

NLP Based Chatbot For Fitness With Mobile App

Submitted in partial fulfilment of the requirements for the award of
Bachelor of Engineering in Computer Science and Engineering

by

DEVISETTY POOJA (Reg.No- 40110313)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

CATEGORY -1 UNIVERSITY BY UGC

Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHISALAI, CHENNAI - 600119

APRIL - 2024



SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Devisetty Pooja (Reg.No 40110313)** who carried out the project entitled **"NLP BASED CHATBOT FOR FITNESS WITH MOBILE APP"** under my supervision from November 2023 to April 2024.

Internal Guide

DR. R. SETHURAMAN, B.E., M.E., Ph.D.

Head of the Department

DR. L. LAKSHMANAN, B.E., M.E., Ph.D.

Submitted for Viva voice Examination Held on _____

Internal Examiner

External Examiner

DECLARATION

I, **Devisetty Pooja (REG NO. 40110313)** hereby declare that the Project Report entitled “**Nlp based chatbot for Fitness with Mobile App**” done by me under the guidance of **Dr. R. Sethuraman, B.E., M.E., Ph.D.**, is submitted in partial fulfillment of the requirements of the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to The **Board of Management** of **Sathyabama Institute of Science and Technology** for their encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, B.E., M.E., Ph.D., Dean**, School of Computing, **Dr. L. Lakshmanan, B.E., M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude my project guide **Dr. R. Sethuraman, B.E., M.E., Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Chatbots have revolutionized the way humans interact with computer systems and they have substituted the use of service agents, call center representatives etc. Fitness industry has always been a growing industry although it has not adapted to the latest technologies like AI, ML and cloud computing. In this paper, we propose an idea to develop a chatbot for fitness management and integrate it with a web application. We proposed using Natural Language Processing (NLP) and Natural Language Understanding (NLU) provided for the Chatbot Assistant. This software uses a serverless architecture to combine the services of a professional by offering diet plans, home exercises, interactive counseling sessions, fitness recommendations. Natural language processing (NLP) technology has become a tool to make chatbots to understand and respond to natural language. By harnessing the power of NLP technology, chatbots can interpret user questions, extract valuable information, and generate meaningful responses in conversations.

FitBot is an innovative Natural Language Processing (NLP) chatbot designed to provide personalized fitness assistance. In today's digital era, the demand for convenient and accessible fitness solutions is on the rise. FitBot addresses this need by leveraging NLP techniques to understand and respond to user queries related to exercise, nutrition, and wellness. By employing machine learning algorithms, FitBot continuously learns from user interactions to offer tailored workout plans, dietary advice, and motivational support. Through seamless integration with wearable devices and health apps, FitBot tracks users' progress and adapts recommendations accordingly. This abstract outlines the development and functionality of FitBot, highlighting its potential to revolutionize the way individuals engage with their fitness goals through natural language interaction. Performance metrics including accuracy, engagement, satisfaction, and conversation flow are analyzed to assess the chatbot's effectiveness and identify areas for improvement. Through iterative refinement and optimization, the chatbot aims to empower users to make informed decisions, adopt healthy behaviors, and achieve their fitness goals. Overall, the abstract highlights the potential of NLP-based fitness chatbots to enhance user experiences and contribute to positive behavior change in the digital age.

TABLE OF CONTENT

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	ix
1	INTRODUCTION	1
	1.1 Study Area	2
	1.2 Problem Statement and Objective	3
2	LITERATURE SURVEY	4
	2.1 Inferences from Literature Survey	7
	2.2 Existing System and Proposed System	8
	2.3 Open Problems in Existing System	8
3	AIM AND SCOPE OF THE PRESENT INVESTIGATION	11
	3.1 Aim of the Project	11
	3.2 Scope of the present investigation	11
4	EXPERIMENTAL METHODS, ALGORITHMS USED	13
	4.1 Process Flow Diagram	14
	4.2 Architecture / Overall Design Of Proposed System	15
	4.3 Selected Methodology Or Process Model	16
	4.4 Description of Software for Implementation and Testing plan of the Proposed Model/System	16
	4.5 Modules	18
	4.6 Proposed Algorithm	20
	4.7 Scale Of Proposed System	21

4.8	Selected Hardware And Software	25
4.9	Requirement Analysis	27
4.10	Non- Functional Requirements	29
5	IMPLEMENTATION DETAILS	32
6	RESULTS AND DISCUSSIONS, PERFORMANCE ANALYSIS	34
6.1	Results and Discussions	34
6.2	Performance Analysis	35
7	SUMMARY AND CONCLUSION	38
	REFERENCES	41
	APPENDIX	44
A.	SOURCE CODE	44
B.	SCREENSHOTS	54
C.	CERTIFICATE	56

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
4.1	Process flow diagram	14
4.2	System Architecture	15
6.1	User interface	36
6.2	User Output	36
6.3	Login Page	37
6.4	Search Bar	38
6.5	Output	39

LIST OF ABBREVIATIONS

ABBREVIATION

EXPANSION

NLP

NATURAL LANGUAGE PROCESS

AI

ARTIFICIAL INTELLIGENCE

BOT

CHATBOT

CHAPTER 1

INTRODUCTION

Natural Language Processing (NLP) serves as the vital link between the intricacies of computer language and the nuances of human communication, representing the closest approximation computers can achieve in understanding human language. Its integration into healthcare stands as a pivotal advancement with profound benefits for humanity. NLP not only enhances clinical support for healthcare professionals but also empowers individuals with a valuable tool for understanding medical information. In the realm of healthcare, NLP's potential is boundless. It facilitates the efficient analysis of vast volumes of medical texts, research papers, and patient records, enabling healthcare workers to make more informed decisions. This technology aids in diagnosing diseases, predicting outbreaks, and advancing medical research by extracting valuable insights from unstructured data sources. Moreover, NLP democratizes healthcare knowledge by making medical information more accessible to the general public. Laypeople can use AI-powered chatbots and virtual assistants to obtain reliable information about symptoms, treatments, and preventive measures. This not only empowers individuals to take control of their health but also reduces the burden on healthcare providers, allowing them to focus on more complex cases. As AI continues to advance, the time has never been riper to harness this human-made marvel for the betterment of society. With its ability to bridge the gap between computer systems and human language, NLP revolutionizes healthcare, making it more efficient, accessible, and inclusive. By embracing AI in healthcare, we pave the way for a healthier and more informed future for all. Whether you're looking to lose weight, build muscle, improve flexibility, or simply enhance your overall well-being, our Fitness Chatbot is here to guide you every step of the way. From creating customized workout plans to offering nutritional recommendations, it's like having a personal trainer and nutritionist right at your fingertips, available 24/7. accessible to the general public. Through AI-driven chatbots and virtual assistants, individuals can easily access reliable information on symptoms, treatments, and preventive measures, promoting health literacy and fostering a sense of personal responsibility for one's well-being.

1.1 STUDY AREA

Natural Language Processing (NLP) serves as an indispensable bridge between the intricate world of computer language and the nuanced realm of human communication. Its application, particularly in the healthcare sector, is transformative and has far-reaching implications for the well-being of society. NLP not only enhances clinical support for healthcare professionals but also empowers individuals with a powerful tool for understanding complex medical information. Within the healthcare landscape, NLP's potential is virtually boundless. It enables the efficient analysis of extensive medical texts, research papers, and patient records, providing healthcare workers with valuable insights that facilitate more informed decision-making. From diagnosing diseases to predicting disease outbreaks and advancing medical research, NLP is instrumental in extracting valuable insights from unstructured data sources, revolutionising the way healthcare is practiced and evolving our understanding of medical science. Furthermore, NLP democratises healthcare knowledge by making medical information more accessible to the general public. Through AI-driven chatbots and virtual assistants, individuals can easily access reliable information on symptoms, treatments, and preventive measures, promoting health literacy and fostering a sense of personal responsibility for one's well-being. This not only empowers individuals to take control of their health but also lightens the load on healthcare providers, allowing them to concentrate on more complex cases and delivering higher-quality care. As artificial intelligence continues to advance, the time is ripe to harness this remarkable human-made technology for the betterment of society. NLP's ability to bridge the gap between computer systems and human language is a cornerstone of progress in healthcare, making it more efficient, accessible, and inclusive. By embracing AI in healthcare, we pave the way for a healthier, more informed, and equitable future for all. This not only empowers individuals to take control of their health but also reduces the burden on healthcare providers, allowing them to focus on more complex cases. When a firm relies exclusively on human power, for example, it can only service a certain number of people at any given time. Human-powered firms are compelled to rely on standardized models to be cost-effective, and their proactive and personalized outreach skills are restricted.

1.2 PROBLEM STATEMENT AND OBJECTIVE

The user can do the important advantage of a chatbot only it can diagnose all quite fitness and supply necessary information. text-to-text diagnosis bot connects users about their fitness issues and gives a customized diagnosis to support their problems. Hence, people will have a thought about their health and have the proper protection. Businesses benefit from chatbots because they increase operational efficiency and save costs while providing convenience and extra services to internal staff and external clients. They enable businesses to quickly answer a variety of client concerns and enquiries while decreasing the requirement for human involvement. A firm can expand, customize, and be proactive all at the same time using chatbots, which is a key differentiation. When a firm relies exclusively on human power, for example, it can only service a certain number of people at any given time. Human-powered firms are compelled to rely on standardized models to be cost-effective, and their proactive and personalized outreach skills are restricted. Chatbots, on the other hand, allow businesses to interact with an endless number of consumers in a personalized manner and can be scaled up or down based on demand and business needs. A company may give humanlike, tailored, proactive service to millions of customers at the same time by deploying chatbots. Our focus in this paper is to implement an interactive chatbot for personal fitness management. To assist users in yoga plan, diet plan, scheduling, users can use this interactive and advanced chatbot. The origins of the chatbot may be traced back to Alan Turing's idea of sentient robots in the 1950s. Since then, artificial intelligence, which is the foundation for chatbots, has advanced to encompass super intelligent supercomputers like IBM Watson. Chatbots are widely used to improve the IT service management experience, which focuses on self-service and automating internal operations. Common activities such as password changes, system status, outage. The proposed system suggests adapting and expanding existing testing concepts to address the unique characteristics of NLP-intensive bots. It also advocates for the consideration of security and privacy testing, Healthcare plays a wide role in our daily lives, whenever a person feels sick he/she visits their family doctor or any nearby clinic just to get to know what issues they are facing, in the recent years many companies & institutions have collaborated.

CHAPTER 2

LITERATURE SURVEY

A Chatbot for Medical Purpose using Deep Learning proposed by Vaibha Tode in Himanshu Gadge , Sudarshan Madane , Prateek Kachare, Anuradha Deokar / 2021. The aim of our Project is to design a Conversational AI Powered Chatbot for Medical Diagnostics using Deep Learning which mainly focuses on rural parts as well as poor and needy people of our country. Our System has the capability to understand the symptoms of the patient and communicates with Patient (End-user) through web-UI. Our system tries to solve their problem with the help of the symptoms provided by Patient itself and help them to give the correct antibiotics/ medicines and precautions. NLTK (Natural Language Toolkit) is a module/ program in python which can able to perform symbolic and statistical Natural Language Processing for English written in Programming. It is used to analyze the input in the form of speech and generate responses that are similar to humans. Its ability to communicate was limited, its knowledge is limited, it cannot keep long conversations and cannot learn or discover context from the discussion.

Contextual Chatbot for Healthcare Purposes (using Deep Learning) proposed Prathamesh Kandpal, Kapil Jasnani, Ritesh Raut , Prof. Dr. Siddharth Bhorge in / 2020. In this paper we will be not only discussing about the working of our model but also the applications and relevant work conducted in this domain, also there will be discussion about the challenges and future scope of this technology. For this work, neural networks have been used to train data and various packages which help us in giving better results. In this chatbot we will be integrating the concepts of Natural Language Processing with Deep Learning for getting better results. Healthcare plays a wide role in our daily lives, whenever a person feels sick he/she visits their family doctor or any nearby clinic just to get to know what issues they are facing, in the recent years many companies & institutions have collaborated. Chatbots can play a major role in reshaping the healthcare industry by providing either predictive diagnosis or any other assistantships like booking an appointment. Chatbots are still in the early phases of development and they are

heavily relying on artificial intelligence and machine learning so there are still some limitations or challenges they face in recent years.

Healthcare Chatbot using Natural Language Processing proposed by Papiya Mahajan, Rinku Wankhade, Anup Jawade, Pragati Dange, Aishwarya Bhoge/ 2020. The system provides text or voice assistance, that means user can use his own convenient language, Bot will provides which type of disease based on the user symptoms.

Thus, people will have an idea about their health and have the right protection.. Chatbots are programs that work on Machine Learning (ML) as well as Artificial Intelligence (AI) Natural Language Processing (NLP) techniques such as NLTK for Python can be applied to analyses speech, and intelligent responses can be found by designing an engine to provide appropriate human like responses

It takes more time to response to the user question. Pay some charges to perform live chat. Users can describe their symptoms to the chatbot, which can then provide possible diagnoses or recommend appropriate actions (e.g., see a doctor, take specific medications, rest).The chatbot can help users schedule appointments with healthcare providers based on availability, location, and type of service needed. Users can ask the chatbot questions about their medications, including dosage, side effects, and interactions with other drugs.For patients with chronic conditions, the chatbot can collect and analyze data from wearable devices or other sources to track health metrics (e.g., blood pressure, blood sugar levels) and provide insights or alerts to both patients and healthcare providers.The chatbot can offer support for mental health issues by providing resources, coping strategies, and referrals to mental health professionals when needed.

ChatBots in Education proposed by Sefio Roos /2018, Gives personalized mentor experience and ans all relevant queries related to the student education domain. Chatbots can offer personalized learning experiences by assessing students' strengths, weaknesses, and learning styles. They can provide customized study plans, recommend resources, and offer interactive learning activities tailored to individual needs.Chatbots can provide round-the-clock

regular classroom hours. This can be particularly helpful for students studying independently or those who need help with homework or assignments at unconventional times. Chatbots can assist in assessing students' understanding of concepts through quizzes, interactive exercises, and simulations. They can provide instant feedback on students' responses, helping them identify areas for improvement and reinforcing learning. Chatbots can facilitate language learning by engaging students in conversation,

practicing vocabulary and grammar, and providing real-time feedback on pronunciation and language usage. Chatbots can assist with administrative tasks such as scheduling appointments, providing reminders for deadlines, and offering information about course registration, campus facilities, and academic policies.

Chatbot Evaluation and Database Expansion via Crowdsourcing proposed by Zhou Yu, Ziyu Xu, Alan W Black, Alexander I. Rudnicky /2016. As a dataset is created of appropriate responses to related queries, considerable level of accuracy is achieved. Define clear evaluation metrics for the chatbot's performance. These metrics may include accuracy in providing information, response time, user satisfaction, and task completion rate. Utilize crowdsourcing platforms to recruit a diverse group of individuals to interact with the chatbot and provide feedback. These individuals can simulate real-world usage scenarios and provide valuable insights into the chatbot's strengths and weaknesses. Crowd workers can help annotate chatbot responses, identifying correct and incorrect answers. They can also provide corrections or additional information where necessary, helping to improve the accuracy and completeness of the chatbot's knowledge base. Crowd workers can contribute to expanding the chatbot's database by providing new questions, scenarios, and responses. They can submit queries that the chatbot currently cannot handle and provide relevant information to enrich its knowledge base. Implement mechanisms to ensure the quality of crowd contributions. This may include providing guidelines for annotating responses, using multiple workers to validate answers, and incorporating feedback loops for continuous improvement. A lot of overhead work has to be done for maintaining database

2.1 INFERENCES FROM LITERATURE SURVEY

The paper titled "Testing Challenges for NLP-Intensive Bots" addresses the burgeoning field of testing NLP-intensive bots, particularly chatbots and voice bots, which have gained significant popularity due to their interactions with users through natural language processing (NLP). The authors underscore the importance of robust testing for these AI-driven conversational agents, as malfunctions could have adverse effects on a company's reputation and financial performance. The acknowledgement of the testing of NLP-intensive bots is still in its infancy and aim to provide a reference framework for evaluating and comparing future research initiatives in bot testing.

The literature survey within the paper delves into key aspects of testing NLP-intensive bots. It highlights the broader challenges of testing AI systems, emphasising the complexities introduced by NLP. The authors define NLP-intensive bots and stress the need to assess not only the NLP aspect but also the underlying software, often implemented as a finite-state machine, as well as the interactions between these components.

The paper references existing research and tools for testing chatbots and NLP-based conversational agents, such as Barium, Zypnos, Chatbottest, QBox, and Dashbot, which offer varying levels of testing support. It discusses various testing techniques applicable to NLP-intensive bots, including input domain-based techniques, code-based techniques, and fault-based techniques. Additionally, it highlights the importance of measuring both the program under test and the effectiveness of the tests performed.

The authors briefly mention the significance of security and privacy testing for NLP bots, considering their exposure to sensitive data and potential security threats. They also touch upon usability and human-computer interaction testing, emphasizing the need to evaluate user interactions and experience metric. Manages errors gracefully and provides helpful prompts or suggestions in case of misunderstandings or failures. Delivers the chatbot's response to the user through the appropriate channel (text, voice, etc.). The proposed system suggests adapting and expanding existing testing concepts to address the unique characteristics of NLP-intensive bots. It also advocates for the consideration of security and privacy testing.

2.2 EXISTING SYSTEM AND PROPOSED SYSTEM

Existing System: The paper discusses the current state of testing NLP-intensive bots, emphasizing that this field is still in its early stages. It recognizes the growing popularity of chatbots and voice bots, driven by their ability to interact with users using Natural Language Processing (NLP). The authors point out that companies are increasingly concerned about the quality and reliability of these bots, as malfunctions can have significant impacts on a company's reputation and revenue. Existing testing tools and techniques for these bots are limited, and the paper suggests that a more comprehensive testing framework is needed.

Proposed System: In response to the challenges in testing NLP-intensive bots, the paper proposes the development of a reference framework. This framework aims to provide a structured approach to testing these bots, considering not only the NLP component but also the underlying software architecture, often implemented as finite-state machines (FSMs), and the interactions between these components. The proposed system suggests adapting and expanding existing testing concepts to address the unique characteristics of NLP-intensive bots. It also advocates for the consideration of security and privacy testing, usability, and human-computer interaction testing to ensure comprehensive evaluation. The goal is to create a unified reference framework that can serve as a foundation for future research initiatives and comparisons in the field of bot testing.

2.3 OPEN PROBLEMS IN EXISTING SYSTEM

Limited Testing Tools: One of the primary challenges in the existing system is the lack of specialized testing tools and frameworks designed specifically for NLP-intensive bots. Many of the existing tools are limited in their capabilities and are not tailored to address the unique characteristics of these bots.

Complexity of NLP: NLP itself poses significant challenges, including natural language ambiguity, varying user inputs, and the need for extensive training data. Testing NLP-intensive bots requires addressing these complexities effectively.

Interaction Complexity: NLP bots often interact with users through dynamic conversations. Testing the dynamic nature of these interactions and ensuring that the bot responds appropriately is a non-trivial problem.

Security and Privacy: Ensuring the security and privacy of user data is crucial, especially when bots handle sensitive information.

1. A Chatbot for Medical Purpose using Deep Learning

The Chatbot is a software programs that is used to interact with clients using natural language Processing via text or text to speech format. Today in the present era, the major challenges that India as a country is facing is to cater good quality and affordable healthcare services to its growing population and at the same time, they are not cost efficient. Nowadays, it is becoming very difficult to provide healthcare facilities as we have seen in COVID-19 critical situations that the condition in India was getting worse because of lack of transportation, availability of doctors and hospitality. Sometimes it causes the people to postpone their treatment as well as there is an increment in death count. The aim of our Project is to design a Conversational AI Powered Chatbot for Medical Diagnostics using Deep Learning which mainly focuses on rural parts as well as poor and needy people of our country. Our System has the capability to understand the symptoms of the patient and communicates with Patient (End- user) through web-UI. Our system tries to solve their problem with the help of the symptoms provided by patient itself and help them to give the correct antibiotics/ medicines and precautions. NLTK (Natural Language Toolkit) is a module/ program in python which can able to perform symbolic and statistical Natural Language Processing for English written in Programming. It is used to analyses the input in the form of speech and generate responses that are similar to humans.

2.Contextual Chatbot for Healthcare Purposes (using Deep Learning)

As the demand in Machine Learning & AI keeps growing, new technologies will keep coming in the market which will impact our day-to-day activities, and one such technology is Virtual Assistant Bots or simply Chatbots. Chatbots have evolved from being Menu/Button based, to Keywords based and now Contextual based. The most advanced among all of the above is contextual based because it uses Machine Learning and Artificial Intelligence techniques to store and process the training models which help the chatbot to give better and appropriate response when user asks domain specific questions to the bot. In this paper we will be not only discussing about the working of our model but also the

work conducted in this domain, also there will be discussion about the challenges and future scope of this technology. For this work, neural networks have been used to train data and various packages which help us in giving better results. In this chatbot we will be integrating the concepts of Natural Language Processing with Deep Learning for getting better results. Healthcare plays a wide role in our daily lives, whenever a person feels sick, he/she visits their family doctor or any nearby clinic just to get to know what issues they are facing, can help doctors and medical staff to deal with patients in better way and reduce their efforts with the help of technology. Chatbots can play a major role in reshaping the healthcare industry by providing either predictive diagnosis or any other assistantships like booking an appointment.

3. Healthcare Chatbot using Natural Language Processing

To start a good life healthcare is very important. But it is very difficult to the consult the doctor if any health issues. The proposed idea is to create a healthcare chatbot using Natural Language Processing technique it is the part of Artificial Intelligence that can diagnose the disease and provide basic. To reduce the healthcare costs and improve accessibility to medical knowledge the Healthcare chatbot is built. Some chatbots acts as a medical reference book, which helps the patient know more about their disease and helps to improve their health. The user can achieve the benefit of a healthcare chatbot only when it can diagnose all kind of disease and provide necessary information. The system provides text or voice assistance, that means user can use his own convenient language, Bot will provide which type of disease based on the user symptoms, and provides doctor and also provides food suggestion that means which type of food you have to take. Thus, people will have an idea about their health and have the right protection. Chatbots are programs that work on Machine Learning (ML) as well as Artificial Intelligence (AI) Natural Language Processing (NLP) techniques such as NLTK for Python can be applied to analyses speech, and intelligent responses can be found by designing an engine to provide appropriate human like responses.

CHAPTER 3

AIM AND SCOPE OF THE PRESENT INVESTIGATION

3.1 AIM

The aim of an NLP-based fitness chatbot with a mobile application could be to provide personalized guidance, support, and motivation to users in achieving their fitness goals. Here are some key objectives and features such a chatbot could have:

- Personalized Fitness Plans:** The chatbot can interact with users to understand their fitness goals, current fitness level, dietary preferences, and any health restrictions. Based on this information, it can generate personalized fitness plans tailored to each user's needs.
- Workout Recommendations:** The chatbot can provide users with workout recommendations based on their goals, preferences, and available equipment or facilities. It can suggest specific exercises, workout routines, and training schedules.
- Nutritional Guidance:** The chatbot can offer nutritional guidance by recommending healthy meal options, suggesting calorie intake goals, and providing information on macronutrients and micronutrients.

3.2 SCOPE OF THE PRESENT INVESTIGATION

The scope of the present investigation of an NLP-based chatbot for fitness with a mobile application could encompass several key aspects:

User Needs and Requirements Analysis: Investigate the needs, preferences, and pain points of potential users regarding fitness and health-related guidance. Understand the features and functionalities users expect from a fitness chatbot and mobile application.

Existing Solutions Review: Analyze existing NLP-based fitness chatbots and mobile applications to identify their strengths, weaknesses, and gaps. Determine what aspects of these solutions can be improved or innovated upon.

Technology Assessment: Evaluate the state-of-the-art NLP techniques, machine learning algorithms, and natural language understanding models.

Data Collection and Annotation: Gather relevant data such as fitness articles, workout routines, nutritional information, and user interaction logs. Annotate the data to train and fine-tune NLP models for understanding user queries and generating

appropriate responses. The chatbot can provide personalized fitness recommendations based on the user's goals, preferences, and current fitness level. This may include workout routines, exercise suggestions, and dietary advice tailored to the individual's needs. Users can engage in interactive workout sessions with the chatbot, receiving real-time feedback, encouragement, and modifications based on their performance and feedback.

The chatbot can offer nutritional guidance, including meal planning, calorie tracking, and dietary recommendations tailored to the user's fitness objectives and dietary restrictions. Users can track their fitness progress over time through the chatbot, recording workout sessions, monitoring achievements, and receiving insights into their performance trends. The chatbot can provide motivational messages, reminders, and encouragement to help users stay motivated and committed to their fitness journey. Users can access educational resources such as articles, videos, and tips on various fitness-related topics, empowering them with knowledge to make informed decisions about their health and wellness. The chatbot can integrate with wearable fitness devices and mobile apps to sync data, such as step counts, heart rate, and sleep patterns, providing a holistic view of the user's health and fitness metrics.

Users can connect with a community of like-minded individuals through the chatbot, sharing experiences, challenges, and success stories, fostering a supportive environment for achieving fitness goals. The Chatbot continuously learns from user interactions, gathering feedback to improve its effectiveness in providing personalized support and enhancing the overall user experience. By encompassing these features within its scope, the NLP-based fitness chatbot aims to revolutionize the way individuals approach and engage with their fitness journey, offering convenient, personalized, and effective support to help them lead healthier and more active lives.

Nlp based fitness chatbot aims to revolutionized. It's a branch of AI that focuss on interaction between computers and humans using natural language. Nlp enables computers to understand, interpret, and generate human language in a way that is meaningful and useful. This field encompasses a wide range of tasks, including but not limited to text analysis.

CHAPTER 4

EXPERIMENTAL METHODS, ALGORITHMS USED

The proposed system for NLP-intensive bot testing is a comprehensive framework designed to tackle the unique challenges posed by bots heavily reliant on Natural Language Processing (NLP). It encompasses several crucial aspects, beginning with a robust testing framework that includes unit testing, integration testing, and system testing, ensuring comprehensive bot evaluation. Intent-based testing forms a core component, enabling the creation of test cases based on expected intent recognition and parameter accuracy. This system's integration with Finite State Machines (FSMs) facilitates testing the interplay between NLP components and FSM-driven bot behaviors, guaranteeing adherence to predefined conversation paths. It also encompasses performance testing to evaluate the bot's responsiveness under load, security and privacy testing to identify vulnerabilities, and usability evaluation to assess user-friendliness and platform-specific UI utilization.

Quality metrics and test oracles help measure testing effectiveness, while scalability and extensibility allow adaptation to evolving bot development landscapes. In summary, the proposed system empowers thorough testing, enhances security, and ensures usability for NLP-intensive bots while accommodating scalability and future expansion. The chatbot accesses relevant information from knowledge bases, databases, or external APIs. Organizes retrieved information into a structured format for easy access and retrieval. Represents knowledge in a hierarchical or graph-based structure to capture relationships between entities. Determines the user's intent based on the input text. Maintains context across multiple interactions to provide coherent and relevant responses. Each token is tagged with its part of speech (noun, verb, adjective, etc.), aiding in grammatical analysis. Identifies and classifies named entities (such as people, organizations, locations) in the text. Resolving ambiguous references to entities based on context

4.1 PROCESS FLOW DIAGRAM

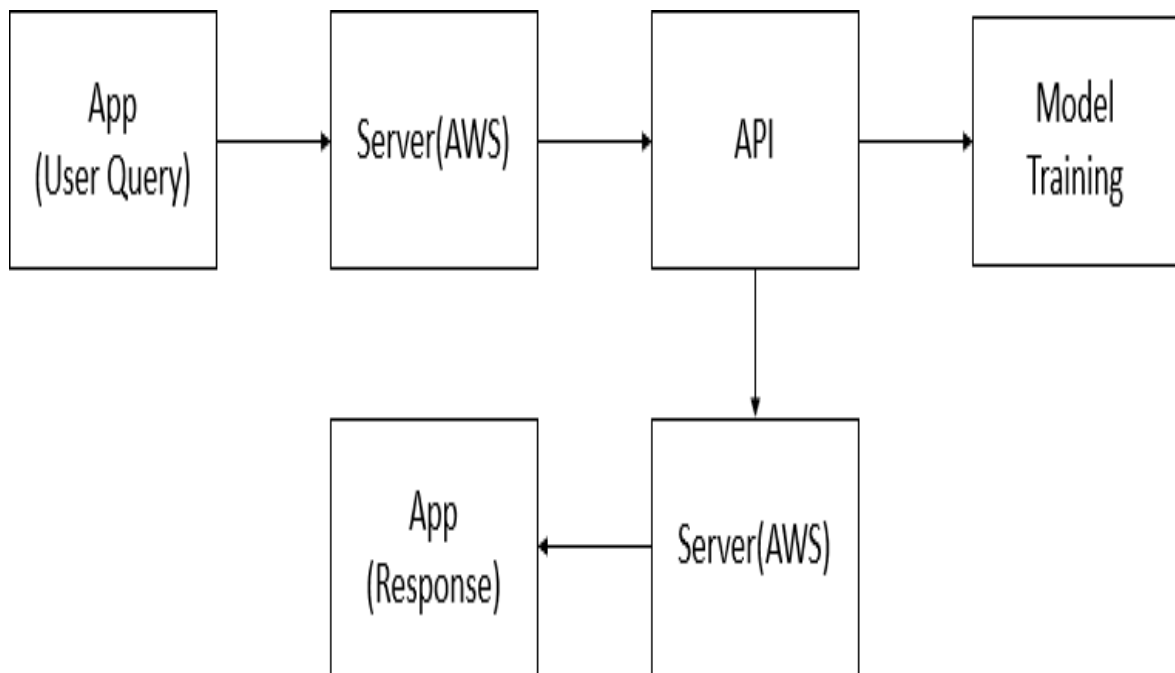


Fig 4.1 : Process Flow Diagram

The input text is broken down into individual tokens (words or phrases) for analysis. Commonly occurring words (stopwords) that do not contribute much to the meaning of the sentence are removed. Each token is tagged with its part of speech (noun, verb, adjective, etc.), aiding in grammatical analysis. Identifies and classifies named entities (such as people, organizations, locations) in the text. Resolving ambiguous references to entities based on context. The chatbot accesses relevant information from knowledge bases, databases, or external APIs. Organizes retrieved information into a structured format for easy access and retrieval. Represents knowledge in a hierarchical or graph-based structure to capture relationships between entities. Determines the user's intent based on the input text. Maintains context across multiple interactions to provide coherent and relevant responses.

It takes the user inputs and send to the servers of Aws check the query related answer in the programed files. API is application interface where the query answer is displayed through internal App response.

4.2 ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM

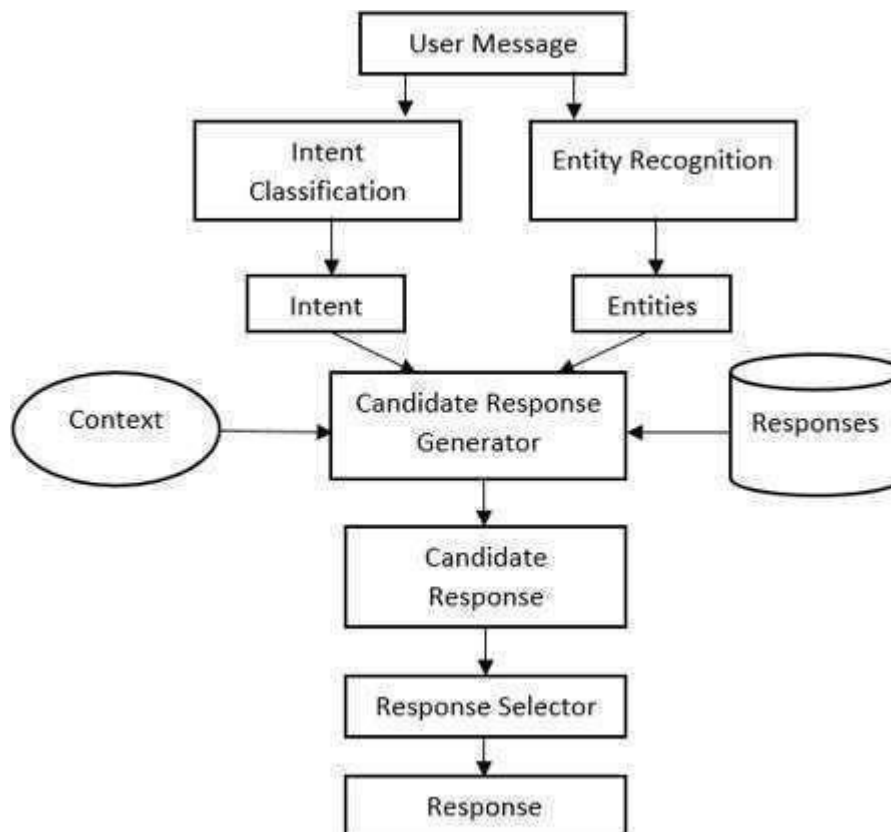


Fig 4.2 : System Architecture

This architecture outlines the components and flow of a NLP-based fitness chatbot system, emphasizing the processing of user inputs, management of dialog, access to knowledge bases, generation of responses, integration with external services, learning and adaptation, as well as analytics and monitoring for continuous improvement and maintenance. Contains a comprehensive list of exercises categorized by muscle group, equipment needed, difficulty level, etc. Stores data related to nutritional content of foods, meal plans, and dietary guidelines. Includes information on achieving various fitness goals (weight loss, muscle gain, endurance improvement) and personalized recommendations based on user profiles and preferences. Maintains context across multiple interactions to provide coherent and relevant responses. Tracks the current state of the conversation and manages the flow of interaction. Generates appropriate responses based on the user's intent, context, and available knowledge.

4.3 SELECTED METHODOLOGY OR PROCESS MODEL

We aim to develop the chatbot for healthcare. As we found that there are very limited resources which specifically target on the healthcare, we decided to make our own dataset to train our health bot. Collecting the data(Questions and answers) to form up the data will require huge amount of time. So we are going with a new method where we will train the data from the questions of the users. So our plan is to use a third party API for the initial days where the responses to the users queries will be stored and used to train our own health bot. Once we get adequate data then we will be shifting to our own trained bot

4.4 DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLAN OF THE PROPOSED MODEL/SYSTEM

The software for implementing and testing the proposed model/system is a comprehensive suite designed to facilitate the development, testing, and evaluation of NLP-intensive bots. It includes a range of tools and components to support various stages of bot creation and validation.

Bot Development Environment: This software provides a user-friendly Integrated Development Environment (IDE) tailored for bot development. It offers code editors, debugging tools, and visualization features to streamline the bot creation process. **Testing Framework:** The testing framework within the software enables developers to design and execute different types of tests, including unit tests, integration tests, system tests, and performance tests. It offers automated testing capabilities to ensure the reliability and accuracy of bot responses.

Intent-Based Testing: The system incorporates intent-based testing, allowing developers to define test cases based on expected user intents and parameter accuracy. This feature aids in validating the core NLP functionality of the bot.

FSM Integration: To assess the interaction between NLP components and FSM-driven bot behaviors, the software integrates FSM modelling and testing capabilities. It helps ensure that bot conversations follow predefined paths and transitions.

Security and Privacy Testing: The software includes tools for conducting security assessments and privacy checks. It identifies vulnerabilities, guards against

potential data breaches, and ensures that the bot complies with privacy regulations. Usability Evaluation: Usability testing components assess the user-friendliness of the bot. It measures metrics such as conversation length, abandonment rates, and Quality Metrics and Test Oracles: The system provides mechanisms to define quality metrics for bot evaluation and establish test oracles for determining the correctness of bot responses.

Scalability and Extensibility: The software is designed to be scalable and extensible, accommodating future updates, and evolving bot development requirements.

Testing Plan: The testing plan outlines the testing process, including test objectives, test cases, test data, and success criteria. It guides the testing efforts to ensure thorough coverage and validation.

Bot Development Environment:

This software provides a user-friendly Integrated Development Environment (IDE) tailored for bot development. It offers code editors, debugging tools, and visualization features to streamline the bot creation process.

Testing Framework: The testing framework within the software enables developers to design and execute different types of tests, including unit tests, integration tests, system tests, and performance tests. It offers automated testing capabilities to ensure the reliability and accuracy of bot responses.

Intent-Based Testing:

The system incorporates intent-based testing, allowing developers to define test cases based on expected user intents and parameter accuracy. This feature aids in validating the core NLP functionality of the bot.

FSM Integration:

To assess the interaction between NLP components and FSM-driven bot behaviors, the software integrates FSM modelling and testing capabilities. It helps ensure that bot conversations follow predefined paths and transitions

Security and Privacy Testing:

The software includes tools for conducting security assessments and privacy checks. It identifies vulnerabilities, guards against potential data breaches, and ensures that the bot.

4.5 MODULES

- *User Validation and Extraction of Problems*
- *Extracted issues With Trained Datasets*
- *Specifying The fitness solution*

MODULE DESCRIPTION**1. User Validation and Extraction of Problems**

The validation of the user details occurs here. Then problems are extracted using String Searching Algorithm where substring representing the problems is identified in the natural language text input. When users give directly the issues details such as the system will identify it. But however, the system should also be able to handle input like, “When I read, I’m okay at first, but over time, I lost my weight and am getting tired.” In this case, the system should extract words like “weight loss” and “tired” (and not substrings like “read” or “okay”).

2. Extracted issues With Trained Datasets

Given some substring from the user’s input, we generate a list of suggested closest problems. We then ask the user to confirm if they have any of the problems from the suggested ones. Based on their reply few solutions are being shortlisted. Then further fitness problems clarification and fitness suggestions are being done by asking the users a series of questions and the mapping of the issues to the exact fitness solution is done.

Specifying The fitness solution

This process carries the list of fitness solutions in the database and each problem being entered is compared to the problems of the common solutions. The next problem is checked until a matching problem is found. They are shortlisted based

on the end users input on the question evaluation. The accurate fitness solution is identified and specified

to the endorser by the chatbot. The chatbot checks whether the identified fitness solution is a major issue or minor issue based on the conditions built in the chatbot.

Stress Testing

Stress testing is the testing to evaluate how system behaves under unfavorable conditions. Testing is conducted at beyond limits of the specifications. It falls under the class of black box testing.

Performance Testing

Performance testing is the testing to assess the speed and effectiveness of the system and to make sure it is generating results within a specified time as in performance requirements. It falls under the class of black box testing.

Usability Testing

Usability testing is performed to the perspective of the client, to evaluate how the GUI is user-friendly? How easily can the client learn? After learning how to use, how proficiently can the client perform? How pleasing is it to use its design? This falls under the class of black box testing.

Acceptance Testing

Acceptance testing is often done by the customer to ensure that the delivered product meets the requirements and works as the customer expected. It falls under the class of black box testing.

Regression Testing

Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is working correctly and is not damaging or imposing other modules to produce unexpected results. It falls under the class of black box testing.

4.6 PROPOSED ALGORITHM:

Step-1: Data Collection and Preprocessing:

Gather a dataset of fitness-related text data, including user queries and responses. Preprocess the text data, including tokenization, removing stop words, and stemming/lemmatization.

Step-2: Intent Recognition:

Utilize algorithms like Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), or Transformers to identify the user's intent in their queries. This might include intents like "Get workout suggestions," "Nutrition advice," or "Track my progress."

Step-3: Entity Recognition:

Extract relevant entities from the user's input, such as exercise types, dietary preferences, or fitness goals. Named Entity Recognition (NER) models can be used for this.

Step-4: Response Generation:

Based on the user's intent and extracted entities, generate appropriate responses. You can use techniques like rule-based systems, template-based responses, or more advanced methods like sequence-to-sequence models with attention mechanisms.

Step-5: Personalization:

If possible, personalize responses based on user profiles and historical interactions. This can be achieved by using collaborative filtering or content-based recommendation algorithms.

Step-6: Language Understanding:

Implement a sentiment analysis component to gauge the sentiment of the user's input. This can help tailor responses to the user's emotional state.

Step-7: Dialogue Management:

Implement a dialogue management system to maintain context across multiple user interactions. This can be done using rule-based approaches or reinforcement learning for more advanced systems.

Step-8: Feedback Loop:

Allow users to provide feedback on the bot's responses and use this feedback to improve the model over time.

Step-9: Deployment:

Deploy the Chabot on your chosen platform, whether it's a website, mobile app, or messaging platform. Ensure it's accessible to users.

Step-10: Continuous Training:

Regularly update and retrain your NLP models with new data to keep the Chabot's responses relevant and accurate.

4.7 SCALE OF PROPOSED SYSTEM

The proposed idea is to make a medical Chabot using AI which will diagnose the disease and supply basic details about the disease before consulting a doctor. to scale back the healthcare costs and improve accessibility to medical knowledge the medical chatbot is made. Few Chabot's acts as reference books, which helps the patient find out about the illness and assists with improving their wellbeing. The user can do the important advantage of a chatbot only it can diagnose all quite disease and supply necessary information. Text-to-text diagnosis bot connects patients about their medical issues and gives a customized diagnosis to support their symptoms. Hence, people will have a thought about their health and have the proper protection.

Test plan Software testing is the process of evaluation a software item to detect differences between given input and expected output. Also, to assess the feature of a software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words, software testing is a verification and validation process.

Verification

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

Validation

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure.

Black box Testing

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called

Design the system architecture to scale seamlessly as the user base and feature set grow over time. Considerations include Horizontal scalability to handle increased user traffic and concurrent interactions Load balancing mechanisms to distribute workload efficiently across servers Database scalability to accommodate growing data volumes and user profiles Determine the expected performance metrics and response times for the chatbot. This may include Response time for user queries and interactions Throughput capacity to handle simultaneous user requests Availability and uptime requirements to ensure uninterrupted service

1. User Base:

- Determine the expected number of users who will interact with the chatbot on a regular basis. This could range from a small group of fitness enthusiasts to a large user base spanning across multiple demographics.

2. Features and Functionalities:

- Define the scope of features and functionalities the chatbot will offer. This may include:
 - Exercise recommendations and workout plans
 - Nutritional advice and meal planning
 - Progress tracking and goal setting
 - Integration with fitness wearable's and tracking devices
 - Personalized coaching and feedback
 - Social integration for sharing achievements and challenges

3. Content and Knowledge Base:

- Establish the extent of the fitness-related content and knowledge base that the chatbot will have access to. This could involve:
 - Curating a database of exercises, workouts, and nutrition information
 - Incorporating a wide range of fitness topics and expertise

Integrating with external APIs or databases for real-time data updates and recommendations

4. Scalability:

Design the system architecture to scale seamlessly as the user base and feature set grow over time. Considerations include:

Horizontal scalability to handle increased user traffic and concurrent interactions

Load balancing mechanisms to distribute workload efficiently across servers

Database scalability to accommodate growing data volumes and user profiles

5. Performance Requirements:

Determine the expected performance metrics and response times for the chatbot. This may include:

Response time for user queries and interactions

Throughput capacity to handle simultaneous user requests

Availability and uptime requirements to ensure uninterrupted service

6. Integration and Interoperability:

Plan for integration with external systems, APIs, or platforms to enhance functionality and interoperability. This could involve:

Integrating with fitness tracking devices, nutrition databases, or workout apps

Seamless integration with messaging platforms, social media channels, or web interfaces

Compatibility with existing IT infrastructure and systems within the organization

7. Security and Compliance:

Implement robust security measures and ensure compliance with data protection regulations. Considerations include:

Data encryption for secure transmission and storage of user information

Access control mechanisms to protect sensitive data and user privacy

8 Continuous Improvement and Feedback Loop:

- Establish processes for collecting user feedback, monitoring system performance, and iterating on features and functionalities to improve the chatbot over time. This could involve:

- Analytics and reporting tools to track user engagement and usage patterns
- Feedback mechanisms for users to provide input and suggestions for enhancements

- Agile development methodologies for iterative development and deployment of updates.

9. User Experience and Usability:

- Prioritize user experience and usability to ensure that the chatbot is intuitive, easy to use, and provides value to users. Considerations include:

- User interface design that is visually appealing and accessible across devices

- Natural language processing capabilities for understanding user intent and providing relevant responses

- Personalization features to tailor recommendations and interactions to individual user preferences

10. Training and Support:

- Provide comprehensive training and support resources for users to learn how to interact with the chatbot effectively. This could involve:

- Onboarding tutorials and guides for new users

Help documentation and FAQs to address common queries and issues

- Customer support channels for resolving technical issues and providing assistance as needed

- By considering these factors, you can determine the appropriate scale for the proposed NLP-based fitness chatbot system to meet the needs of users and stakeholders effectively. **Data Collection:** Gather a diverse dataset of user queries and corresponding responses relevant to the fitness domain. Include a variety of topics such as workout routines, nutrition advice, goal setting, and motivational support.

- Data Preprocessing:** Clean and preprocess the collected data to remove noise, standardize text formats, and handle inconsistencies. This may involve tasks

4.7 SELECTED HARDWARE AND SOFTWARE

We aim to develop the Chabot for healthcare. As we found that there are very limited resources which specifically target on the healthcare, we decided to make our own dataset to train our health bot. collecting the data (Questions and answers) to form up the data will require huge amount of time. So, we are going with a new method where we will train the data from the questions of the users.

SOFTWARE REQUIREMENT SPECIFICATION SYSTEM REQUIREMENTS

HARDWARE REQUIREMENTS

System : Pentium i3

ProcessorHard Disk 500
GB.

Monitor : 15" LED

Input Devices : Keyboard, MouseRam : 2 GB

Hardware Configuration

The hardware configuration section defines the hardware components and infrastructure required to support the development and execution of the NLP-intensive bot testing project. It includes details on the server environment, computational resources, and any specialized hardware needs. Here's what this section might include:

Server Environment

Server Infrastructure: Describe the server infrastructure needed for hosting the testing environment. Specify the number of servers, their specifications, and their roles in the testing process.

Operating System: Specify the required operating system(s) for the servers, such as Linux, Windows, or others.

Server Hardware Specifications: Detail the hardware specifications of the servers, including CPU, RAM, storage capacity, and network connectivity.

Scalability: Discuss the ability to scale the server environment to accommodate growing testing needs. Consider load balancing and redundancy for high availability.

Computational Resources

Processing Power: Specify the minimum CPU requirements for the servers, especially if complex NLP models and simulations are involved. Consider multi-core processors for parallel processing.

Memory (RAM): Define the minimum RAM requirements to handle large datasets and NLP processing tasks efficiently.

Storage: Detail the storage requirements, including both local and network-attached storage (NAS). Discuss the capacity needed for storing datasets, test cases, and logs.

GPU Acceleration (Optional): If GPU acceleration is beneficial for NLP tasks (e.g., deep learning), specify the GPU requirements and compatibility.

Network Infrastructure

Network Topology: Describe the network topology required for communication between servers, clients, and external systems. Discuss factors like network latency and bandwidth requirements.

Firewall and Security: Address network security concerns, including firewall configurations and access control lists (ACLs) to protect sensitive data.

Specialized Hardware (if applicable)

Speech Recognition Hardware: If the project involves voice bots, specify any specialized hardware needed for speech recognition and audio processing.

IoT Devices (if applicable): If the bots interact with Internet of Things (IoT) devices for testing, mention the hardware and connectivity requirements for these devices.

Backup and Redundancy

Backup Strategy: Outline the backup strategy for data and configurations to ensure data integrity and disaster recovery.

Redundancy: Discuss redundancy measures to minimize downtime in case of hardware failures and external systems. Discuss factors like network latency and bandwidth requirements.

System Requirements

Software Compatibility: Ensure that the hardware components are compatible with the required software components, including operating systems, testing frameworks, and NLP libraries.

Environmental Considerations: Mention any specific environmental conditions required for hardware stability, such as temperature and humidity control in server rooms.

The hardware configuration section of the SRS document should provide a clear understanding of the infrastructure needed to support the NLP-intensive bot testing project. It helps project stakeholders, including developers, IT administrators, and management, make informed decisions about hardware procurement and setup to ensure the successful execution of the project.

4.8 REQUIREMENT ANALYSIS

Requirement analysis, also called requirement engineering, is the process of determining user expectations for a new modified product. It encompasses the tasks that determine the need for analyzing, documenting, validating, and managing software or system requirements. The requirements should be documentable, actionable, measurable, testable, and traceable related to identified business needs or opportunities and define to a level of detail, sufficient for system design.

FUNCTIONAL REQUIREMENTS

It is a technical specification requirement for the software products. It is the first step in the requirement analysis process which lists the requirements of software systems including functional, performance and security requirements. The function of the system depends mainly on the quality hardware used to run the software with given functionality.

Usability

It specifies how easy the system must be use. It is easy to ask queries in any format which is short or long, porter stemming algorithm stimulates the desired response for user.

Robustness

It refers to a program that performs well not only under ordinary conditions but also under unusual conditions. It is the ability of the user to cope with errors for irrelevant queries during execution.

Security

The state of providing protected access to resource is security. The system provides good security and unauthorized users cannot access the system there by providing high security.

Reliability

It is the probability of how often the software fails. The measurement is often expressed in MTBF (Mean Time Between Failures). The requirement is needed to ensure that the processes work correctly and completely without being aborted. It can handle any load and survive and survive and even capable of working around any failure.

Compatibility

It is supported by version above all web browsers. Using any web servers like localhost makes the system real-time experience.

Flexibility

The flexibility of the project is provided in such a way that is could run on different environments being executed by different users.

Safety

Safety is a measure taken to prevent trouble. Every query is processed in a secured manner without letting others to know one's personal information. A conversational

agent that interacts with users using natural language is called a Chatbot. Many chatbots have been developed using text communication starting from ELIZA that simulates a psychotherapist to PARRY which simulates a paranoid patient. ELIZA is well known artificial therapist. The bot attempts to rephrase the questions of the client and responds on certain keywords. If no keyword is found ELIZA replies with fixed phrases to keep the conversation going. Medicine is a field in which help is critically needed. This is because these chatbots lack the human touch and empathy that a fitness professional can provide. In many cases, users may need emotional support, reassurance, and guidance as to how to proceed with their problems. A chatbot may never be able to fill in a human agent's shoes. It encompasses the tasks that determine the need for analyzing, documenting, validating, and managing software or system requirements. The requirements should be documentable, actionable, measurable, testable, and traceable related to identified business needs or opportunities and define to a level of detail, sufficient for system design. It is the first step in the requirement analysis process which lists the requirements of software systems including functional, performance and security requirements. The function of the system depends mainly on the quality hardware used to run the software with given functionality.

4.9 NON- FUNCTIONAL REQUIREMENTS

Portability

It is the usability of the same software in different environments. The project can be run in any operating system.

Performance

These requirements determine the resources required, time interval, throughput and everything that deals with the performance of the system.

Accuracy

The result of the requesting query is very accurate and high speed of retrieving information. The degree of security provided by the system is high and effective.

Maintainability

Project is simple as further updates can be easily done without affecting its stability. Maintainability basically defines that how easy it is to maintain the system. It means that how easy it is to maintain the system, analyze, change and test the application. Maintainability of this project is simple as further updates can be easily done without affecting its stability.

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

The feasibility study investigates the problem and the information needs of the stakeholders. It seeks to determine the resources required to provide an information systems solution, the cost and benefits of such a solution, and the feasibility of such a solution.

The goal of the feasibility study is to consider alternative information systems solutions, evaluate their feasibility, and propose the alternative most suitable to the organization. The feasibility of a proposed solution is evaluated in terms of its components.

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must justify. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

Social feasibility for an NLP-based fitness chatbot involves considering various factors related to its acceptance and adoption by users within a social context. Here are some key aspects to assess:

User Privacy and Security: Users should feel comfortable sharing personal fitness information with the chatbot. Ensure robust privacy policies, data encryption, and transparent data handling practices to gain users' trust.

Cultural Sensitivity: The chatbot should be culturally sensitive and avoid language or content that may offend or alienate users from different backgrounds. Adapt the chatbot's responses and recommendations to respect diverse cultural norms and values.

Accessibility: Ensure the chatbot is accessible to users with different abilities, including those with disabilities. Provide options for text-based and voice-based interactions, and consider accessibility features such as screen readers for visually impaired users.

User Empowerment: The chatbot should empower users to make informed decisions about their fitness goals and behaviors. Provide accurate information, personalized recommendations, and educational resources to help users achieve their objectives.

Inclusivity: Design the chatbot to cater to users of all fitness levels, body types, and age groups. Avoid promoting unrealistic body standards or perpetuating stereotypes that may discourage certain users from engaging with the chatbot.

CHAPTER 5

IMPLEMENTATION DETAILS

Building an NLP-based fitness chatbot involves several key components:

Natural Language Understanding (NLU) : Implement NLU models to comprehend user messages. Techniques like intent recognition and entity extraction are crucial for understanding user queries related to fitness, such as workout routines, diet plans, or health-related questions.

Dialogue Management : Develop a dialogue management system to maintain context and manage the flow of conversation. This involves determining appropriate responses based on the current state of the conversation and the user's intents.

Knowledge Base : Integrate a knowledge base containing information on fitness, exercise routines, nutrition, and health tips. This can be used to provide accurate and relevant responses to user queries.

Personalization : Incorporate personalization features to tailor recommendations and advice based on user preferences, goals, fitness level, and health conditions.

Feedback Loop : Implement a feedback loop mechanism to continuously improve the chatbot's performance. This involves collecting user feedback and using it to update the bot's knowledge base and improve its understanding and responses over time.

Integration with APIs : Integrate with external APIs to fetch real-time data, such as nutritional information, workout schedules, or fitness tracking data from wearable devices.

User Engagement : Include features to keep users engaged, such as progress tracking, goal setting, motivational messages, and reminders.

Multimodal Capabilities : Consider adding support for multimodal interactions, allowing users to communicate via text, voice, or even images/videos for exercises or meal planning.

Privacy and Security : Ensure the bot complies with privacy regulations and employs security measures to protect sensitive user information.

Testing and Evaluation : Thoroughly test the chatbot to ensure it performs well across various scenarios and user inputs. Evaluation metrics could include

accuracy of responses, user satisfaction, and engagement metrics.

By combining these elements, you can create an effective NLP-based fitness chatbot that provides personalized guidance and support to users on their fitness journey.

Define Goals and Scope : Determine the objectives of your chatbot (e.g., provide workout suggestions, track progress, offer nutritional advice) and the specific functionalities it will offer.

Data Collection : Gather a diverse dataset of fitness-related information, including workout plans, exercise routines, nutrition tips, and common user queries.

Natural Language Understanding (NLU) : Use NLP techniques to preprocess and understand user inputs. This involves tasks such as tokenization, stemming, and named entity recognition to extract relevant information.

Intent Recognition : Implement a mechanism to classify user intents based on their input. This could involve training a machine learning model (e.g., using techniques like support vector machines or neural networks) on labeled data or using pre-trained models like BERT or GPT.

Dialog Management : Design the conversation flow of your chatbot, including handling greetings, providing responses, and managing context across multiple turns in the conversation.

Integration with Fitness APIs : Integrate your chatbot with fitness-related APIs to access real-time data on exercises, workouts, nutrition information, and other relevant content.

Personalization : Tailor the responses and recommendations of the chatbot based on user preferences, goals, fitness level, and any other relevant information collected during the conversation.

Feedback and Iteration : Continuously collect feedback from users and iterate on your chatbot's design and functionality to improve its performance and user satisfaction over time. Continuously collect feedback from users and iterate on your chatbots design and fuctionally to improve its performance and user satisfaction

CHAPTER 6

RESULTS AND DISCUSSIONS, PERFORMANCE ANALYSIS

6.1 RESULTS

The project's results encompass several key aspects, starting with the development of a robust NLP-intensive bot system. This system serves as the project's core outcome, designed to effectively process natural language text, discern user intentions, and deliver appropriate responses through chat or voice interfaces. It represents a significant technological achievement in bridging the gap between human language and computer understanding, with applications spanning from customer service to healthcare and information retrieval.

To ensure the system's reliability and quality, a comprehensive testing framework has been meticulously crafted. This framework accommodates a variety of testing types, including unit tests for intent matching, integration tests for Finite State Machine (FSM) behaviour, security assessments, usability evaluations, and performance testing. Rigorous testing ensures that the bot functions seamlessly, addresses security threats like DDoS attacks, and offers a user-friendly experience.

In addition to the technical aspects, the project results feature thorough documentation, comprising user manuals, developer guides, and system documentation. These resources facilitate ease of use, maintenance, and further development. Usability evaluations and improvements enhance the user experience, while a formal review and acceptance process guarantee alignment with initial objectives. Finally, a deployment plan sets the stage for real-world implementation, marking the culmination of the project's successful execution. Assess the coherence and fluency of the conversational flow between the user and the chatbot. Ensure that the chatbot can maintain context and handle follow-up questions effectively.

6.2 PERFORMANCE ANALYSIS

Response Time: Measure the time taken by the chatbot to respond to user queries. Responses should be timely to maintain a seamless conversational experience.

Accuracy of Responses: Evaluate the accuracy of the chatbot's responses by comparing them to expected or correct answers. Use test datasets and user feedback to assess the correctness of responses.

Natural Language Understanding (NLU): Analyze the chatbot's ability to understand user intents and extract relevant information from user queries. Evaluate the accuracy of intent classification and entity extraction.

Conversational Flow: Assess the coherence and fluency of the conversational flow between the user and the chatbot. Ensure that the chatbot can maintain context and handle follow-up questions effectively.

Personalization: Measure the chatbot's ability to provide personalized recommendations and guidance based on user preferences, goals, and past interactions. Evaluate the effectiveness of personalized features in improving user engagement and satisfaction.

User Satisfaction Surveys: Collect user feedback through surveys or feedback forms to assess user satisfaction with the chatbot's performance. Ask users to rate aspects such as response accuracy, helpfulness, responsiveness, and overall satisfaction.

Error Analysis: Conduct error analysis to identify common errors or misunderstandings made by the chatbot. Analyze misclassified intents, incorrectly recognized entities, or poorly generated responses to identify areas for improvement.

Below are the snapshots of the performance system:

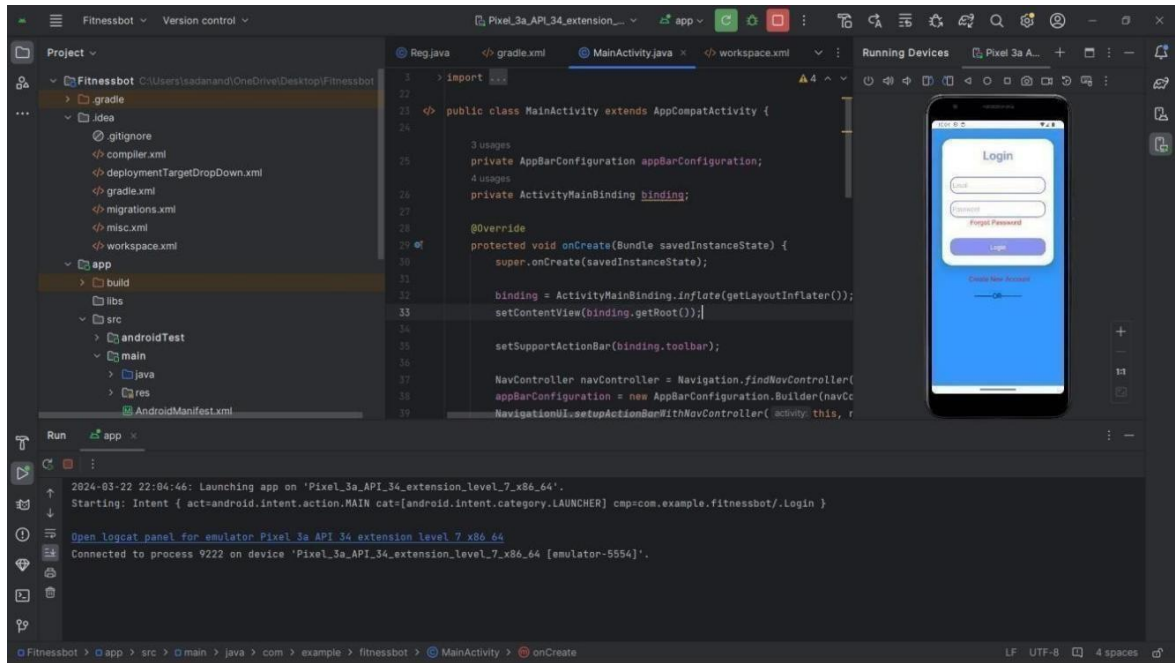


Fig 6.1: User interface

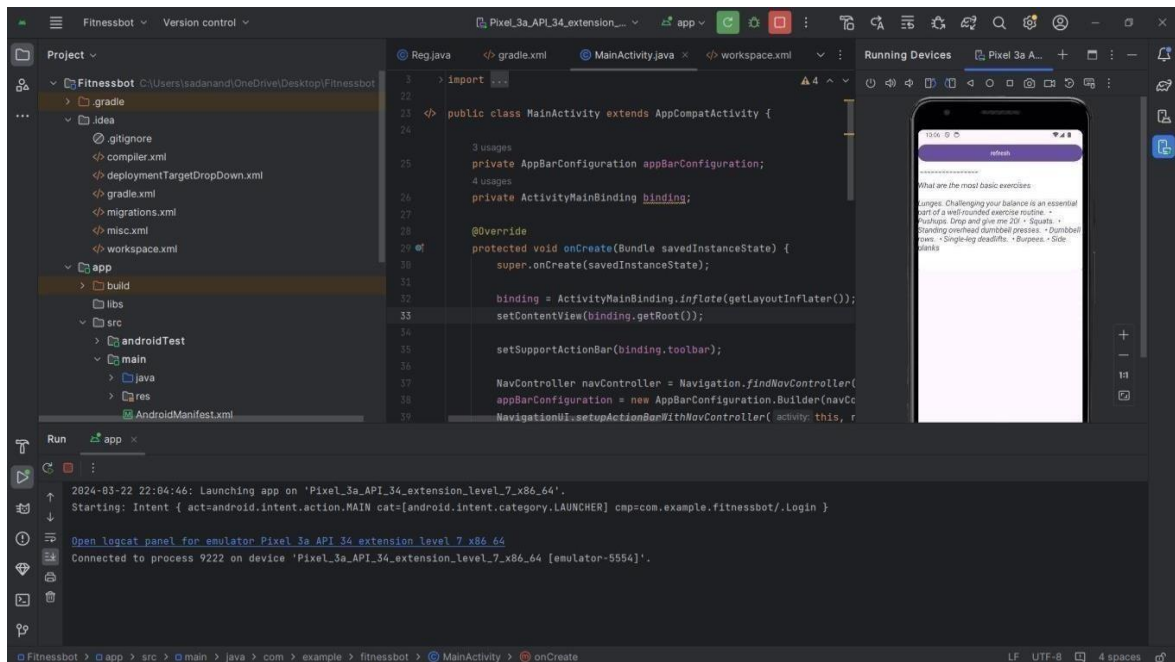


Fig 6.2: User Output

6.3 : LOGIN PAGE

The login form allows users to enter their credentials to authenticate and access the chatbot application. It typically includes fields for: Username or email address: Users enter their unique identifier to log in. Password: Users input their password associated with their account for authentication. Remember me option: A checkbox allowing users to choose whether they want their login credentials to be remembered for future sessions. Forgot password link: A hyperlink that users can click to initiate the password recovery process if they forget their password. Option for new users to navigate to the registration page if they don't have an account yet. Design the login page with a responsive layout to ensure compatibility across various devices and screen sizes (desktop, tablet, mobile). Provide visual feedback (e.g., loading spinners, success/error messages) to indicate the progress and outcome of the login process.



Fig 6.3 : Login Page

6.4 : SEARCH BAR

The search bar features a text input field where users can type their search queries. This field is typically accompanied by a blinking cursor to indicate the active input focus. Placeholder text inside the input field provides users with guidance on what type of information they can search for. For example, it might say "Search here" or "Type to search." A search icon or button adjacent to the text input field allows users to initiate the search action. Clicking on the icon or pressing the enter/return key on the keyboard triggers the search functionality. The NLP-based fitness chatbot search bar allows users to quickly find relevant fitness-related information, exercises, nutrition tips, or answers to specific queries using natural language input. A text input field where users can type their queries or keywords related to fitness. Placeholder text to provide guidance on the expected format (e.g., "Search for exercises, nutrition tips, etc.").

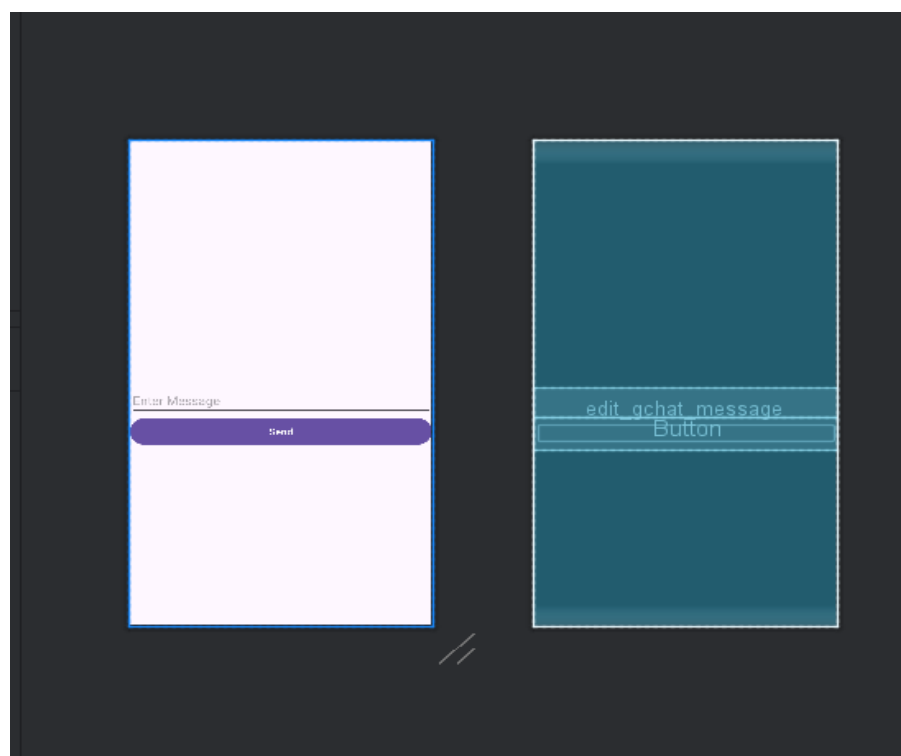


Fig 6.4: Search Bar

6.5 : OUTPUT

Describe the text-based responses provided by the chatbot, including informative messages, recommendations, instructions, and motivational prompts. Explain any visual elements incorporated into the chatbot's responses, such as images, charts, or diagrams related to exercise routines, nutrition plans, or progress tracking. Detail any interactive elements within the chatbot's responses, such as buttons, menus, or links, allowing users to take further actions or navigate to additional content. The chatbot may suggest specific exercises tailored to the user's fitness goals, such as strength training exercises for muscle building or cardio exercises for weight loss. Provide nutritional advice, meal plans, or dietary recommendations based on the user's fitness objectives and dietary preferences. Respond to user queries with relevant information, such as explanations of fitness concepts, definitions of exercise terms, or explanations of workout techniques.

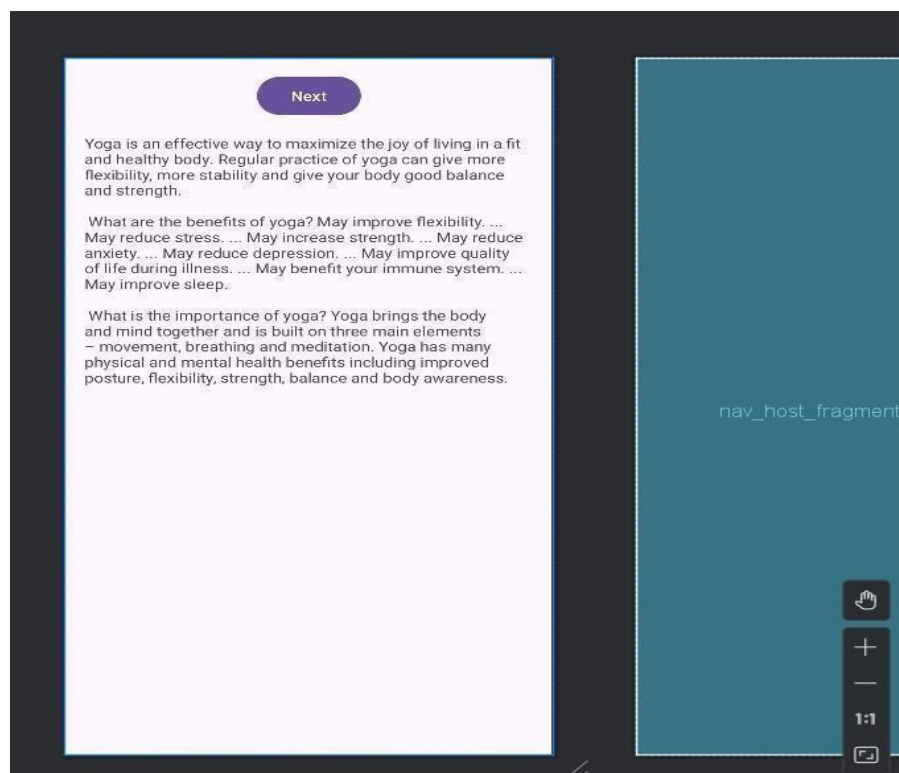


Fig 6.5: Output

CHAPTER 7

SUMMARY AND CONCLUSION

7.1 SUMMARY

1. User Needs Analysis: Understand user needs, preferences, and pain points related to fitness.
2. Technology Assessment: Evaluate NLP techniques, machine learning algorithms, and natural language understanding models suitable for building the Chabot.
3. Data Collection and Annotation: Gather relevant data and annotate it to train NLP models for understanding user queries and generating responses.
4. Chabot Design and Development: Design the conversational flow, user interface, and functionality of the Chabot within the mobile application.
5. Integration of Fitness Tracking Features: Integrate fitness tracking capabilities such as workout logging, meal tracking, and progress monitoring into the mobile application.
6. Privacy and Security Considerations: Implement robust security measures to protect user data, especially sensitive health information.
7. Performance Analysis: Evaluate response time, accuracy of responses, NLU capabilities, conversational flow, personalization, user engagement metrics, error handling, scalability, and security.
8. Feedback and Iteration: Gather feedback from users and use it to iterate on the design and functionality of the Chabot and mobile application.

7.2 CONCLUSION

In summary, the development and deployment of an NLP-based fitness Chabot with a mobile application involve several key steps and considerations:

1. User Needs Analysis: Understand user needs, preferences, and pain points related to fitness and health.
2. Technology Assessment: Evaluate NLP techniques, machine learning algorithms, and natural language understanding models suitable for building.

3. Data Collection and Annotation: Gather relevant data and annotate it to train NLP models for understanding user queries and generating responses.
4. Chatbot Design and Development: Design the conversational flow, user interface, and functionality of the Chatbot within the mobile application.
5. Privacy and Security Considerations: Implement robust security measures to protect user data, especially sensitive health information.
6. Performance Analysis: Evaluate response time, accuracy of responses, NLU capabilities, conversational flow, personalization, user engagement metrics, error handling, scalability, and security.
7. Feedback and Iteration: Gather feedback from users and use it to iterate on the design and functionality of the Chatbot and mobile application.

Moreover, NLP-powered fitness Chatbot's possess the capacity to tackle prevalent challenges encountered by individuals in their pursuit of fitness goals. They can combat misinformation by providing evidence-based information, dispelling myths, and guiding users toward scientifically supported practices. Additionally, they foster accountability and motivation by offering encouragement, setting achievable goals, and tracking progress. This can help combat the issue of adherence, as many people struggle to maintain their fitness routines over time. However, there are challenges associated with NLP fitness Chatbot's that must be acknowledged. One of the primary concerns is the accuracy and reliability of the information they provide. Ensuring that the content and recommendations are based on sound scientific principles is crucial to their effectiveness and user safety. Furthermore, privacy and data security must be a top priority, as these Chatbot's may collect sensitive personal.

In conclusion, NLP fitness Chatbot's represent a transformative force in the fitness and health industry, offering personalized guidance and support to individuals of all fitness levels. They have the potential to democratize access to fitness knowledge and address common challenges such as misinformation and adherence. However, ensuring the accuracy and reliability of the information provided and safeguarding user privacy are critical considerations.

Looking ahead, the future of NLP fitness Chabot's is promising. With advancements in machine learning and NLP, these virtual assistants will become even more sophisticated, offering increasingly personalized and effective guidance. Integrating them with wearable devices and smart technology could further enhance their capabilities. Nevertheless, it will be essential to continuously monitor and regulate these Chabot's to ensure they meet ethical and safety standards.

In a world where health and well-being are of paramount importance, NLP fitness Chabot's have the potential to empower individuals to take control of their fitness journeys and make informed, sustainable choices. As technology continues to evolve, so too will the role of these Chabot's in shaping a healthier future for all. Throughout the development and deployment process, it's essential to prioritize user privacy, cultural sensitivity, inclusivity, and ethical considerations to ensure the chatbot's social feasibility and acceptance within diverse user communities. By integrating domain-specific knowledge, continuously fine-tuning the model, and leveraging user feedback, the chatbot can offer accurate, helpful, and engaging interactions tailored to individual user needs. Performance analysis of the chatbot should encompass various metrics, including accuracy, user engagement, satisfaction, and conversation flow, to assess its effectiveness and identify areas for improvement. Through iterative refinement and optimization, the chatbot can evolve into a valuable tool for empowering users to make informed decisions, adopt healthy behaviors, and maintain long-term fitness habits. feedback, the chatbot can offer accurate, helpful, and engaging interactions .

Overall, an NLP-based fitness chatbot has the potential to enhance user experiences, promote positive behavior change, and contribute to the broader goal of improving public health and well-being in the digital age. With ongoing advancements in NLP technology and continued efforts to address user needs and preferences, the future of fitness chatbots is promising and full of opportunities for innovation and impact, continuously fine-tuning the model, and leveraging user feedback, the chatbot can offer accurate, helpful, and engaging interactions .

REFERENCES

- [1] Wendy Chapman, Clinical natural language processing in 2025: Leveraging the variety of texts of clinical interest, American medical informatics,2016.
- [2] Ronan M. T.Fleming, A Survey of Current Trends in Computational Drug Repositioning, Bioinformatics,2016.
- [3] Marcos Tuliode Oliveiraet, Natural Language Processing in Radiology: A Systematic Review, Radiologic Brasilia, 2018.
- [4] Yann LeCun, Deep Learning Applicationsin Medical Image, IEEE Access, 2017.
- [5] Ibrahim M. Hefny, Deep Learning for Healthcare Applications Based on Physiological Signals,Healthcare Engineering,2018.
- [6] Alessia D'Amico , Natural Language Processing in Oncology, Medical Systems, 2019..
- [7] Noha M. Esawy, Natural Language Processing for EHR-Based Pharmacological, Healthcare Engineering,2019.
- [8] Gustavo Carneiro, A Surveyon Deep Learningin Medical Image Analysis, Medical Systems, 2017.
- [9] Ye Jia, A Review on Deep Learning Techniques in Speech Synthesis, Healthcare Engineering,2019.
- [10] N.D.Kononowicz , Artificial Intelligence in Health Care: Anticipating Challenges, Medical Science, 2019.
- [11] Ibrahim M. Hefny, Deep Learning for Healthcare Applications Based on Physiological Signals,Healthcare Engineering,2018.
- [12] Alessia D'Amico , Natural Language Processing in Oncology, Medical Systems, 2019..
- [13] Marcos Tuliode Oliveiraet, Natural Language Processing in Radiology: A Systematic Review, Radiologic Brasilia, 2018.

APPENDIX

A. SOURCE CODE:

```
package com.example.fitnessbot.sharedprefarence;

import android.content.Context;
import
android.content.SharedPrefer
ences; import
android.content.SharedPrefer
ences.Editor;
//import android.preference.PreferenceManager;

public class SharedPreference {

    public static final String PREFS_NAME =
"AOP_PREFS"; public static final String
PREFS_KEY = "AOP_PREFS_String";

    public
    Share
    dPrefe
    rence(
    ){
    super(
    );
    }

    public void save(Context
    context, String text) {
    SharedPreferences
```

```

settings;
Editor editor;

//settings = PreferenceManager.getDefaultSharedPreferences(context);
settings = context.getSharedPreferences(PREFS_NAME,
Context.MODE_PRIVATE); //1editor = settings.edit(); //2

editor.putString(PREFS_KEY, text); //3

editor.commit(); //4
}

public String
getValue(Context
context) {
    SharedPreferences
    s settings;
    String text;

//settings = PreferenceManager.getDefaultSharedPreferences(context);
settings = context.getSharedPreferences(PREFS_NAME,
Context.MODE_PRIVATE);text = settings.getString(PREFS_KEY, null);
return text;
}

public void clearSharedPreferences(Context context) {SharedPreferences settings;
Editor editor;

//settings = PreferenceManager.getDefaultSharedPreferences(context);
settings = context.getSharedPreferences(PREFS_NAME,
Context.MODE_PRIVATE);editor = settings.edit();

editor.clear(); editor.commit();
}

public void removeValue(Context context) {SharedPreferences settings;
Editor editor;

```

```
settings = context.getSharedPreferences(PREFS_NAME,
Context.MODE_PRIVATE);editor = settings.edit();
```

```
editor.remove(PREFS_KEY);editor.commit();
}
}
```

#LOGIN CODE

```
package com.example.fitnessbot;
```

```
import androidx.appcompat.app.AppCompatActivity;
import android.app.Activity; import android.content.Intent;
import android.content.SharedPreferences; import android.os.Bundle;
import android.view.View; import android.widget.Button; import
android.widget.EditText; import android.widget.TextView; import
android.widget.Toast;
```

```
import com.example.fitnessbot.sharedprefarence.SharedPreference;
```

```
public class Login extends AppCompatActivity {
```

```
TextView textView;
```

```
private static final int RC_SIGN_IN = 1; String text;
```

```
private SharedPreference sharedPreference; Activity context = this;
```

```
EditText et1,et2; String st1,st2; Button btn;
```

```
TextView fpwd,register; @Override
```

```
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState); setContentView(R.layout.activity_login);
sharedPreference = new SharedPreference();
```

```
text = sharedPreference.getValue(context); et1=(EditText)
```

```
findViewById(R.id.username); et2=(EditText) findViewById(R.id.password);
```

```
fpwd=(TextView) findViewById(R.id.fpwd); register=(TextView)
```

```

findViewById(R.id.newacc);register.setOnClickListener(new
View.OnClickListener() {
@Override
public void onClick(View view) {
Intent intent = new Intent(Login.this, Reg.class);startActivity(intent);
}
});
btn=(Button) findViewById(R.id.loginButton); btn.setOnClickListener(new
View.OnClickListener() {
@Override
public void onClick(View view) {

SharedPreferences sh = getSharedPreferences("MySharedPref",
MODE_PRIVATE);String un = sh.getString("uname", "");
String up = sh.getString("upwd", "");if(st1!=un){
Intent intent = new Intent(Login.this, MainActivity.class);startActivity(intent);
}else{

Intent intent = new Intent(Login.this, MainActivity.class);startActivity(intent);
}
}
});
}
}

```

#SEARCH CODE

```

package com.example.fitnessbot;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent; import android.os.Bundle; import android.view.View; import
android.widget.Button; import android.widget.EditText;

```

```

import com.google.firebase.database.DatabaseReference;import
com.google.firebase.database.FirebaseDatabase;

public class Search extends AppCompatActivity {EditText txt;
Button btn; DatabaseReference db;@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState); setContentView(R.layout.activity_search);
txt=(findViewById(R.id.edit_gchat_message));
btn=(findViewById(R.id.button_gchat_send)); Datacraft dc=new Datacraft();
db = FirebaseDatabase.getInstance().getReference();dc.setQuestlon("qqq");
dc.setAns("AAA"); db.child("FitnesData").child("NOTE").push().setValue(dc);
btn.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
String etxt=txt.getText().toString();
Intent intent = new Intent(Search.this, viewdata.class);intent.putExtra("TEXT",etxt);
startActivity(intent);
}
});
}
}

```

#MAIN ACTIVITY

```

package com.example.fitnessbot;

import android.content.Intent;import android.os.Bundle;

import com.google.android.material.snackbar.Snackbar;

import androidx.appcompat.app.AppCompatActivity;

import android.view.View;

import androidx.core.view.WindowCompat;import
androidx.navigation.NavController; import androidx.navigation.Navigation;
import androidx.navigation.ui.AppBarConfiguration;import
androidx.navigation.ui.NavigationUI;

```

```

import com.example.fitnessbot.databinding.ActivityMainBinding;

import android.view.Menu; import android.view.MenuItem;

public class MainActivity extends AppCompatActivity {

    private AppBarConfiguration appBarConfiguration; private ActivityMainBinding
    binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        binding = ActivityMainBinding.inflate(getLayoutInflater());

        setContentView(binding.getRoot());

        setSupportActionBar(binding.toolbar);

        NavController navController = Navigation.findNavController(this,
        R.id.nav_host_fragment_content_main);
        appBarConfiguration = new
        AppBarConfiguration.Builder(navController.getGraph()).build();
        NavigationUI.setupActionBarWithNavController(this, navController,
        appBarConfiguration);

        binding.fab.setOnClickListener(new View.OnClickListener() { @Override
        public void onClick(View view) {
            Intent intent = new Intent(MainActivity.this, Search.class); startActivity(intent);
        }
        });
    }
}

```

```
@Override
```

```
public boolean onCreateOptionsMenu(Menu menu) {  
    // Inflate the menu; this adds items to the action bar if it is present.  
    getMenuInflater().inflate(R.menu.menu_main, menu);  
    return true;  
}
```

```
@Override
```

```
public boolean onOptionsItemSelected(MenuItem item) {  
    // Handle action bar item clicks here. The action bar will  
    // automatically handle clicks on the Home/Up button, so long  
    // as you specify a parent activity in AndroidManifest.xml. int id = item.getItemId();  
  
    //noinspection SimplifiableIfStatement if (id == R.id.action_settings) {  
        return true;  
    }  
  
    return super.onOptionsItemSelected(item);  
}
```

```
return super.onOptionsItemSelected(item);  
}
```

```
@Override
```

```
public boolean onSupportNavigateUp() {  
    NavController navController = Navigation.findNavController(this,  
        R.id.nav_host_fragment_content_main);  
    return NavigationUI.navigateUp(navController, appBarConfiguration)  
        || super.onSupportNavigateUp();  
}  
}
```

#VIEW DATA

```
package com.example.fitnessbot;
import androidx.appcompat.app.AppCompatActivity;
import android.app.Activity; import android.os.Bundle; import android.view.View;
import android.widget.Button; import android.widget.ListView; import
android.widget.Toast;
import com.example.fitnessbot.sharedprefarence.SharedPreference; import
com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError; import
com.google.firebase.database.DatabaseReference; import
com.google.firebase.database.FirebaseDatabase; import
com.google.firebase.database.ValueEventListener; import
com.google.firebase.storage.StorageReference;
```

```
import java.util.ArrayList;
```

```
public class viewdata extends AppCompatActivity { private StorageReference
storageReference; private StorageReference riversRef; FirebaseDatabase
database; DatabaseReference myRef; DatabaseReference db;
```

```
CustomAdapter adapter; ListView lv;
String otp;
Button refresh1, send;
```

```
Boolean saved;
ArrayList<Datacraft> fbhdata=new ArrayList<>(); private SharedPreference
sharedPreference; Activity context=this;
String text; @Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState); setContentView(R.layout.activity_viewdata);
lv=(findViewById(R.id.vlist)); refresh1=(findViewById(R.id.ref1));
```

```

Bundle bundle = getIntent().getExtras(); text = bundle.getString("TEXT");
getDddata();
db = FirebaseDatabase.getInstance().getReference(); adapter = new
CustomAdapter(viewdata.this, getDddata());

lv.setAdapter(adapter); Datacraft dc = new Datacraft();

refresh1.setOnClickListener(new View.OnClickListener() {

@Override
public void onClick(View v) { getDddata(); lv.setAdapter(adapter);
}
});
}

public ArrayList<Datacraft> getDddata() { database =
FirebaseDatabase.getInstance(); myRef = database.getReference("FitnesData");
// text=etext. getText().toString();
// text="Achalasia";
Toast.makeText(getApplicationContext(), "eeee=" +text,
Toast.LENGTH_SHORT).show();

myRef.child("NOTE").orderByChild("questlon").equalTo(text).addListenerForSingl
eValueEvent(new ValueEventListener() {
@Override
public void onDataChange(DataSnapshot dataSnapshot) { fbhdata.clear();
for (DataSnapshot d : dataSnapshot.getChildren()) { Datacraft user =
d.getValue(Datacraft.class); fbhdata.add(user);
}
}
}
}

```

```
super.onCreate(savedInstanceState); setContentView(R.layout.activity_reg);
et1=(EditText) findViewById(R.id.regEmail); et2=(EditText)
findViewById(R.id.regPassword);et2=(EditText) findViewById(R.id.cpass);
btn=(Button) findViewById(R.id.regButton);
```

```
btn.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) { st1=et1.getText().toString();
st2=et2.getText().toString(); st2=et1.getText().toString(); SharedPreferences
sharedPreferences =
getSharedPreferences("MySharedPref", MODE_PRIVATE);
SharedPreferences.Editor myEdit = sharedPreferences.edit();
```

```
// write all the data entered by the user in SharedPreference and apply
myEdit.putString("uname", st1);
myEdit.putString("upwd", st2);myEdit.apply();
Intent intent = new Intent(Reg.this, Login.class);startActivity(intent);
Toast.makeText(Reg.this, sharedPreferences.getString("uname",""),
Toast.LENGTH_SHORT).show();
}
}
```

B. SCREENSHOTS :

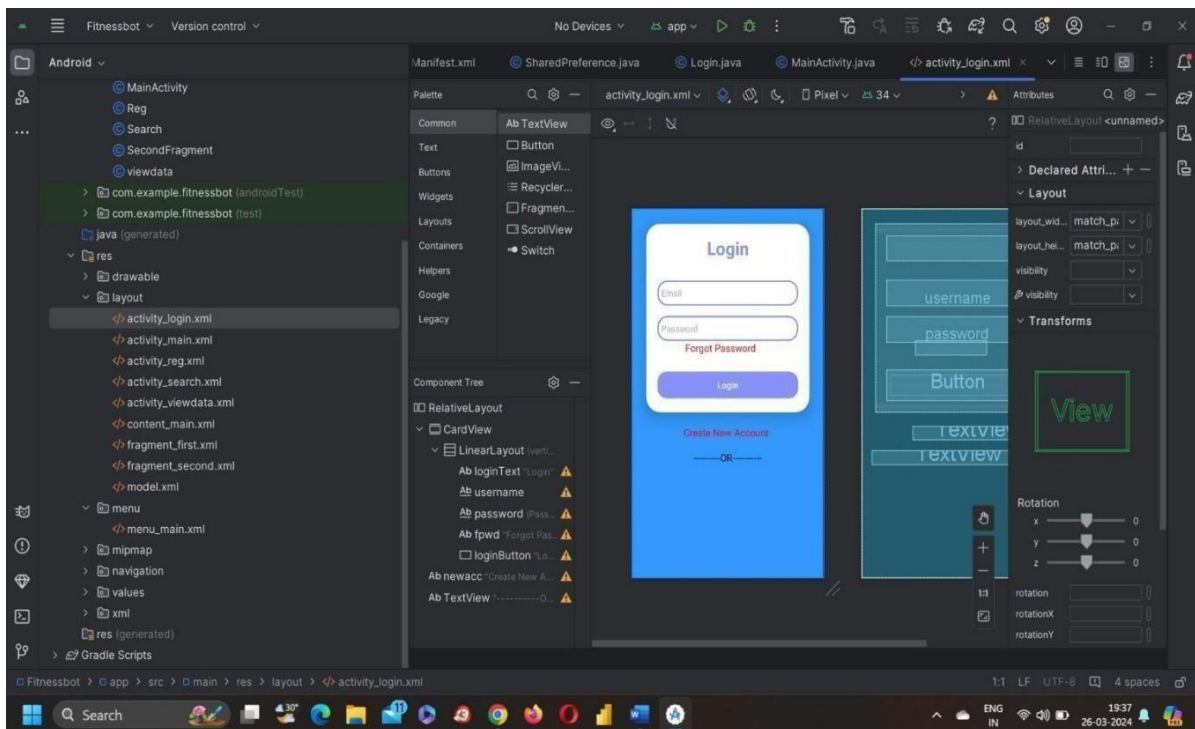


Fig B.1

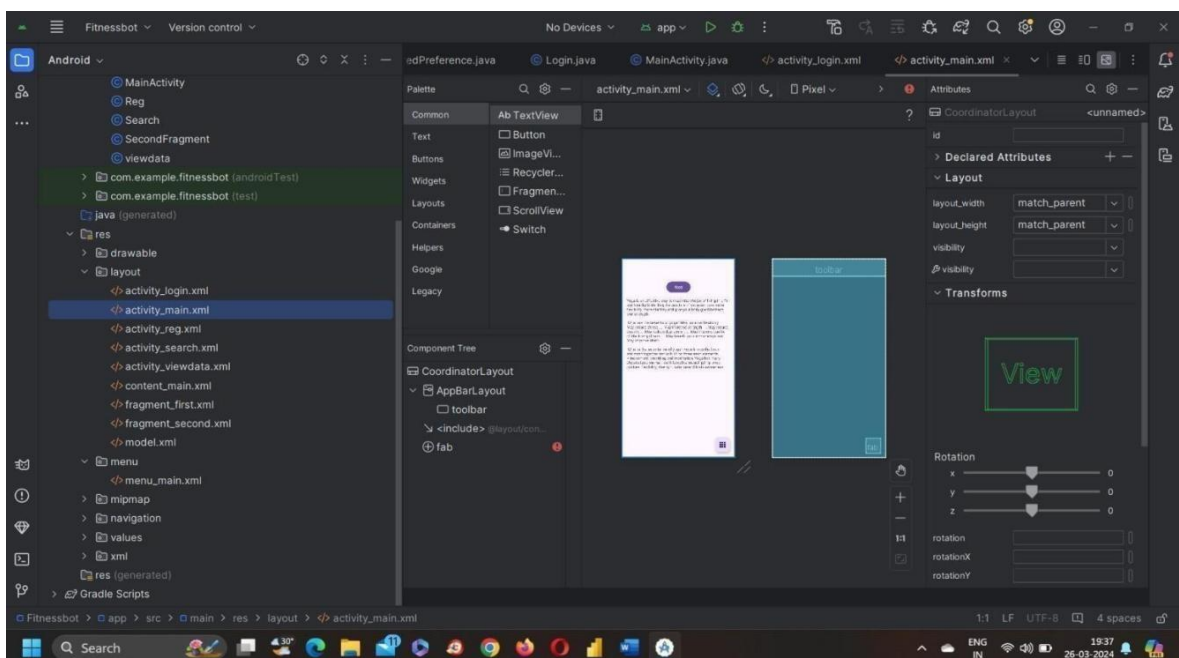


Fig B.2

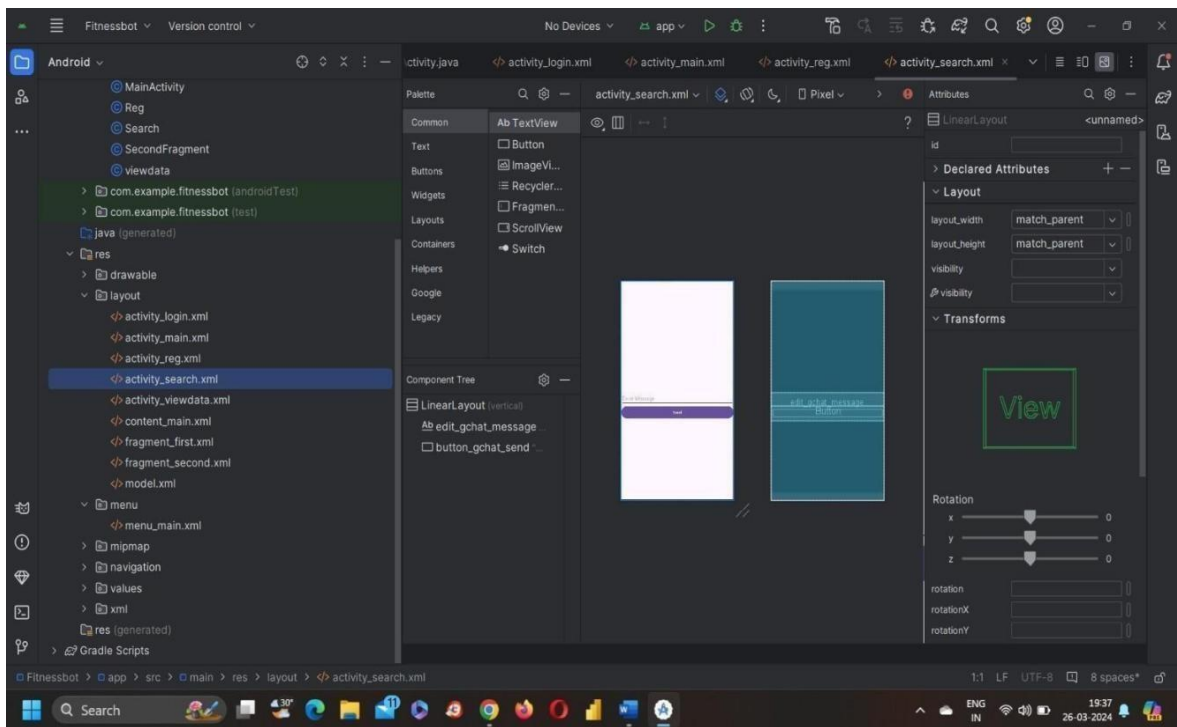


Fig B.3

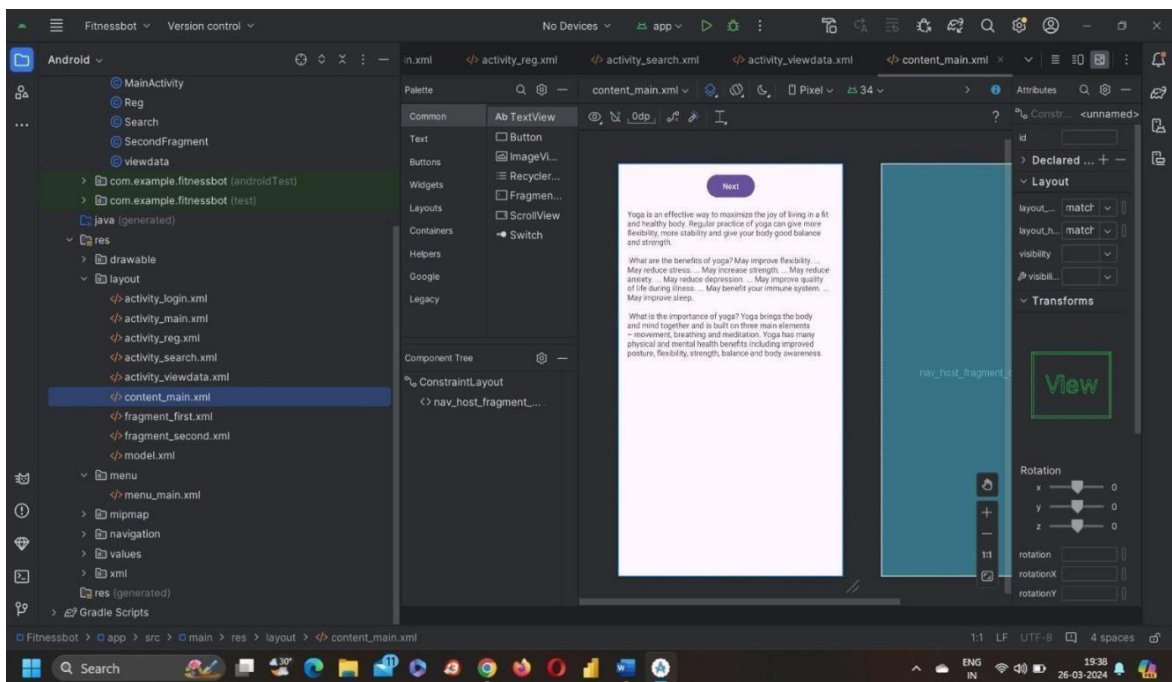


Fig B.4

C.CERTIFICATE :

