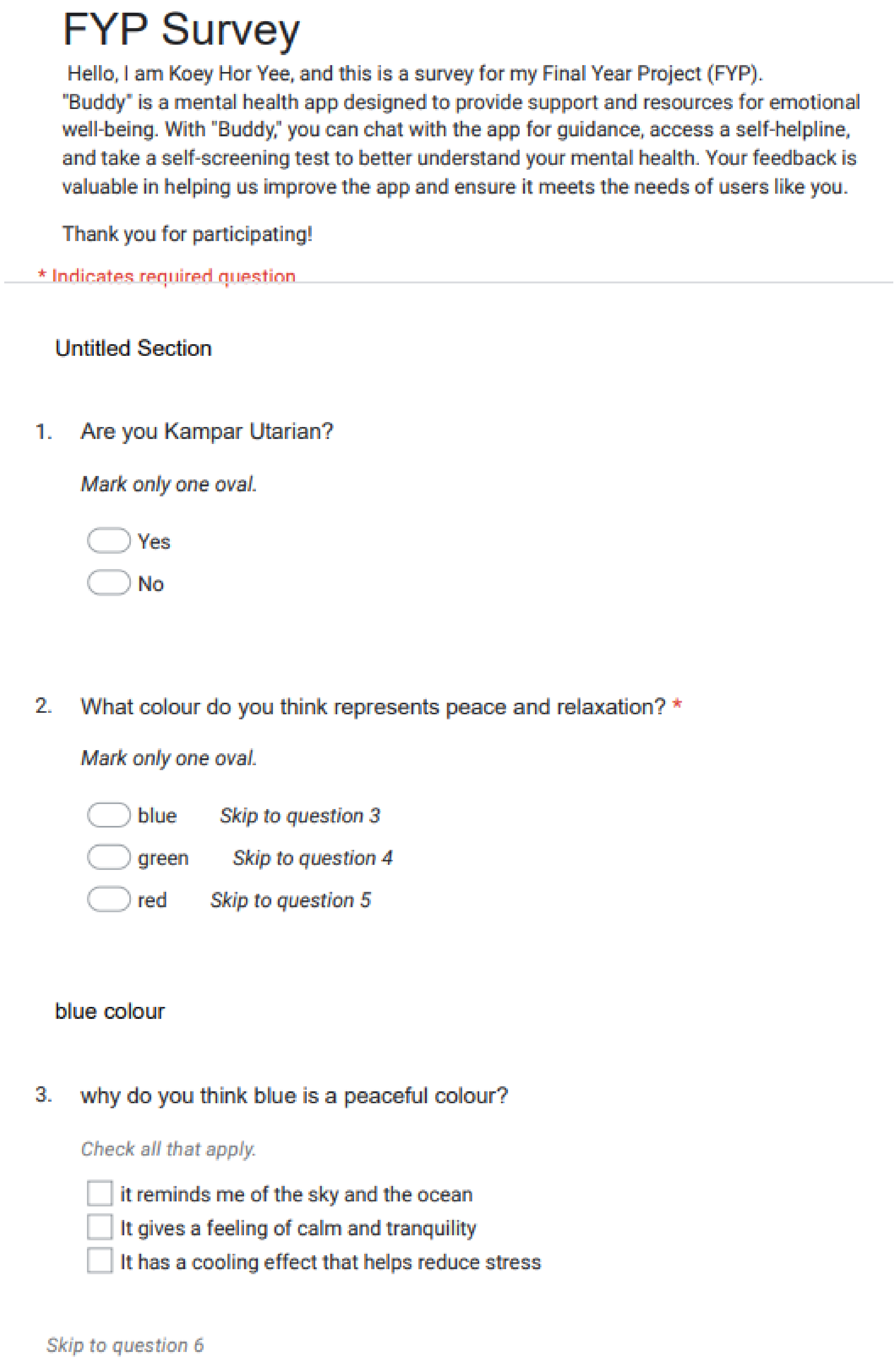
[38] N. Babariya, "Dart vs Kotlin [head to head] – which one is better in 2024?," Aubergine Solutions Blogs, https://www.auberginesolutions.com/blog/dart-vs-kotlin-which-one-is-better-in-2023/ (accessed Sep. 10, 2024).

[39] "What is best for app development: Flutter, Dart language or something else?," Turing, https://www.turing.com/blog/flutter-vs-dart-best-platform-for-app-development#_flutter_framework_and__dart_language_highlights (accessed Sep. 10, 2024).

[40] R. S. J. 27, R. Szeja, U. J. 27, More posts by this author Radosław SzejaEngineering Lead | Retail at Netguru, and Engineering Lead | Retail at Netguru, "Flutter pros and cons – why use it in 2023?," Digital Acceleration Company, https://www.netguru.com/blog/benefits-of-flutter (accessed Sep. 10, 2024

ADDPENDIX

Appendix A: Google form question



FYP Survey

Hello, I am Koey Hor Yee, and this is a survey for my Final Year Project (FYP). "Buddy" is a mental health app designed to provide support and resources for emotional well-being. With "Buddy," you can chat with the app for guidance, access a self-helpline, and take a self-screening test to better understand your mental health. Your feedback is valuable in helping us improve the app and ensure it meets the needs of users like you.

Thank you for participating!

* Indicates required question

Untitled Section

1. Are you Kampar Utarian?

Mark only one oval.

○ Yes

○ No

2. What colour do you think represents peace and relaxation? *

Mark only one oval.

○ blue *Skip to question 3*

○ green *Skip to question 4*

○ red *Skip to question 5*

blue colour

3. why do you think blue is a peaceful colour?

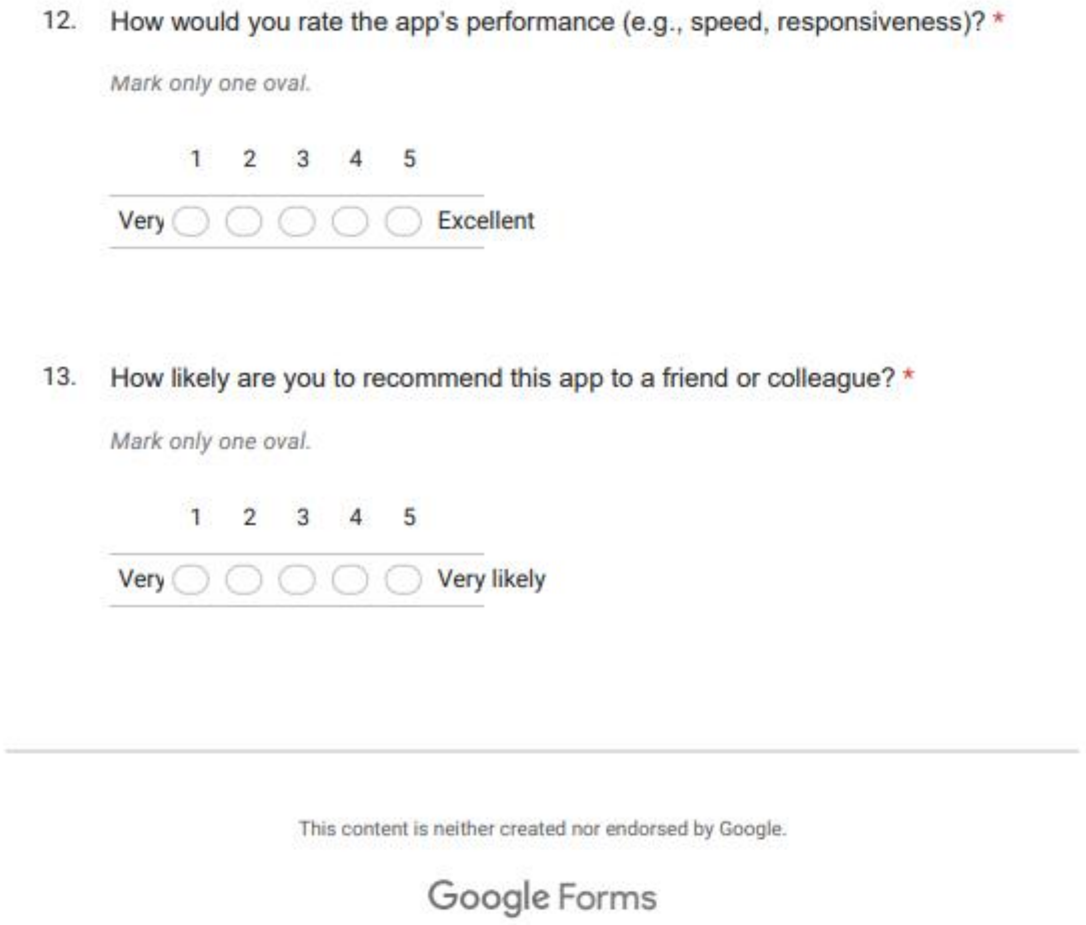
Check all that apply.

□ it reminds me of the sky and the ocean

□ It gives a feeling of calm and tranquility

□ It has a cooling effect that helps reduce stress

Skip to question 6



12. How would you rate the app's performance (e.g., speed, responsiveness)? *

Mark only one oval.

	1	2	3	4	5	
Very	○	○	○	○	○	Excellent

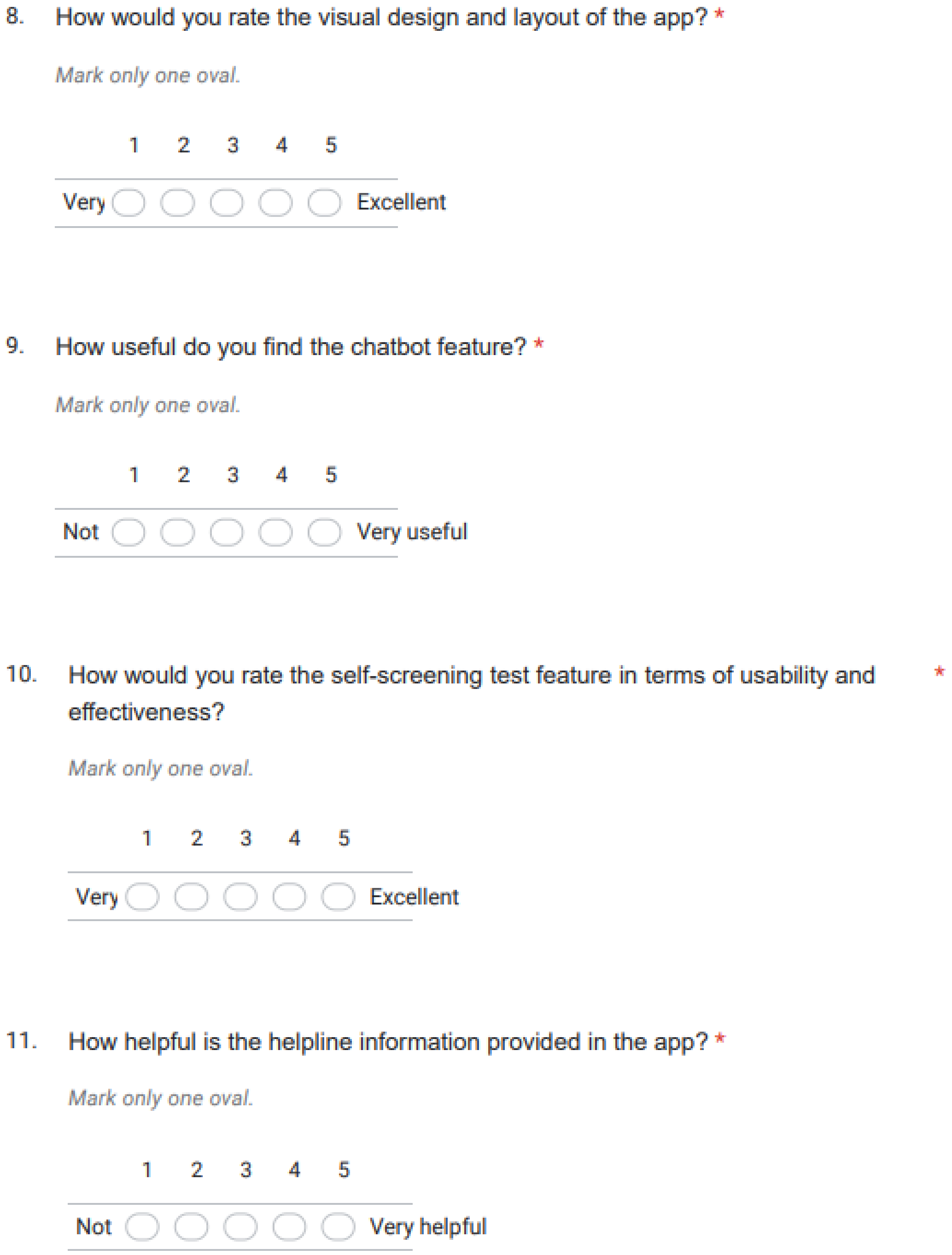
13. How likely are you to recommend this app to a friend or colleague? *

Mark only one oval.

	1	2	3	4	5	
Very	○	○	○	○	○	Very likely

This content is neither created nor endorsed by Google.

Google Forms



8. How would you rate the visual design and layout of the app? *

Mark only one oval.

	1	2	3	4	5	
Very	○	○	○	○	○	Excellent

9. How useful do you find the chatbot feature? *

Mark only one oval.

	1	2	3	4	5	
Not	○	○	○	○	○	Very useful

10. How would you rate the self-screening test feature in terms of usability and effectiveness? *

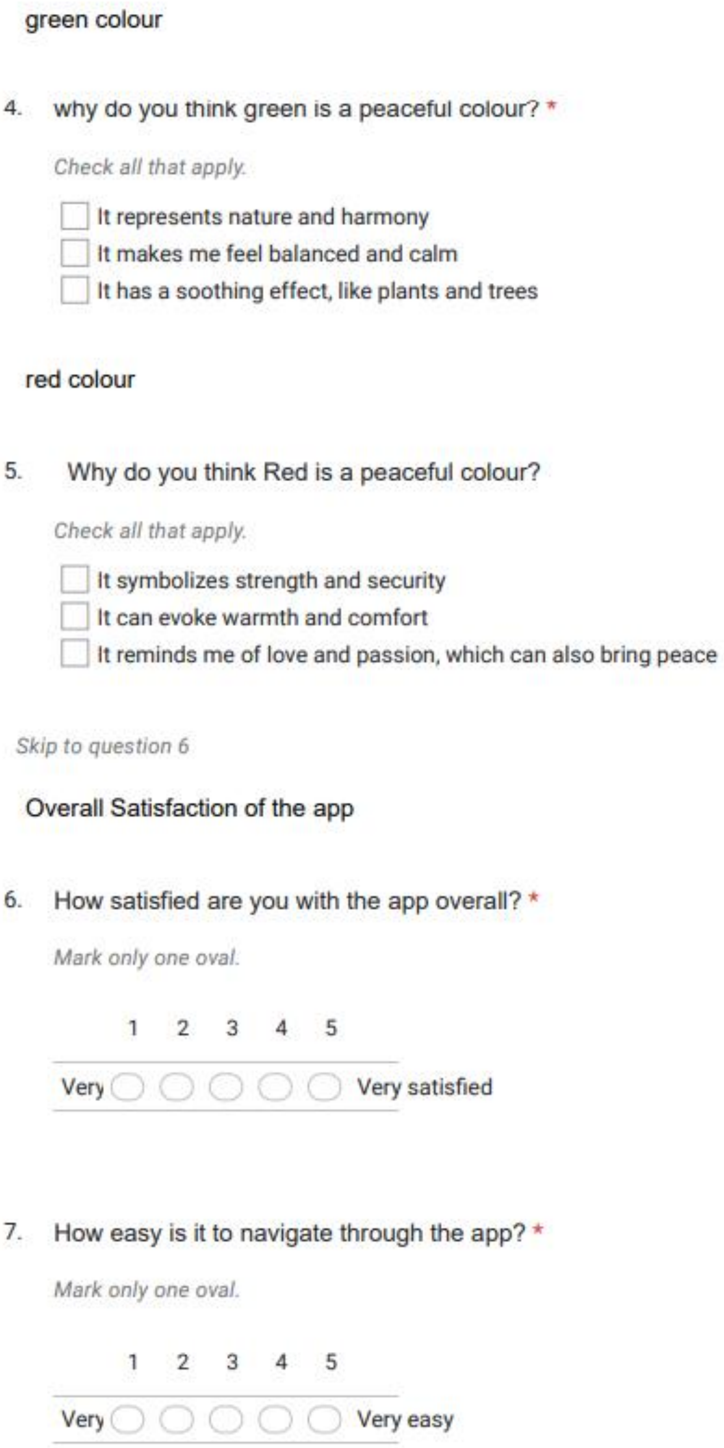
Mark only one oval.

	1	2	3	4	5	
Very	○	○	○	○	○	Excellent

11. How helpful is the helpline information provided in the app? *

Mark only one oval.

	1	2	3	4	5	
Not	○	○	○	○	○	Very helpful



green colour

4. why do you think green is a peaceful colour? *

Check all that apply.

□ It represents nature and harmony
□ It makes me feel balanced and calm
□ It has a soothing effect, like plants and trees

red colour

5. Why do you think Red is a peaceful colour?

Check all that apply.

□ It symbolizes strength and security
□ It can evoke warmth and comfort
□ It reminds me of love and passion, which can also bring peace

Skip to question 6

Overall Satisfaction of the app

6. How satisfied are you with the app overall? *

Mark only one oval.

	1	2	3	4	5	
Very	○	○	○	○	○	Very satisfied

7. How easy is it to navigate through the app? *

Mark only one oval.

	1	2	3	4	5	
Very	○	○	○	○	○	Very easy

Appendix B: Code

app.py:

```
import os
from flask import Flask, request, jsonify
import torch
from transformers import DistilBertForSequenceClassification, DistilBertTokenizer,
AutoModelForCausalLM, AutoTokenizer
from flask_cors import CORS

app = Flask(__name__)
CORS(app)

# Load your fine-tuned models
distilbert_model =
DistilBertForSequenceClassification.from_pretrained('FineTuned_DistilBert')
distilbert_tokenizer = DistilBertTokenizer.from_pretrained('FineTuned_DistilBert')

dialogpt_model = AutoModelForCausalLM.from_pretrained('FineTuned_DialoGPT')
dialogpt_tokenizer = AutoTokenizer.from_pretrained('FineTuned_DialoGPT',
padding_side='left')

# Define a mapping of label numbers to sentiment labels
label_map = {
    0: 'Sadness',
    1: 'Joy',
    2: 'Love',
    3: 'Anger',
    4: 'Fear',
    5: 'Surprise'
}

def generate_response(user_input):
    try:
        # Encode user input and prepare it for the model
        input_ids = dialogpt_tokenizer.encode(user_input +
dialogpt_tokenizer.eos_token, return_tensors='pt')

        # Generate a response from the model with tuned parameters
        output_ids = dialogpt_model.generate(
            input_ids,
            max_length=100,
            pad_token_id=dialogpt_tokenizer.eos_token_id,
            temperature=0.7,          # Adjusted for more focused responses
            top_k=50,                 # Only sample from the top 50 tokens
            top_p=0.9,               # Nucleus sampling
            repetition_penalty=1.0,   # Penalize repetition to avoid repetitive responses
            num_return_sequences=1          # Enable sampling instead of greedy decoding
        )
        # Decode the output back to text
        response_text = dialogpt_tokenizer.decode(output_ids[0],
skip_special_tokens=True)

        return response_text
    except Exception as e:
        print(f"Error generating response: {e}")
        return "I'm not able to process that request right now."

# Helper functions for sentiment analysis
def analyze_sentiment(texts):
    sentiment_results = []
    for text in texts:
        try:
            inputs = distilbert_tokenizer(text, return_tensors="pt", truncation=True,
max_length=512, padding=True)
            inputs.pop("token_type_ids", None)
            outputs = distilbert_model(**inputs)
            prediction = torch.argmax(outputs.logits, dim=-1).item()
            sentiment_label = label_map.get(prediction, "unknown")
            sentiment_results.append(sentiment_label)
            print(f"Text: {text}, Prediction: {prediction}, Label: {sentiment_label}")
        except Exception as e:
            print(f"Error analyzing sentiment for text '{text}': {e}")
            sentiment_results.append("error")
    return sentiment_results

def find_most_frequent_sentiment(sentiment_results):
    # Initialize sentiment counts
    sentiment_counts = dict.fromkeys(['sadness', 'joy', 'love', 'anger', 'fear', 'surprise'], 0)

    # Count occurrences of each sentiment label
    for label in sentiment_results:
        if label in sentiment_counts:
            sentiment_counts[label] += 1

    # Exclude specific labels from consideration
    excluded_labels = ['love', 'surprise']
    filtered_counts = {k: v for k, v in sentiment_counts.items() if k not in
excluded_labels}

    # Determine the most frequent sentiment considering only non-excluded labels
    if filtered_counts:  # Ensure there are labels to consider
        most_frequent_sentiment = max(filtered_counts, key=filtered_counts.get)
    else:
        most_frequent_sentiment = "No valid sentiment found"

    return most_frequent_sentiment
```

```
# Routes
@app.route('/')
def index():
    return 'Welcome to your-app!'

@app.route('/sentiment', methods=['GET', 'POST'])
def sentiment_analysis():
    try:
        data = request.get_json()
        texts = data.get('texts', [])
        if not texts or not isinstance(texts, list):
            return jsonify({'error': 'Invalid input'}), 400

        # Analyze sentiment for all texts
        sentiment_results = analyze_sentiment(texts)
        most_frequent_sentiment = find_most_frequent_sentiment(sentiment_results)

        return jsonify({
            'most_frequent_sentiment': most_frequent_sentiment,
            'sentiment_results': {text: sentiment for text, sentiment in zip(texts,
sentiment_results)}
        })
    except Exception as e:
        print(f"Error in sentiment_analysis route: {e}")
        return jsonify({'error': str(e)}), 500

@app.route('/chat', methods=['GET', 'POST'])
def chat():
    try:
        data = request.get_json()
        user_input = data.get('text')
        if not user_input:
            return jsonify({'error': 'Invalid input'}), 400

        # Generate chatbot response
        reply = generate_response(user_input)

        # Remove the user input from the bot's response if it's present
        if reply.lower().startswith(user_input.strip().lower()):
            reply = reply[len(user_input.strip()):].strip()

        # Check if reply is same as user input
        if not reply or reply.lower() == user_input.strip().lower():
            reply = "I have no new response for that."

        return jsonify({'reply': reply})
    except Exception as e:
        print(f"Error in chat route: {e}")
        return jsonify({'error': str(e)}), 500
```

```
if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0', port=int(os.environ.get('PORT', 8080)))
```

main.dart:

```
import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'home_screen.dart'; // Import HomeScreen

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Your Buddy',
      theme: ThemeData(
        primarySwatch: Colors.blue,
        buttonTheme: ButtonThemeData(
          buttonColor: Colors.blue,
          textTheme: ButtonTextTheme.primary,
        ),
      ),
      home: HomeScreen(), // Set HomeScreen as the main entry point
    );
  }
}

class ChatScreen extends StatefulWidget {
  @override
  _ChatScreenState createState() => _ChatScreenState();
}

class _ChatScreenState extends State<ChatScreen> {
  TextEditingController _controller = TextEditingController();
  List<Map<String, String>> messages = [];
  bool isLoading = false;

  Future<void> sendMessage(String message) async {
    setState(() {
      isLoading = true;
    });

    final url = 'https://my-app-aplsdlqgma-as.a.run.app/chat';
    try {
      final response = await http.post(
        Uri.parse(url),
        headers: {'Content-Type': 'application/json'},
        body: jsonEncode({'text': message}),
      );

      if (response.statusCode == 200) {
        final reply = jsonDecode(response.body)['reply'].trim();

        // Add user's message
        setState(() {
          messages.add({'role': 'user', 'content': message});

          // Remove user's input from the bot's reply if it's present
          String cleanReply = reply;
          if (reply.toLowerCase().startsWith(message.toLowerCase())) {
            cleanReply = reply.substring(message.length).trim();
          }

          // Add bot's reply if it's different from the user's message
          if (cleanReply.isNotEmpty && cleanReply.toLowerCase() !=
message.toLowerCase()) {
            messages.add({'role': 'bot', 'content': cleanReply});
          } else {
            messages.add({'role': 'bot', 'content': "I have no new response for that."});
          }
        });
      } else {
        _showErrorDialog('Failed to send message: ${response.reasonPhrase}');
      }
    } catch (e) {
      _showErrorDialog('Exception during message sending: $e');
    } finally {
      setState(() {
        isLoading = false;
      });
    }
  }

  Future<void> endChat() async {
    final messagesText = messages.where((msg) => msg['role'] == 'user').map((msg)
=> msg['content']).toList();
    final sentimentUrl = 'https://my-app-aplsdlqgma-as.a.run.app/sentiment';

    try {
      final response = await http.post(
        Uri.parse(sentimentUrl),
        headers: {'Content-Type': 'application/json'},
        body: jsonEncode({'texts': messagesText}),
      );

      if (response.statusCode == 200) {
        final sentiment = jsonDecode(response.body)['most_frequent_sentiment'];
        _showInfoDialog('Chat Ended', 'Sentiment Analysis: $sentiment');
      } else {
        _showErrorDialog('Failed to perform sentiment analysis:
${response.reasonPhrase}');
      }
    } catch (e) {
      _showErrorDialog('Exception during sentiment analysis: $e');
    }
  }

  void _showErrorDialog(String message) {
    showDialog(
      context: context,
      builder: (BuildContext context) => AlertDialog(
        title: Text('Error'),
        content: Text(message),
        actions: <Widget>[
          TextButton(
            child: Text('OK'),
            onPressed: () {
              Navigator.of(context).pop();
            },
          ),
        ],
      ),
    );
  }

  void _showInfoDialog(String title, String message) {
    showDialog(
      context: context,
      builder: (BuildContext context) => AlertDialog(
        title: Text(title),
        content: Text(message),
        actions: <Widget>[
          TextButton(
            child: Text('OK'),
            onPressed: () {
              Navigator.of(context).pop();
            },
          ),
        ],
      ),
    );
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Chatbox'),
      ),
      body: Column(
        children: <Widget>[
          // Instruction Text at the Top
          Padding(
            padding: const EdgeInsets.all(16.0),
            child: Text(
              'To perform sentiment analysis on your chat, press the "End Chat" button.',
              style: TextStyle(fontSize: 16, fontWeight: FontWeight.w400),
            ),
          ),
          Expanded(
            child: Column(
              children: <Widget>[
                // Chat Messages List
                Expanded(
                  child: ListView.builder(
                    itemCount: messages.length,
                    itemBuilder: (context, index) {
                      final message = messages[index];
                      return ListTile(
                        title: Text(
                          message['role'] == 'user' ? 'You' : 'Bot',
                          style: TextStyle(fontWeight: FontWeight.bold),
                        ),
                        subtitle: Text(message['content'] ?? "),
                      );
                    },
                  ),
                ),
                if (isLoading) CircularProgressIndicator(),
              ],
            ),
          ),
          Padding(
            padding: const EdgeInsets.all(8.0),
            child: Row(
              children: <Widget>[
                Expanded(
                  child: TextField(
                    controller: _controller,
                    decoration: InputDecoration(
                      hintText: 'Type a message...',
                    ),
                  ),
                ),
                IconButton(
                  icon: Icon(Icons.send),
                  onPressed: () {
                    if (_controller.text.isNotEmpty) {
                      sendMessage(_controller.text);
                      _controller.clear();
                    }
                  },
                ),
                IconButton(
                  icon: Icon(Icons.stop),
                  onPressed: () {
                    endChat(); // Trigger sentiment analysis on chat end
                  },
                ),
              ],
            ),
          ),
        ],
      ),
    );
  }
}
```

home_screen.dart:

```
import 'package:flutter/material.dart';
import 'main.dart'; // Ensure ChatScreen is properly imported from main.dart
import 'self_screening_test_screen.dart'; // Ensure SelfScreeningTestScreen is
imported correctly
import 'helpline_screen.dart'; // Ensure HelplineScreen is imported correctly

class HomeScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Buddy'),
        centerTitle: true,
      ),
      body: Container(
        decoration: BoxDecoration(
          gradient: LinearGradient(
            colors: [Colors.blue[100]!, Colors.white],
            begin: Alignment.topLeft,
            end: Alignment.bottomRight,
          ),
        ),
        child: Center(
          child: SingleChildScrollView(
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              children: <Widget>[
                SizedBox(height: 20),
                Image.asset(
                  'assets/logo.png', // Ensure the logo path is correct
                  width: 150,
                  height: 150,
                ),
                SizedBox(height: 20),
                Text(
                  'Welcome to Your Virtual Buddy!',
                  style: TextStyle(
                    fontSize: 24,
                    fontWeight: FontWeight.bold,
                    color: Colors.blue[800], // Ensure this color is defined
                  ),
                ),
                SizedBox(height: 30),
                _buildCard(
                  context,
                  icon: Icons.chat,
                  title: 'Start to chat',
                  subtitle: 'Feel bored? Come chat with me!',
                  onTap: () {
                    Navigator.push(
                      context,
                      MaterialPageRoute(builder: (context) => ChatScreen()),
                    );
                  },
                ),
                SizedBox(height: 20),
                _buildCard(
                  context,
                  icon: Icons.health_and_safety,
                  title: 'Self-Screening Test',
                  subtitle: 'This self-test is provided by Mentari Malaysia.\nWant to know
your mental health status? Here!',
                  onTap: () {
                    Navigator.push(
                      context,
                      MaterialPageRoute(builder: (context) => SelfScreeningTestScreen()),
                    );
                  },
                ),
                SizedBox(height: 20),
                _buildCard(
                  context,
                  icon: Icons.help,
                  title: 'Find a Helpline',
                  subtitle: 'Looking for help? Here!',
                  onTap: () {
                    Navigator.push(
                      context,
                      MaterialPageRoute(builder: (context) => HelplineScreen()),
                    );
                  },
                ),
              ],
            ),
          ),
        ),
      ),
    );
  }

  Widget _buildCard(
    BuildContext context, {
    required IconData icon,
    required String title,
    required String subtitle,
  required VoidCallback onTap,
 }) {
  return Card(
   margin: EdgeInsets.symmetric(vertical: 10, horizontal: 20),
   elevation: 5,
   child: ListTile(
    contentPadding: EdgeInsets.all(16),
    leading: Icon(icon, size: 40, color: Colors.blue),
    title: Text(title),
    subtitle: Text(subtitle, style: TextStyle(fontSize: 16)),
    onTap: onTap,
   ),
  );
 }
}
```

helpline_screen.dart:

```
import 'package:flutter/material.dart';
import 'package:flutter_inappwebview/flutter_inappwebview.dart';
import 'package:url_launcher/url_launcher.dart';

class HelplineScreen extends StatelessWidget {
  final String counsellingUrl = 'https://utarhospital.org.my/counselling-services/';
  final String instagramUrl = 'https://www.instagram.com/utarhospitalofficial/';
  final String facebookUrl = 'https://www.facebook.com/hospitalutarofficial';
  final String phoneNumber = 'tel:+6056181300'; // Direct phone number link

  Future<void> _launchURL(String url) async {
    final Uri uri = Uri.parse(url);
    if (await canLaunchUrl(uri)) {
      await launchUrl(uri, mode: LaunchMode.externalApplication);
    } else {
      print('Could not launch $url');
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Find a Helpline'),
        backgroundColor: Colors.lightBlue, // Updated AppBar color
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          crossAxisAlignment: CrossAxisAlignment.center,
          children: [
            // Counselling Services Button
            ElevatedButton(
              onPressed: () {
                Navigator.push(
                  context,
                  MaterialPageRoute(
                    builder: (context) => WebViewScreen(url: counsellingUrl),
                  ),
                );
              },
              style: ElevatedButton.styleFrom(
                backgroundColor: Colors.lightBlue, // Button background color
                foregroundColor: Colors.white, // Text color
                padding: EdgeInsets.symmetric(vertical: 15, horizontal: 25),
                textStyle: TextStyle(fontSize: 16, fontWeight: FontWeight.bold),
              ),
              child: Text('Counselling Services'),
            ),
            SizedBox(height: 8),
            Text(
              'Know more about what UTAR Hospital counselling services offer.',
              textAlign: TextAlign.center,
              style: TextStyle(fontSize: 16, color: Colors.grey[700]),
            ),
            SizedBox(height: 20),

            // Contact Us Section
            Text(
              'Contact Us:',
              style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold, color:
Colors.lightBlue),
            ),
            SizedBox(height: 16),
            ElevatedButton(
              onPressed: () => _launchURL(instagramUrl),
              style: ElevatedButton.styleFrom(
                backgroundColor: Colors.lightBlue, // Button background color
                foregroundColor: Colors.white, // Text color
                padding: EdgeInsets.symmetric(vertical: 15, horizontal: 25),
                textStyle: TextStyle(fontSize: 16, fontWeight: FontWeight.bold),
              ),
              child: Text('Instagram'),
            ),
            SizedBox(height: 8),
            ElevatedButton(
              onPressed: () => _launchURL(facebookUrl),
              style: ElevatedButton.styleFrom(
                backgroundColor: Colors.lightBlue, // Button background color
                foregroundColor: Colors.white, // Text color
                padding: EdgeInsets.symmetric(vertical: 15, horizontal: 25),
                textStyle: TextStyle(fontSize: 16, fontWeight: FontWeight.bold),
              ),
              child: Text('Facebook'),
            ),
            SizedBox(height: 8),
            ElevatedButton(
              onPressed: () => _launchURL(phoneNumber),
              style: ElevatedButton.styleFrom(
                backgroundColor: Colors.lightBlue, // Button background color
                foregroundColor: Colors.white, // Text color
                padding: EdgeInsets.symmetric(vertical: 15, horizontal: 25),
                textStyle: TextStyle(fontSize: 16, fontWeight: FontWeight.bold),
              ),
              child: Text('Call Us'),
            ),
          ],
        ),
      ),
    );
  }
}

class WebViewScreen extends StatelessWidget {
  final String url;

  WebViewScreen({required this.url});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Counselling Services'),
        backgroundColor: Colors.lightBlue, // Updated AppBar color
      ),
      body: InAppWebView(
        initialUrlRequest: URLRequest(
          url: WebUri(url),
        ),
        initialOptions: InAppWebViewGroupOptions(
          crossPlatform: InAppWebViewOptions(
            javaScriptEnabled: true,
          ),
        ),
        // Add any additional configurations or overrides here if needed
      ),
    );
  }
}
```

Self_screening_test_screen.dart:

```
import 'package:flutter/material.dart';
import 'package:flutter_inappwebview/flutter_inappwebview.dart';

class SelfScreeningTestScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Self-screening Test'),
        leading: IconButton(
          icon: Icon(Icons.arrow_back),
          onPressed: () {
            Navigator.pop(context);
          },
        ),
      ),
      body: InAppWebView(
        initialUrlRequest: URLRequest(url: WebUri("https://mentari.moh.gov.my/self-
test/")),
        initialOptions: InAppWebViewGroupOptions(
          crossPlatform: InAppWebViewOptions(
            javaScriptEnabled: true,
          ),
        ),
      ),
    );
  }
}
```

Weekly Reports

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	Study week no.: 1
Student Name & ID: Koey Hor Yee & 2006028	
Supervisor: Dr Siti Nurlaili Karim	
Project Title: Empowering Mental Health Care with Artificial Intelligence Tool	

1. WORK DONE

Built a local chat app prototype with Flutter, enabling user input and displaying responses from a local Flask backend.

2. WORK TO BE DONE

Moving forward, the upcoming week will focus on evaluating the model's responses to ensure they are informative, engaging, and align with user expectations.

3. PROBLEMS ENCOUNTERED

One significant challenge encountered was the app's limited accessibility, as it currently lacks global access due to the absence of a cloud service.

4. SELF EVALUATION OF THE PROGRESS

Despite challenges, completing the basic functionality of the app marks significant progress. The focus remains on meeting UI design goals and integrating additional features within the project timeline.

Siti Nurlaili

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	Study week no.: 2
Student Name & ID: Koey Hor Yee & 2006028	
Supervisor: Dr Siti Nurlaili Karim	
Project Title: Empowering Mental Health Care with Artificial Intelligence Tool	

1. WORK DONE

In Week 2, I conducted extensive testing of the Flutter app, Flask backend, and integrated models. This testing phase highlighted issues related to the accuracy and responsiveness of the models used in the app.

2. WORK TO BE DONE

The subsequent week will involve further refinement of the models to improve their performance in real-time conversation scenarios. Adjustments in model parameters and continuous testing will be prioritized to enhance user interaction quality.

3. PROBLEMS ENCOUNTERED

Challenges included optimizing model parameters to achieve desired response quality without compromising app performance. Iterative testing and adjustment were necessary to achieve optimal results.

4. SELF EVALUATION OF THE PROGRESS

Progress in identifying model-related challenges demonstrates a proactive approach to ensuring app functionality aligns with user expectations. Continued refinement will support improved app performance and user satisfaction.

Siti Nurlaili

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	Study week no.: 3
Student Name & ID: Koey Hor Yee & 2006028	
Supervisor: Dr Siti Nurlaili Karim	
Project Title: Empowering Mental Health Care with Artificial Intelligence Tool	

1. WORK DONE

In Week 3, I successfully set up Docker to containerize the Flask application and integrated models. Additionally, I built the Docker image to prepare for deployment on a cloud platform, ensuring compatibility and functionality.

2. WORK TO BE DONE

The next steps involve deploying the Docker image on the selected cloud platform and configuring environment variables for secure and efficient operation.

3. PROBLEMS ENCOUNTERED

Decision-making regarding the optimal cloud platform posed challenges, considering factors such as ease of use, cost-effectiveness, and scalability. Research and consultation were crucial in making informed decisions.

4. SELF EVALUATION OF THE PROGRESS

Progress in Docker setup and image building supports timely deployment preparations. Addressing cloud platform considerations ensures readiness for the next phase, aligning with project milestones and objectives.

Siti Nurlaili

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	Study week no.: 4
Student Name & ID: Koey Hor Yee & 2006028	
Supervisor: Dr Siti Nurlaili Karim	
Project Title: Empowering Mental Health Care with Artificial Intelligence Tool	

1. WORK DONE

In Week 4, I successfully deployed the Dockerized Flask application on Google Cloud Run.

2. WORK TO BE DONE

The next steps involve building the basic Flutter APK by focusing on the main.dart file. This will prepare the app for distribution and further testing.

3. PROBLEMS ENCOUNTERED

Troubleshooting deployment errors related to IAM policies and ensuring the Docker image was correctly configured for the cloud environment posed significant challenges.

4. SELF EVALUATION OF THE PROGRESS

Successfully deploying the application on Google Cloud Run marks a crucial milestone. Overcoming the deployment challenges demonstrates problem-solving capabilities and ensures the backend is ready for further development and integration with the Flutter app.

Siti Nurlaili

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	Study week no.: 5
Student Name & ID: Koey Hor Yee & 2006028	
Supervisor: Dr Siti Nurlaili Karim	
Project Title: Empowering Mental Health Care with Artificial Intelligence Tool	

1. WORK DONE

In Week 5, I successfully built the Flutter APK, focusing on modifying the main.dart file. This step was essential in preparing the app for user testing and distribution.

2. WORK TO BE DONE

The next steps involve finding a suitable mental health test to integrate into the app. This requires selecting a test that aligns with the app's goals and ensuring it can be legally and ethically included.

3. PROBLEMS ENCOUNTERED

A significant challenge is the need to consider the copyright issues associated with the mental health test. It's crucial to determine how to add the test to the app while respecting intellectual property rights and obtaining the necessary permissions.

4. SELF EVALUATION OF THE PROGRESS

Building the Flutter APK marks significant progress, demonstrating the app's readiness for user testing and further enhancements. Addressing copyright concerns and integrating a suitable mental health test will add valuable functionality to the app, aligning with its goals and user needs.

Siti Nurlaili

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	Study week no.: 6
Student Name & ID: Koey Hor Yee & 2006028	
Supervisor: Dr Siti Nurlaili Karim	
Project Title: Empowering Mental Health Care with Artificial Intelligence Tool	

1. WORK DONE

In Week 6, I successfully found a suitable mental health test to integrate into the app. I created an in-app browser, allowing users to take the test directly within the app, ensuring a seamless user experience.

2. WORK TO BE DONE

The next steps involve implementing a home page that allows users to select between chatting with the bot or taking the mental health test. This home page will help users navigate the app more efficiently.

3. PROBLEMS ENCOUNTERED

A significant challenge this week was separating the chatbot function from the test function within the same app to reduce user confusion. Ensuring that both functionalities are accessible yet distinct required careful planning and design.

4. SELF EVALUATION OF THE PROGRESS

Finding and integrating the mental health test marks substantial progress, adding valuable functionality to the app. Creating the in-app browser enhances user experience by keeping everything within the app. The upcoming task of implementing a home page will further improve usability and navigation.

Siti Nurlaili

Supervisor’s signature

Student’s signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	Study week no.: 7
Student Name & ID: Koey Hor Yee & 2006028	
Supervisor: Dr Siti Nurlaili Karim	
Project Title: Empowering Mental Health Care with Artificial Intelligence Tool	

\

1. WORK DONE

In Week 7, I completed the implementation of a home page for the app, allowing users to select between the chatbot and the mental health test.

2. WORK TO BE DONE

The next steps involve continued testing of the app to identify and address any issues.

3. PROBLEMS ENCOUNTERED

I realized that the chatbot's responses were not intelligent enough. Specifically, the bot occasionally repeats the user's input, which can lead to a subpar user experience.

4. SELF EVALUATION OF THE PROGRESS

Completing the home page is a significant milestone, as it enhances the app's usability and navigation. Continued testing will help identify areas for improvement, particularly in the chatbot's intelligence.

Siti Nurlaili

Supervisor’s signature

Student’s signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	Study week no.: 8
Student Name & ID: Koey Hor Yee & 2006028	
Supervisor: Dr Siti Nurlaili Karim	
Project Title: Empowering Mental Health Care with Artificial Intelligence Tool	

1. WORK DONE

In Week 8, I focused on adjusting the parameters of the DialoGPT model to make the responses more contextually appropriate. This involved fine-tuning the model's parameters such as temperature, top_p, top_k, and repetition_penalty to improve the coherence and relevance of the generated responses.

2. WORK TO BE DONE

The next task is to change the sentiment label from a numerical representation to actual and remove unnecessary labels such as "love" and "surprise."

3. PROBLEMS ENCOUNTERED

A significant challenge was realizing that the sentiment analysis model returns numerical values instead of actual labels.

4. SELF EVALUATION OF THE PROGRESS

Adjusting the DialoGPT parameters has significantly improved the chatbot's response quality. The next steps will further enhance the app's functionality and user experience.

Siti Nurlaili

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	Study week no.: 9
Student Name & ID: Koey Hor Yee & 2006028	
Supervisor: Dr Siti Nurlaili Karim	
Project Title: Empowering Mental Health Care with Artificial Intelligence Tool	

1. WORK DONE

In Week 9, I successfully changed the sentiment analysis output from numerical values to actual labels. This improvement makes the sentiment feedback more understandable and useful for users.

2. WORK TO BE DONE

The next step is to plan for the UI/UX design of the app. This involves creating a more intuitive and visually appealing interface that enhances user interaction and improves the overall user experience.

3. PROBLEMS ENCOUNTERED

The primary challenge encountered this week was the time constraint for UI/UX design. Balancing feature development with designing an intuitive and user-friendly interface has been difficult.

4. SELF EVALUATION OF THE PROGRESS

Successfully converting sentiment analysis outputs to labels marks a significant improvement in the app's usability. Planning for the UI/UX design is crucial to ensure the app is both functional and easy to use, although managing time effectively for this remains a priority.

Siti Nurlaili

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	Study week no.: 10
Student Name & ID: Koey Hor Yee & 2006028	
Supervisor: Dr Siti Nurlaili Karim	
Project Title: Empowering Mental Health Care with Artificial Intelligence Tool	

1. WORK DONE

In week 10, I paused the UI/UX design tasks and added a new in-app browser feature. This feature allows users to easily contact hotlines directly from within the app. The browser was implemented to open a webpage where users can find hotline information.

2. WORK TO BE DONE

For week 11, the plan is to adjust the in-app browser settings to allow users to directly trigger phone calls, open hotline websites, and launch WhatsApp chats from within the app.

3. PROBLEMS ENCOUNTERED

The main challenge encountered in week 10 was that users could only view the hotline website and phone numbers but could not directly initiate calls or open WhatsApp chats. This limitation required adjustments to the in-app browser settings to allow direct interactions.

4. SELF EVALUATION OF THE PROGRESS

The addition of the in-app browser feature was a crucial step in enhancing the app’s functionality by providing easy access to helpline resources.

Siti Nurlaili

Supervisor’s signature

Student’s signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	Study week no.: 11
Student Name & ID: Koey Hor Yee & 2006028	
Supervisor: Dr Siti Nurlaili Karim	
Project Title: Empowering Mental Health Care with Artificial Intelligence Tool	

1. WORK DONE

In week 11, I successfully implemented improvements to the in-app browser feature. The app now allows users to directly initiate phone calls, open hotline websites, and start WhatsApp chats with helpline numbers.

2. WORK TO BE DONE

For week 12, I will resume the UI/UX design work to improve the overall look and feel of the app. The goal is to make the interface more intuitive and user-friendly, addressing any usability issues identified in previous versions.

3. PROBLEMS ENCOUNTERED

In week 11, a significant challenge was making improvements to the UI/UX design of the app. It was difficult to achieve a satisfactory level of improvement in the user interface and experience, which will be addressed in the upcoming week (week 12).

4. SELF EVALUATION OF THE PROGRESS

The progress made in week 11 was substantial, addressing the limitations from the previous week by allowing direct calls, website access, and WhatsApp chats. These enhancements greatly improved the app's usability.

Siti Nurlaili

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	Study week no.: 12
Student Name & ID: Koey Hor Yee & 2006028	
Supervisor: Dr Siti Nurlaili Karim	
Project Title: Empowering Mental Health Care with Artificial Intelligence Tool	

1. WORK DONE

n week 12, I completed all remaining implementation tasks, including finalizing the UI/UX design improvements. The app interface was refined to enhance usability, with user feedback considered during the design process. All major features, including the chatbot, sentiment analysis, and helpline integration, were tested and optimized.

2. WORK TO BE DONE

For week 13, the focus will shift to writing the final project report. This includes documenting all features, testing results, and challenges encountered during the project.

3. PROBLEMS ENCOUNTERED

There were no significant technical challenges during week 12, as most issues were resolved in earlier stages. Minor adjustments were made to ensure a smooth user experience across all devices.

4. SELF EVALUATION OF THE PROGRESS

Week 12 marked the completion of the app's development phase. All planned features have been implemented, and the UI/UX is now more intuitive, creating a user-friendly experience.

Siti Nurlaili

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3, Y3	Study week no.: 13
Student Name & ID: Koey Hor Yee & 2006028	
Supervisor: Dr Siti Nurlaili Karim	
Project Title: Empowering Mental Health Care with Artificial Intelligence Tool	

1. WORK DONE

In week 13, I focused on finalizing the project report. All sections of the report were completed, including a detailed discussion of system features, testing outcomes, implementation challenges, and future recommendations.

2. WORK TO BE DONE

With all project tasks and the final report completed, no further work is planned.

3. PROBLEMS ENCOUNTERED

The main challenge this week was ensuring that all aspects of the project were thoroughly documented in the report, which required careful review and editing.

4. SELF EVALUATION OF THE PROGRESS

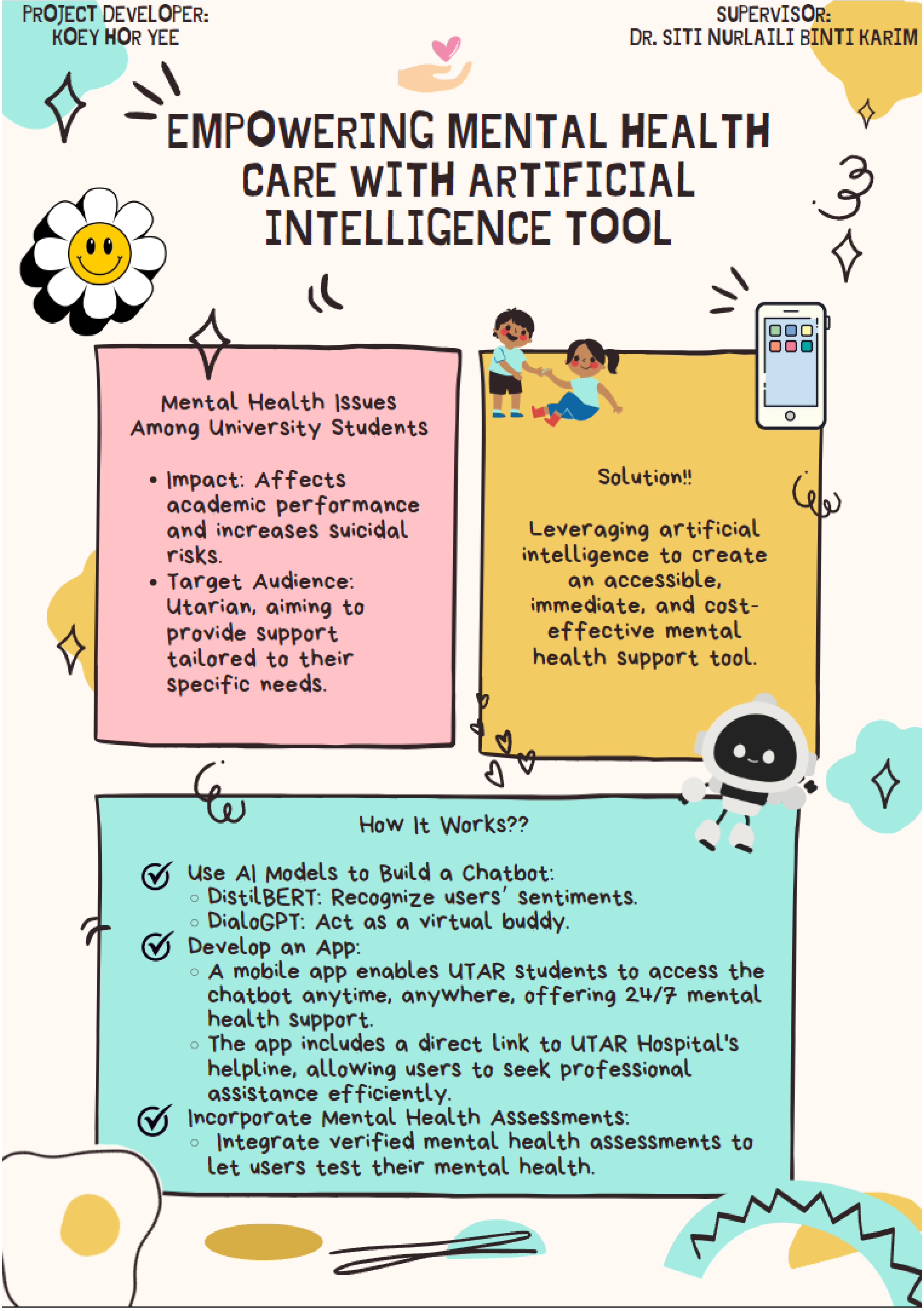
Week 13 was productive, as the final report was successfully completed. This marks the conclusion of the project, with all objectives met and documented.

Siti Nurlaili

Supervisor's signature

Student's signature

POSTER



PROJECT DEVELOPER:
KOEY HOR YEE

SUPERVISOR:
DR. SITI NURLAILI BINTI KARIM

EMPOWERING MENTAL HEALTH CARE WITH ARTIFICIAL INTELLIGENCE TOOL

Mental Health Issues Among University Students

- Impact: Affects academic performance and increases suicidal risks.
- Target Audience: Utarian, aiming to provide support tailored to their specific needs.

Solution!!

Leveraging artificial intelligence to create an accessible, immediate, and cost-effective mental health support tool.

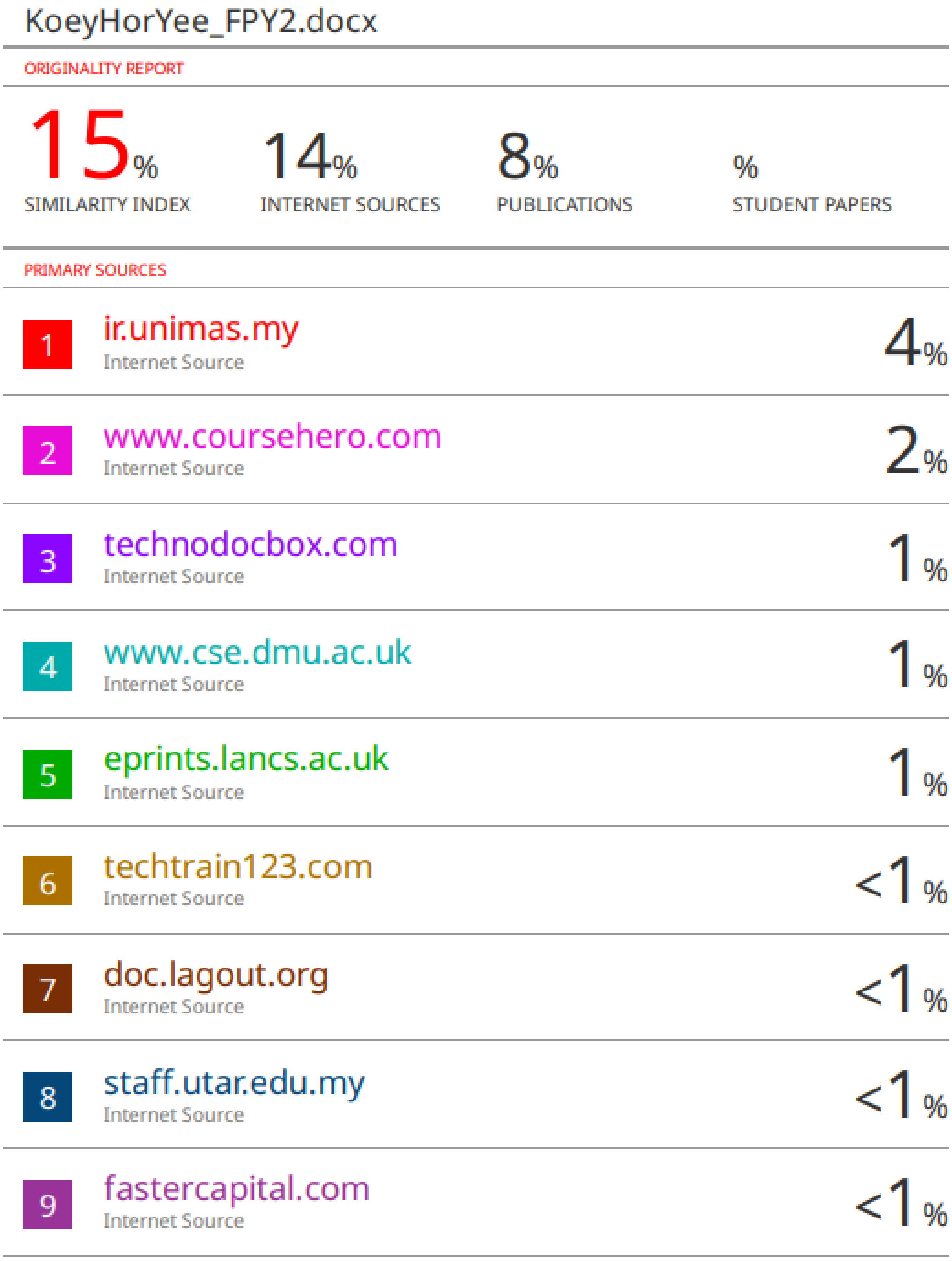
How It Works??

✓ Use AI Models to Build a Chatbot:
- ○ DistilBERT: Recognize users' sentiments.
- ○ DialoGPT: Act as a virtual buddy.

✓ Develop an App:
- ○ A mobile app enables UTAR students to access the chatbot anytime, anywhere, offering 24/7 mental health support.
- ○ The app includes a direct link to UTAR Hospital's helpline, allowing users to seek professional assistance efficiently.

✓ Incorporate Mental Health Assessments:
- ○ Integrate verified mental health assessments to let users test their mental health.

PLAGIARISM CHECK RESULT



KoeyHorYee_FPY2.docx

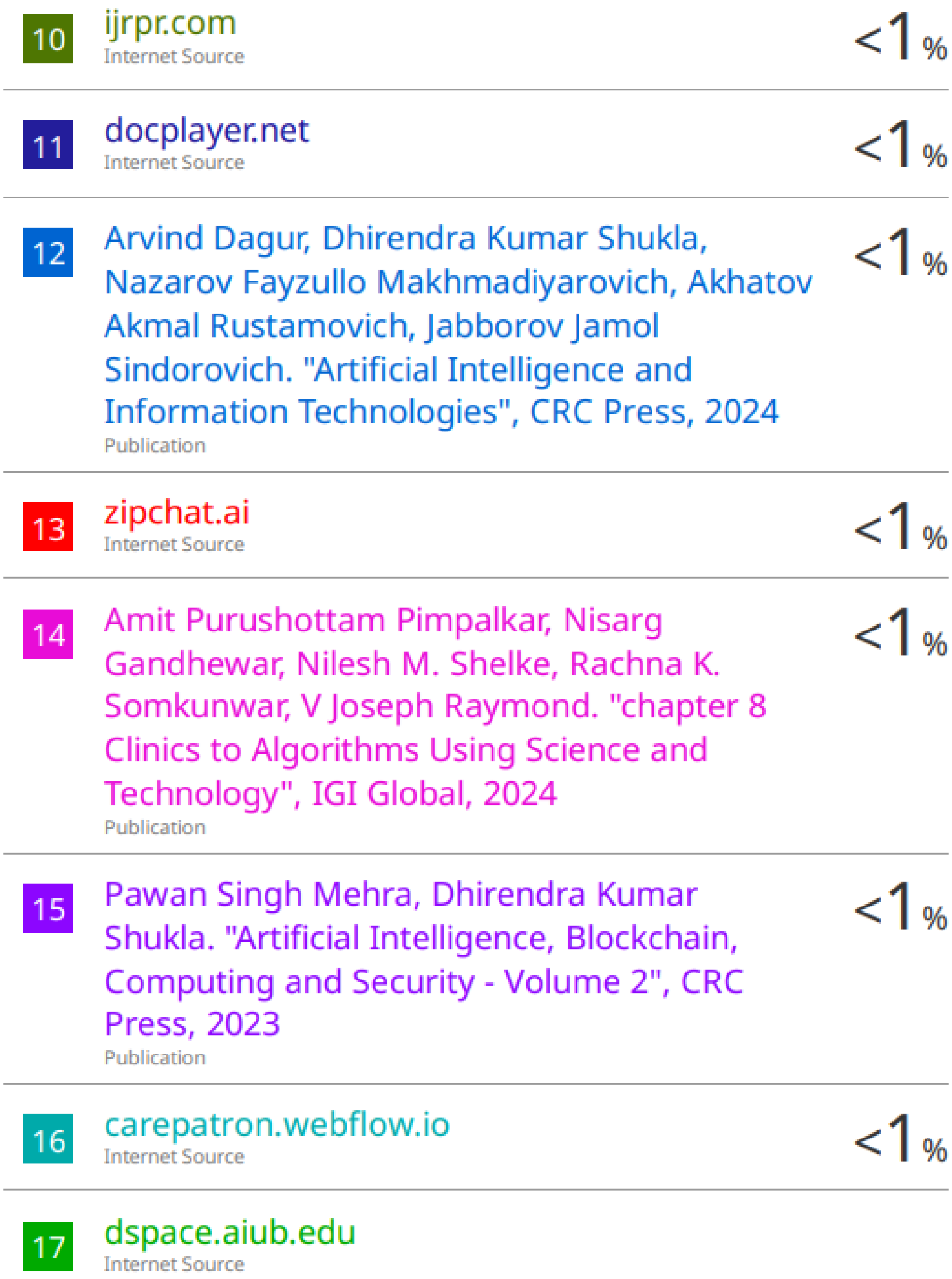
ORIGINALITY REPORT

15%	14%	8%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

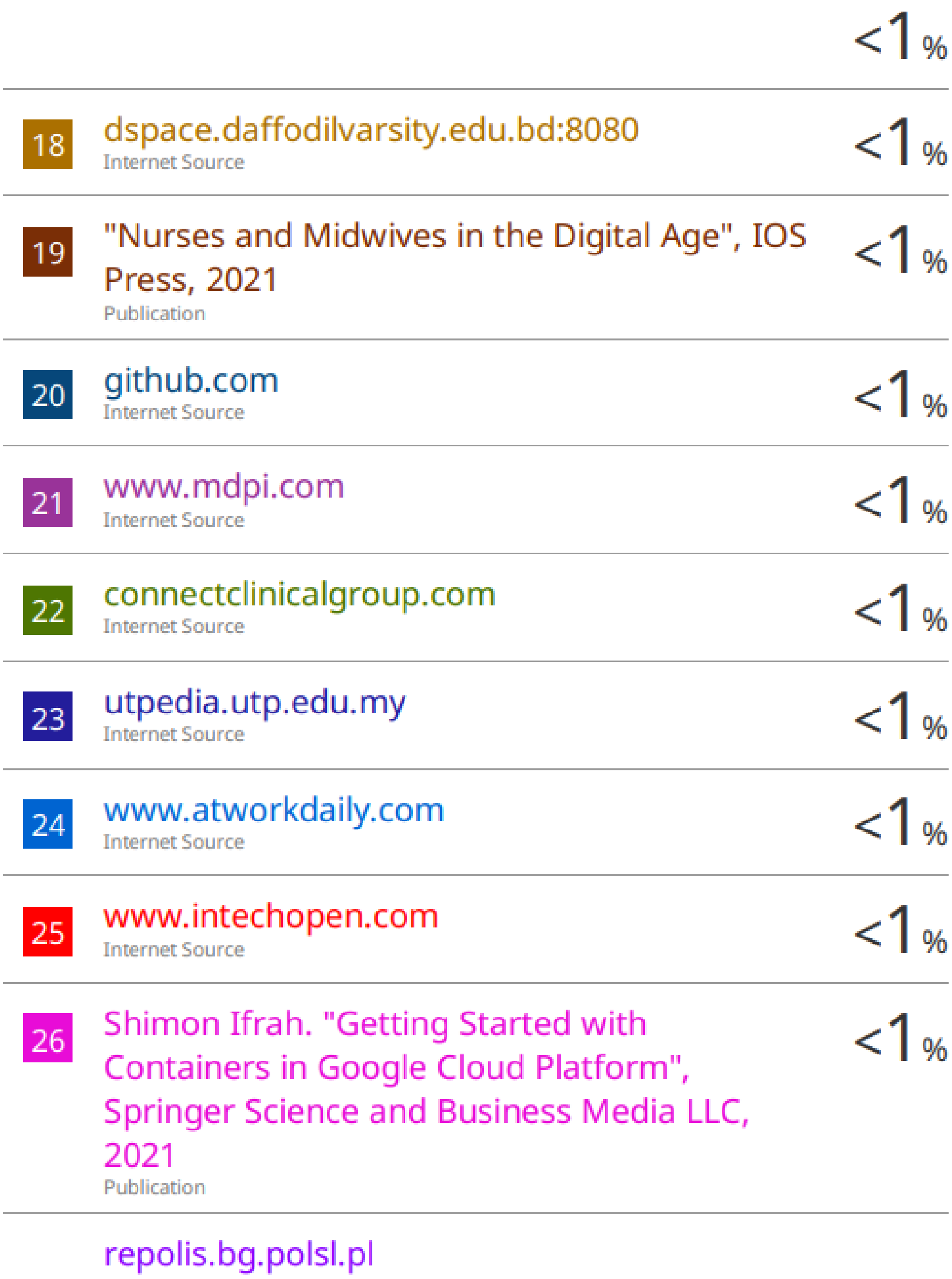
1	ir.unimas.my Internet Source	4%
2	www.coursehero.com Internet Source	2%
3	technodocbox.com Internet Source	1%
4	www.cse.dmu.ac.uk Internet Source	1%
5	eprints.lancs.ac.uk Internet Source	1%
6	techtrain123.com Internet Source	<1%
7	doc.lagout.org Internet Source	<1%
8	staff.utar.edu.my Internet Source	<1%
9	fastercapital.com Internet Source	<1%

PLAGIARISM CHECK RESULT



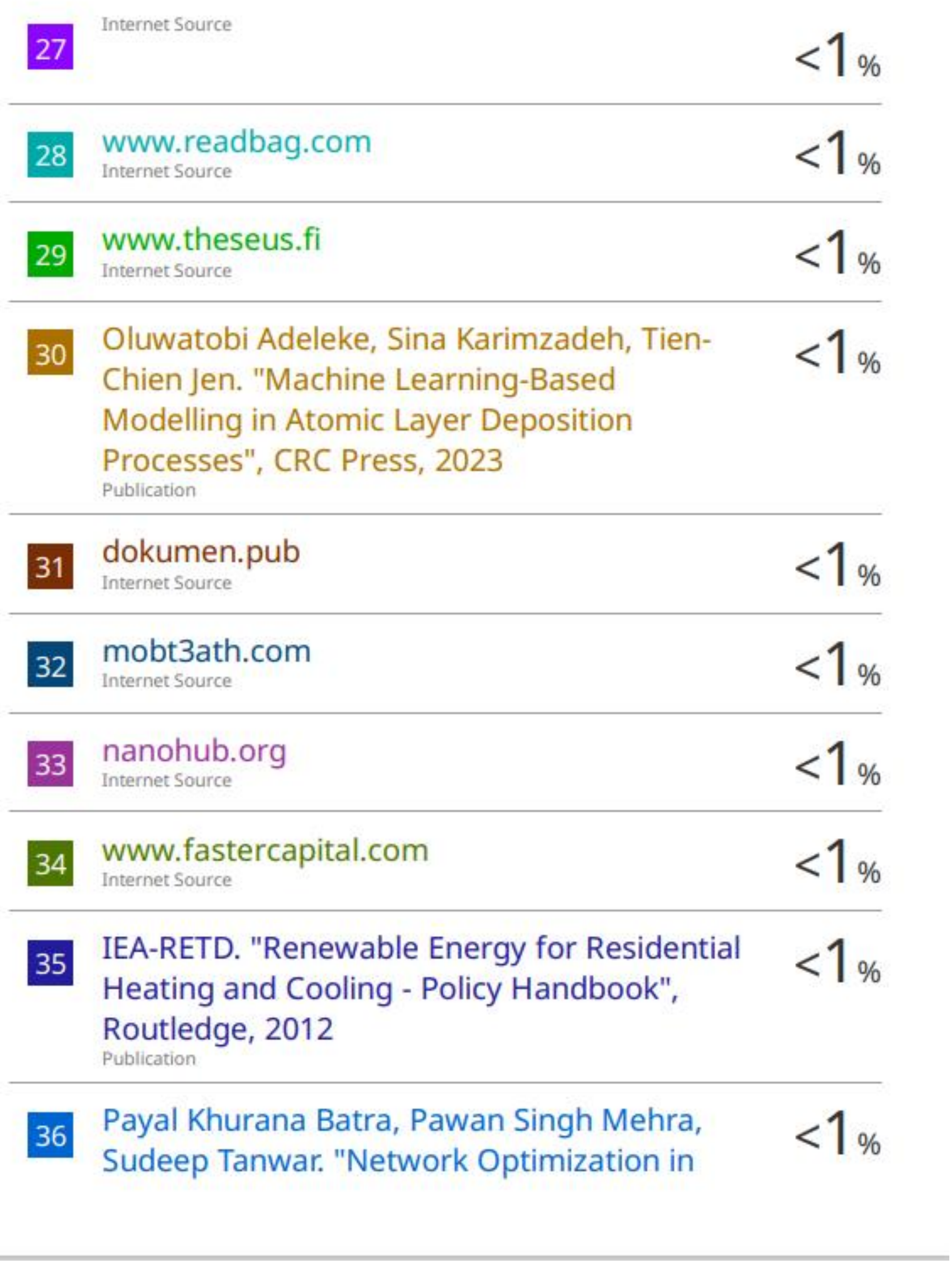
10	ijrpr.com Internet Source	<1 %
11	docplayer.net Internet Source	<1 %
12	Arvind Dagur, Dhirendra Kumar Shukla, Nazarov Fayzullo Makhmadiyarovich, Akhatov Akmal Rustamovich, Jabborov Jamol Sindorovich. "Artificial Intelligence and Information Technologies", CRC Press, 2024 Publication	<1 %
13	zipchat.ai Internet Source	<1 %
14	Amit Purushottam Pimpalkar, Nisarg Gandhewar, Nilesh M. Shelke, Rachna K. Somkunwar, V Joseph Raymond. "chapter 8 Clinics to Algorithms Using Science and Technology", IGI Global, 2024 Publication	<1 %
15	Pawan Singh Mehra, Dhirendra Kumar Shukla. "Artificial Intelligence, Blockchain, Computing and Security - Volume 2", CRC Press, 2023 Publication	<1 %
16	carepatron.webflow.io Internet Source	<1 %
17	dspace.aiub.edu Internet Source	

PLAGIARISM CHECK RESULT

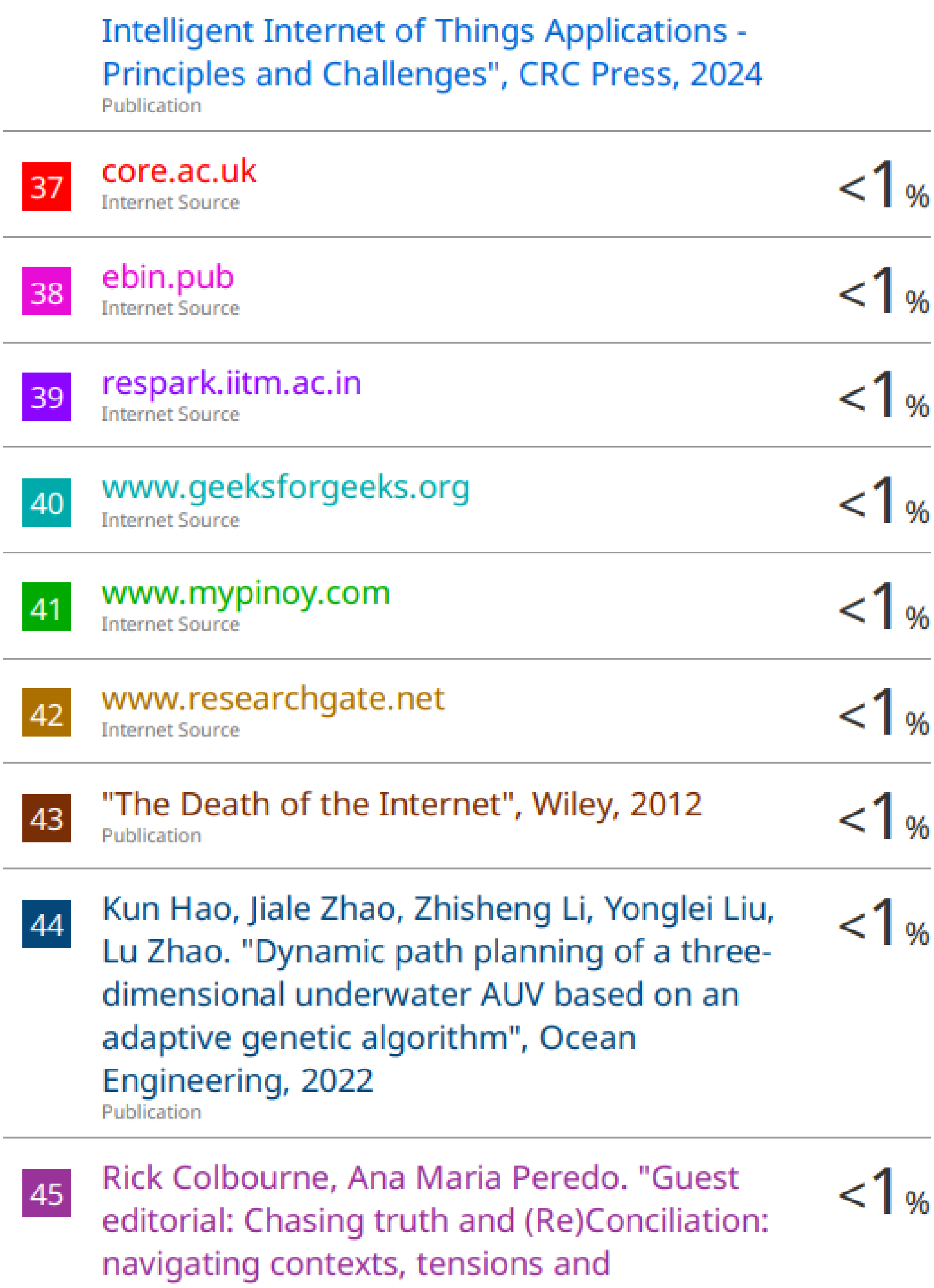


		<1 %
18	dspace.daffodilvarsity.edu.bd:8080 Internet Source	<1 %
19	"Nurses and Midwives in the Digital Age", IOS Press, 2021 Publication	<1 %
20	github.com Internet Source	<1 %
21	www.mdpi.com Internet Source	<1 %
22	connectclinicalgroup.com Internet Source	<1 %
23	utpedia.utp.edu.my Internet Source	<1 %
24	www.atworkdaily.com Internet Source	<1 %
25	www.intechopen.com Internet Source	<1 %
26	Shimon Ifrah. "Getting Started with Containers in Google Cloud Platform", Springer Science and Business Media LLC, 2021 Publication	<1 %
	repolis.bg.polsl.pl	

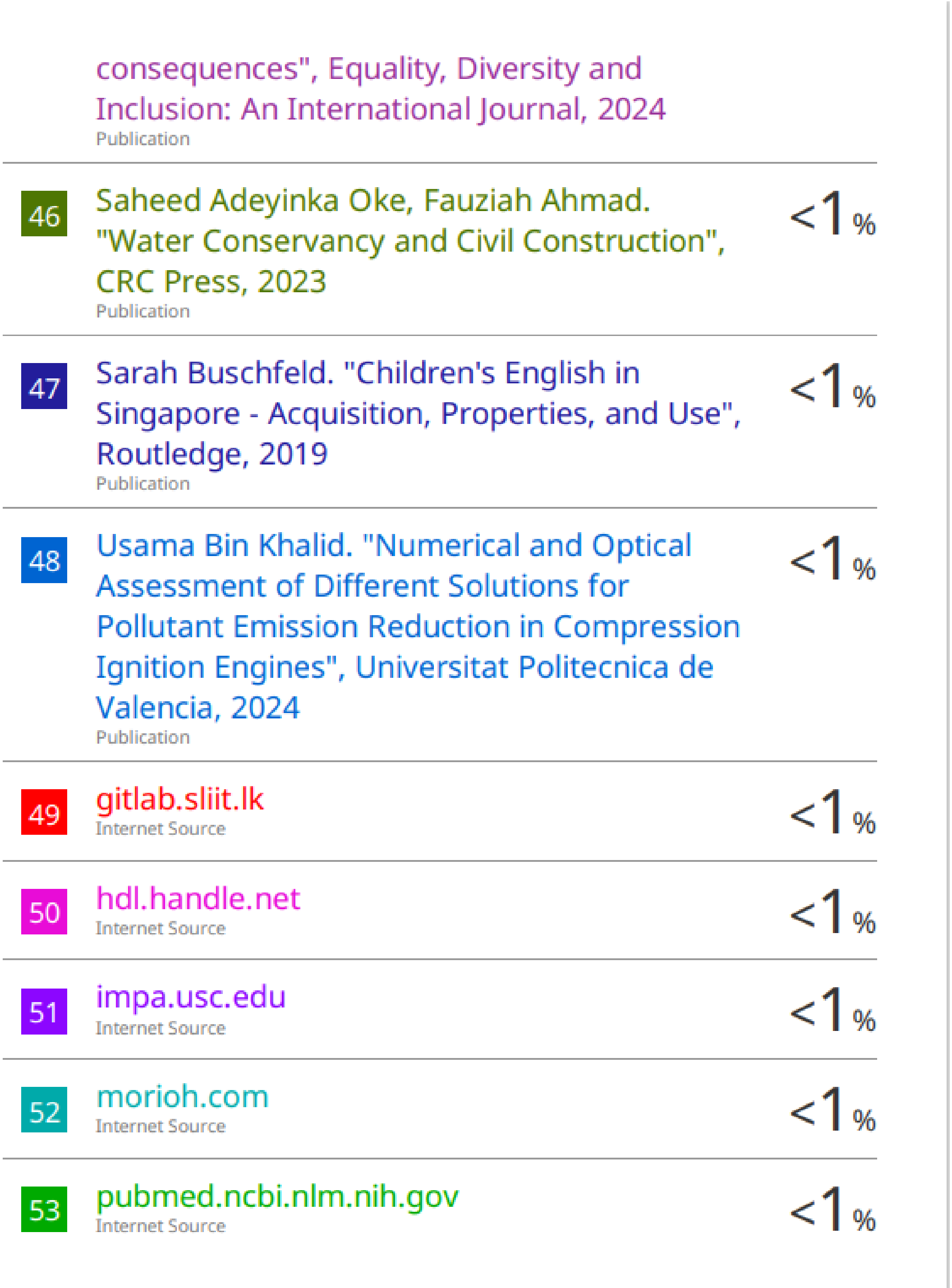
PLAGIARISM CHECK RESULT



27	Internet Source	<1 %
28	www.readbag.com Internet Source	<1 %
29	www.theseus.fi Internet Source	<1 %
30	Oluwatobi Adeleke, Sina Karimzadeh, Tien-Chien Jen. "Machine Learning-Based Modelling in Atomic Layer Deposition Processes", CRC Press, 2023 Publication	<1 %
31	dokumen.pub Internet Source	<1 %
32	mobt3ath.com Internet Source	<1 %
33	nanohub.org Internet Source	<1 %
34	www.fastercapital.com Internet Source	<1 %
35	IEA-RETD. "Renewable Energy for Residential Heating and Cooling - Policy Handbook", Routledge, 2012 Publication	<1 %
36	Payal Khurana Batra, Pawan Singh Mehra, Sudeep Tanwar. "Network Optimization in	<1 %

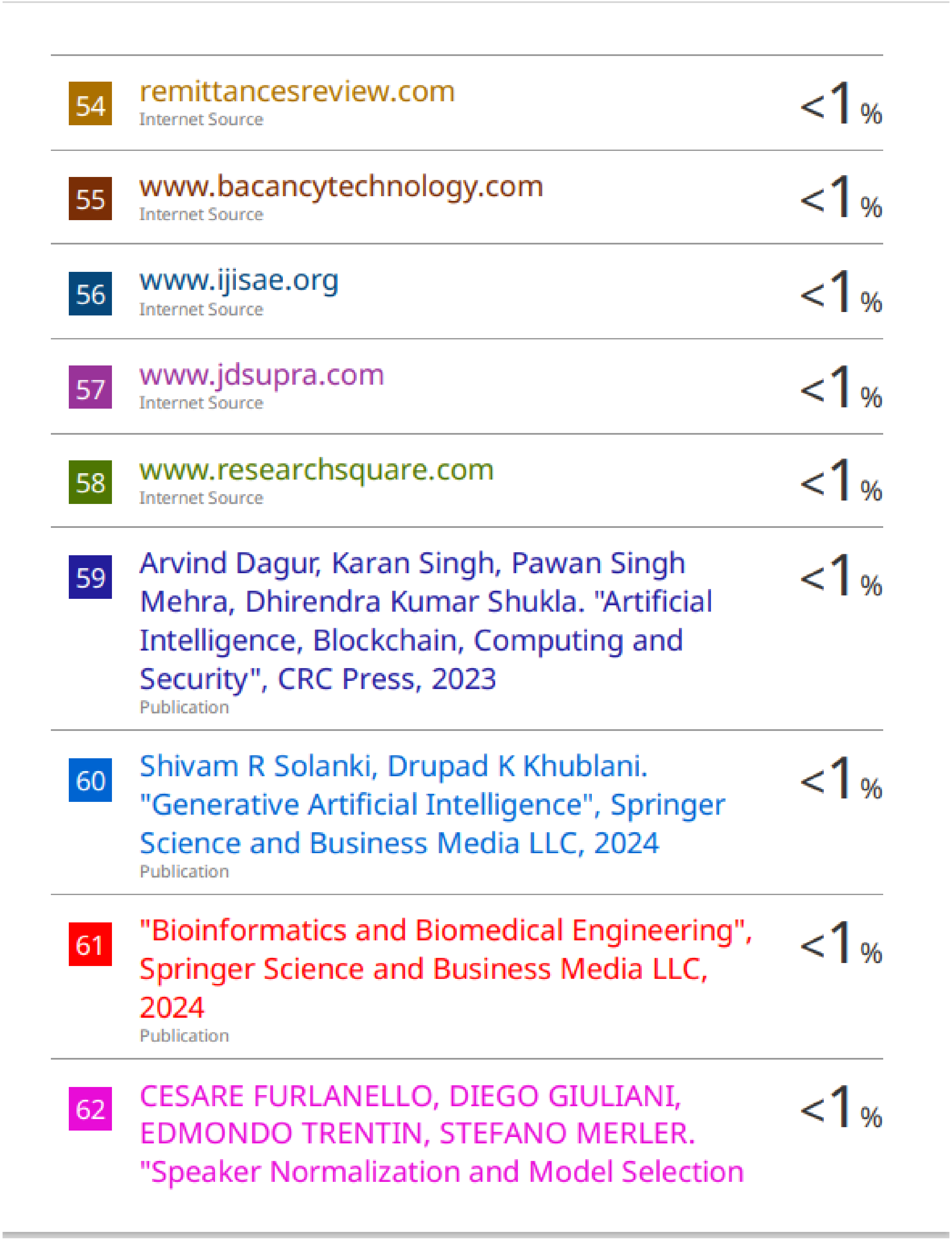


	Intelligent Internet of Things Applications - Principles and Challenges", CRC Press, 2024 Publication	
37	core.ac.uk Internet Source	<1 %
38	ebin.pub Internet Source	<1 %
39	respark.iitm.ac.in Internet Source	<1 %
40	www.geeksforgeeks.org Internet Source	<1 %
41	www.mypinoy.com Internet Source	<1 %
42	www.researchgate.net Internet Source	<1 %
43	"The Death of the Internet", Wiley, 2012 Publication	<1 %
44	Kun Hao, Jiale Zhao, Zhisheng Li, Yonglei Liu, Lu Zhao. "Dynamic path planning of a three-dimensional underwater AUV based on an adaptive genetic algorithm", Ocean Engineering, 2022 Publication	<1 %
45	Rick Colbourne, Ana Maria Peredo. "Guest editorial: Chasing truth and (Re)Conciliation: navigating contexts, tensions and	<1 %

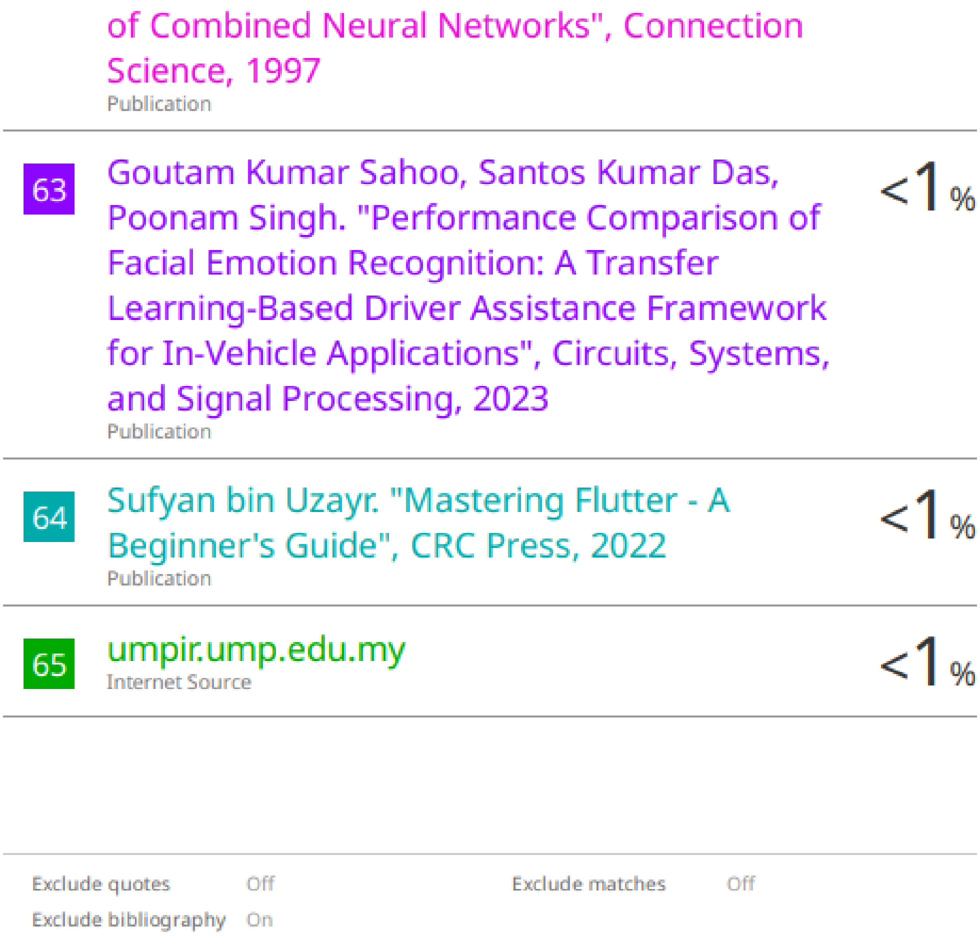


	consequences", Equality, Diversity and Inclusion: An International Journal, 2024 Publication	
46	Saheed Adeyinka Oke, Fauziah Ahmad. "Water Conservancy and Civil Construction", CRC Press, 2023 Publication	<1 %
47	Sarah Buschfeld. "Children's English in Singapore - Acquisition, Properties, and Use", Routledge, 2019 Publication	<1 %
48	Usama Bin Khalid. "Numerical and Optical Assessment of Different Solutions for Pollutant Emission Reduction in Compression Ignition Engines", Universitat Politecnica de Valencia, 2024 Publication	<1 %
49	gitlab.sliit.lk Internet Source	<1 %
50	hdl.handle.net Internet Source	<1 %
51	impa.usc.edu Internet Source	<1 %
52	morioh.com Internet Source	<1 %
53	pubmed.ncbi.nlm.nih.gov Internet Source	<1 %

PLAGIARISM CHECK RESULT



54	remittancesreview.com Internet Source	<1 %
55	www.bacancytechnology.com Internet Source	<1 %
56	www.ijisae.org Internet Source	<1 %
57	www.jdsupra.com Internet Source	<1 %
58	www.researchsquare.com Internet Source	<1 %
59	Arvind Dagur, Karan Singh, Pawan Singh Mehra, Dhirendra Kumar Shukla. "Artificial Intelligence, Blockchain, Computing and Security", CRC Press, 2023 Publication	<1 %
60	Shivam R Solanki, Drupad K Khublani. "Generative Artificial Intelligence", Springer Science and Business Media LLC, 2024 Publication	<1 %
61	"Bioinformatics and Biomedical Engineering", Springer Science and Business Media LLC, 2024 Publication	<1 %
62	CESARE FURLANELLO, DIEGO GIULIANI, EDMONDO TRENTIN, STEFANO MERLER. "Speaker Normalization and Model Selection	<1 %



	of Combined Neural Networks", Connection Science, 1997 Publication	
63	Goutam Kumar Sahoo, Santos Kumar Das, Poonam Singh. "Performance Comparison of Facial Emotion Recognition: A Transfer Learning-Based Driver Assistance Framework for In-Vehicle Applications", Circuits, Systems, and Signal Processing, 2023 Publication	<1 %
64	Sufyan bin Uzayr. "Mastering Flutter - A Beginner's Guide", CRC Press, 2022 Publication	<1 %
65	umpir.ump.edu.my Internet Source	<1 %

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		

PLAGIARISM CHECK RESULT

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	KOEY HOR YEE
ID Number(s)	2006028
Programme / Course	CS
Title of Final Year Project	Empowering Mental Health Care with Artificial Intelligence Tool

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 15 **%** **Similarity by source** Internet Sources: 14 % Publications: 8 % Student Papers: 0 %	The similarity index does not exceed the limit.
Number of individual sources listed of more than 3% similarity: 1	The similarity index for this individual source is 4%, insignificant similarity is identified.
Parameters of originality required and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Siti Nurlaili

Signature of Supervisor

Name: Dr Siti Nurlaili Binti Karim

Date: 12 September 2024

Signature of Co-Supervisor

Name:

Date:

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	2006028
Student Name	Koey Hor Yee
Supervisor Name	Dr Siti Nurlaili Binti Karim

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

____
(Signature of Student)
Date: 10 September 2024