

SPORTS FITNESS FOR COLLEGE STUDENTS USING NATURAL LANGUAGE PROCESSING

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

By

PARUPALLI EESHWAR RAJ(Reg.No – 40110908)
POLLI NAVEEN KUMAR REDDY(Reg.No -40110952)



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

SATHYABAMA

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

Accredited with Grade “A++” by NAAC

JEEPIAAR NAGAR, RAJIV GANDHISALAI, CHENNAI – 600119

APRIL – 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Category - I University by UGC

Accredited "A++" by NAAC | 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 Parupalli Eeshwar Raj (40110908) and Poli Naveen Kumar Reddy(40110952) who carried out the Project entitled "Sports Fitness For College Students Using Natural Language Processing" under my supervision from November 2023 to April 2024.


13/4/24
Internal Guide

Dr. M. SREE KRISHNA, M.E., Ph.D.


Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.

HEAD OF THE DEPARTMENT

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

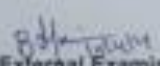
Submitted for Viva voce Examination

Jayalaxmi Nagar, Rajiv Gandhi Salai,

Chennai - 600 119.

15/04/2024


15/4
Internal Examiner


External Examiner

DECLARATION

I, Parupalli Eeshwar Raj(40110908), hereby declare that the Project Report entitled "Sports Fitness For College Students Using Natural Language Processing" done by me under guidance of Dr. M. SREE KRISHNA, M.E., Ph.D., is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE: 15/04/2024

PLACE: Chennai

P. Eeshwar Raj
SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph. D, Dean**, School of Computing, **Dr. L. Lakshmanan 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 to my Project Guide **Ms.SREEKRISHNA M.M.E.,(Ph.D)** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

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

ABSTRACT

Mobile Apps 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 Mobile App 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 Mobile App 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. The chatbot utilizes natural language processing techniques to understand user queries and respond with relevant information and recommendations. The mobile app offers an intuitive and user-friendly interface, allowing users to interact with the chatbot seamlessly. Privacy and security considerations are prioritized throughout the development process. 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. The proposed idea is to create a chatbot that interacts like a human being, giving the consumer the impact that she or he is interacting with every other individual. Chatbot applications permit college students to get right of entry to sports-associated facts from everywhere, every time. Students need to pick out a discipline question type, and the fields will be dispatched to the bot, in order to be used to interact. Students can ask statistics and many different questions about sports health or sports activities pastime. This device reduces the quantity of work the university sports instruct has to do in providing statistics to college students, as well as the amount of labor the teach has to do in answering all the college students' questions. Team information, player performances and game schedules. A scoping overview targeted on AI applications to sell PA or expect associated behaviors or health effects was carried out in PubMed, Web of Science, Cochrane Library and EBSCO. AI techniques have been summarized and labeled to discover synergy, patterns, and traits that could inform destiny studies. In addition, a brief primer on key AI methods in the scope of PA is provided to sell understanding and broader software.

TABLE OF CONTENT

	TITLE		Page No.
	ABSTRACT		v
	LIST OF FIGURES		vii
1	INTRODUCTION		08
2	LITERATURE SURVEY		10
	2.1	A Chatbot For Medical Purpose	10
	2.2	Contextual Chatbot For Healthcare Purposes	11
	2.3	Healthcare Chatbot Using NLP	11
	2.4	Risk Analysis Of The Project	12
	2.5	Objectives	12
3	REQUIREMENTS ANALYSIS		13
	3.1	Functional And Non Functional Requirements	13
	3.2	Hardware Requirement	14
	3.3	Software Requirement	14
4	DESCRIPTION OF PROPOSED SYSTEM		15
	4.1	Existing and Proposed System	15
	4.2	Feasibilitys	16
	4.3	Selected Methodology	17
	4.4	System Architecture And Flow Diagram	18
	4.5	Testing and It's Types	19
	4.6	Project Management Plan Modules	22
5	RESULT AND DISCUSSIONS		24
6	URL DIAGRAMS		28
7	REFERENCES		34
8	SOURCE CODE		38

LIST OF FIGURES

LIST OF FIGURS		
FIGURE NO	FIGURE NAME	Page No.
4.4	Flow Diagram	18
4.4.1	System Architecture Diagram	18
6.1	Data Flow Diagram	29
6.2	Use Case Diagram	29
6.3	Class Diagram	30
6.4	Sequence Diagram	31
6.5	Activity Diagram	32

CHAPTER 1

INTRODUCTION

A chatbot is a computer software that uses text-to-text technology to conduct an online chat conversation. University ChatBot is an internet program that uses artificial intelligence to converse with humans. In the Other online programs like the "Eliza," "Cleverbot," and others have been created in the past. So this system is an online application that responds to a question from a learner. In most cases, the Student's Question is a conversation. There is no fixed format for chatting with the bot; it can be done in any way. In our project, we employed NLP to accomplish this. The goal of this project is to create an intelligent chat-bot system that will handle sports activities/sports fitness like as diet plan, sports scheduling, sports day, team statistics, player performance, and game schedules, and details on the documents that must be attached, among other things. The method is designed to assist in the streamlining of the coach process for various sports. It interacts with students in real time, removing any lingering questions they may have about the sports process. Users can get information on college-related events through the system. The system has an easy-to-use graphical user interface.

The system uses A.I. and M.L. to resolve queries. The concept for the college chatbot is based on Google Assistant, which will answer student questions via a web application. It provides links that are related to the user's query. By clicking on the link, the user can get answers to their inquiries. The chatbot extracts data and displays conversations between the human user and the bot using a standard algorithm. It provides information about sports activities such as results diet plan, sports scheduling, sports day, intra and inter college sports information, departmental timetables, among other things. The system, in most situations, finds the intentions, or keywords.

Mobile Apps are most typically employed in customer contact centers to manage incoming conversations and guide consumers to the right resource. Internally, they're typically used for onboarding new workers and assisting all staff with routine tasks like vacation scheduling, training, purchasing computers and business supplies, and other self-service activities that don't require human participation. The possibilities they open up in the fitness business are practically limitless, but for now, they're mostly being used to meet initial marketing and communication objectives. Firstly, it aims to provide rational feedback to users from diverse perspectives, fostering a well-rounded and insightful interaction. Secondly, the system is designed to be a learning platform, leveraging experienced inputs to continually improve and deliver higher-quality outputs.

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 notifications, and knowledge management may be easily automated and made available 24 hours a day, seven days a week with an intelligent

chatbot, while expanding access to widely used voice and text-based conversational interfaces. Chatbots are most typically employed in customer contact centers to manage incoming conversations and guide consumers to the right resource. Internally, they're typically used for onboarding new workers and assisting all staff with routine tasks like vacation scheduling, training, purchasing computers and business supplies, and other self-service activities that don't require human participation. The possibilities they open up in the fitness business are practically limitless, but for now, they're mostly being used to meet initial marketing and communication objectives.

While currently available in English, a future goal is to make the system multilingual, ensuring accessibility for a broader audience. Furthermore, users will have the option to contribute guidance and tips throughout the gaming process, promoting a collaborative and engaging environment. Lastly, the system is committed to delivering accurate information, covering team details, player performance, and game schedules to keep users well-informed about the entire operation. These objectives collectively reflect a commitment to creating a versatile, user-friendly, and informative platform.

CHAPTER 2

LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then the next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system. The major part of the project development sector considers and fully survey all the required needs for developing the project. For every project Literature survey is the most important sector in software development process. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, resource requirement, man power, economy, and company strength. Once these things are satisfied and fully surveyed, then the next step is to determine about the software specifications in the respective system such as what type of operating system the project would require, and what are all the necessary software are needed to proceed with the next step such as developing the tools, and the associated operations.

2.1 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 analyze the input in the form of speech and generate responses that are similar to humans.

2.2 Contextual Chatbot for Healthcare Purposes

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 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 with hospitals to provide support which 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.

2.3 Healthcare Chatbot Using NLP

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 books, which helps the patient know more about their disease and helpsto improve their health. The user can achieve the benefit of a healthcare chatbot only when itcan 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 provides 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 natural language processing techniques such as NLTK for Python can be applied to analyse speech and intelligent responses can be found by designing an engine to provide appropriate human like responses.

2.4 Risk Analysis Of The Project

Students need to manually visit the college to get their queries answered by the college help desk. This process consumes a lot of time as well as money as the customer needs to visit college if it's miles away from home. Also, this process may lead to communication gaps between students and colleges. To design and create an online chatbot system for helping students based on a well-informed database that will be used for pattern matching using Artificial Intelligence and Natural Language Processing.

2.5 Objective

The objectives are as follows:

- 1) Reasoning: It will answer the user from any perspective.
- 2) Learning: It will learn from the inputs it experiences for better output.
- 3) Multilingual: It will be accessible in 1 language – English.
- 4) Guide throughout the sports process and will give option to users to give suggestions.
- 5) Giving accurate information about the whole activity including the team statistics, player performance, and game schedules, etc.

CHAPTER 3

3. REQUIRMENTS 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 analysing, 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.

3.1 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 particular 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 users.

- **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 in order 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 it has the ability to 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.

3.1.1 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, analyse, change and test the application. Maintainability of this project is simple as further updates can be easily done without affecting its stability.

3.2 HARDWARE REQUIREMENTS

System	: Pentium i3 Processor
Hard Disk	: 500 GB.
Monitor	: 15" LED
Input Devices	: Keyboard, Mouse
Ram	: 2 GB

3.3 SOFTWARE REQUIREMENTS

Operating System : Windows 10
Coding Language : Python

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

4.1 EXISTING SYSTEM

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.

PROBLEMS OF EXISTING SYSTEM

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.

PROPOSED SYSTEM

The proposed system would be an application that delivers answers to the scholar's or user's questions. Users will simply use the chatbot that is currently in use to ask queries. Students are free to communicate in any format they wish; there is no restriction that they follow a set format. The responses are relevant to the user's questions. If the responses are invalid or unavailable, the queries are stored in the unrequited table, which is essentially created by the coach. Later, the coach can update such queries; however, in the event of an emergency, we will send a message stating that "our coach will contact you quickly. In the sports domain, a chatbot system can be designed to provide users with information on team statistics, player performance, and game schedules.

4.2 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 be justified. Thus the developed system is 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 amodest 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.

4.3 Selected Methodology

Natural Language Processing(NLP)

The term "Natural Language Processing" (NLP) refers to the area of artificial intelligence that studies how computers and human language communicate. It entails the creation of computational models and algorithms that let machines decipher, comprehend, and produce natural language. Language translation, sentiment analysis, voice recognition, and chatbots are just a few of the many uses for NLP. Enabling computers to comprehend the nuances of human language, including grammar, syntax, semantics, and context, is the main problem in NLP. Several methods, including machine learning, deep learning, and natural language comprehension, are used in NLP. Deep learning models employ neural networks to comprehend the content, while machine learning algorithms are used to train models that can recognise patterns in language data. It is designed to understand natural language inputs and generate human-like responses.

4.4 SYSTEM ARCHITECTURE

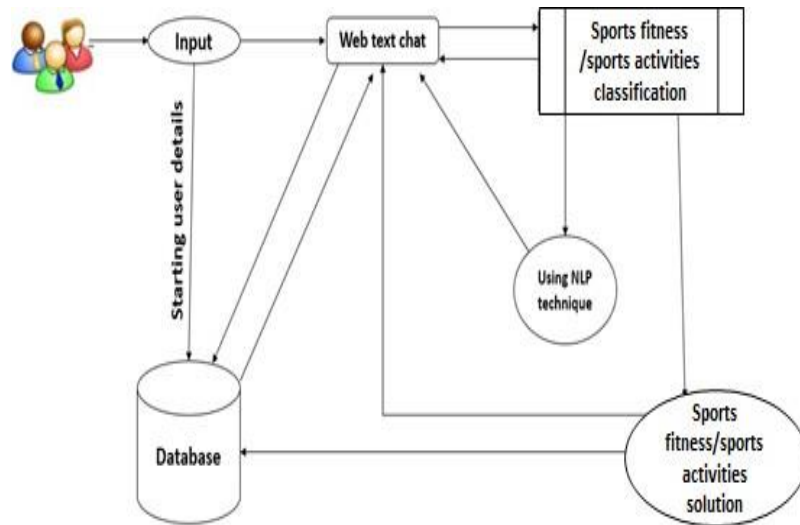


Figure 1:User interaction with the chatbot

- **FLOW DIAGRAM**

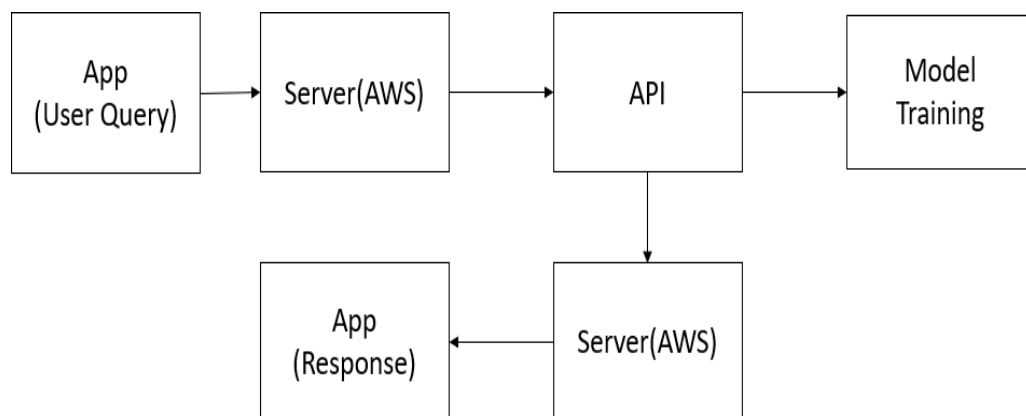


Figure 2: How the user queries are solve

4.5 Testing And It's Types

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.

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 the product is built as per customer requirements.

Basics of software testing

There are two basics of software testing: black box testing and white box testing.

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 functional testing.

White box Testing

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing. Black box testing is often used for validation and white box testing is often used for verification.

Types of testing

There are many types of testing like

- Unit Testing
- Integration Testing
- Functional Testing
- System Testing
- Stress Testing
- Performance Testing
- Usability Testing
- Acceptance Testing
- Regression Testing
- Beta Testing

Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

Integration Testing

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing.

Functional Testing

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing.

System Testing

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

Stress Testing

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

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 Project Management Plan Modules

- User Validation And Extraction Of Problems
- Extracted issues With Trained Datasets
- Specifying The Fitness Solution

MODULE DESCRIPTION

- **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 loss myweight and am getting tired.” In this case, the system should extract words like “weight loss” and “tired” (and not substrings like “read” or “okay”).

- **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 enduser by the Mobile App. The Mobile App checks whether the identified fitness solution is a major issue or minor issue based on the conditions built in the Mobile App.

CHAPTER 5

RESULT AND DISCUSSIONS

Interaction with the bot :

Instead of contacting the sport assist desk or game group of workers, customers can connect with the bot to ask or inquire about game-associated activities. Student questions usually follow discussion. There's no set layout for chatting with a bot, so you can do something you need. This was achieved the use of NLP in our assignment. Users can seek information approximately sports activities related events which includes in shape day, team information, player overall performance and sports activities time table and the date and time of every. Reduce the wide variety of adolescents and apprentices at the activity to save time and money. It refers back to the capacity of an teacher and student to talk successfully with every other.

Offer information :

The chat-bot machine will take care of sports education/sports like meal planning, sports making plans, sports day, intra and inter-university sports activities statistics, departmental time desk and details of files to be connected. Whenever the user submits a question or query, the precise query/queries could be observed. The chat-bot tests whether the query already exists in the database or not. If the solution is inside the database, the answer is sent to the person.

If the query does not exist in the database, the person will get the solution as an invalid query. And such questions might be sent to the administrator. Admin can take a look at the questions submitted by using the person and if those questions are found valid, then Admin can save the facts related to such questions within the database.

LOGIN PAGE :

This represents home page of the project, which displays the Begin by accessing the chatbot interface, usually located on a website, app, or messaging platform. This may involve clicking on a chat icon or starting a conversation through a designated chat window.

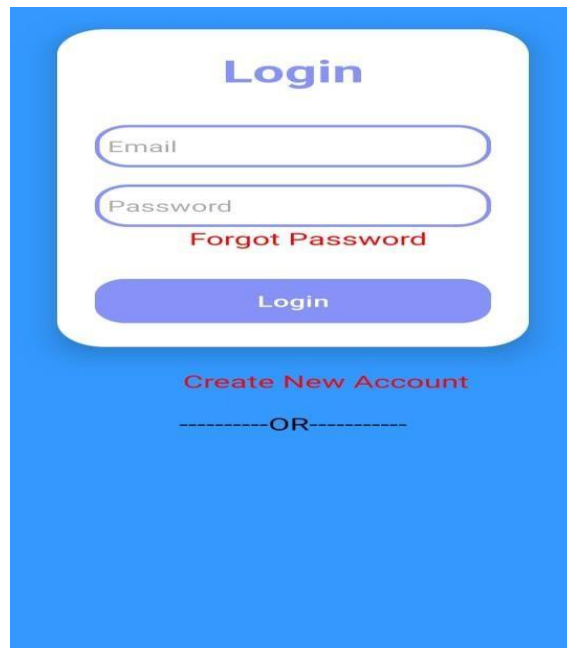


Figure 3:Login page of the user.

Instructions by bot:

This represents the interaction between user and the bot. Start interacting with the chatbot by typing your questions, requests, or providing relevant information. The chatbot will process your input and generate responses based on its programming and the data it has access to. Be specific in your queries to receive accurate and relevant information. Engage in a back-and-forth conversation by responding to the bot's prompts or asking follow-up questions.

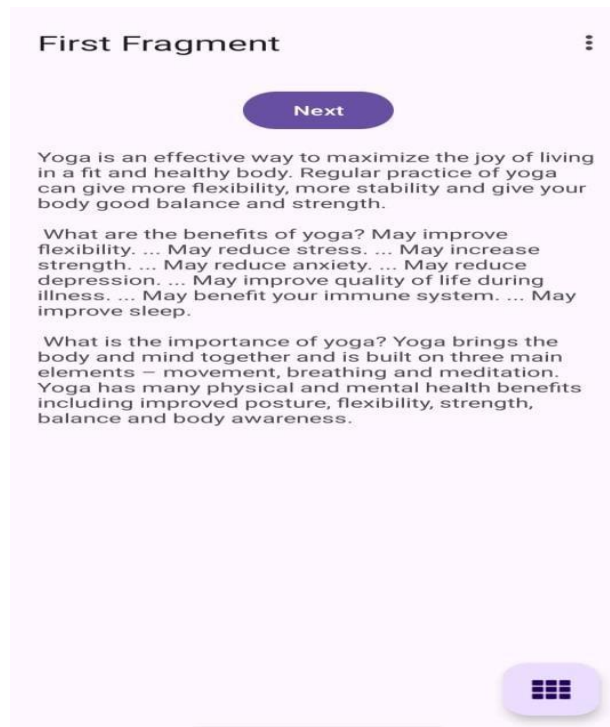


Figure 4:Instructions should be followed by the user.

Request query :

The user need to send the queries to the bot .As the conversation progresses, the bot will generate responses based on the information you provide and its programming. Follow the flow of the conversation, ask questions if needed, and utilize the bot's capabilities. Once your interaction is complete, the bot may offer a summary or ask if there's anything else it can assist you with. Express gratitude or end the conversation based on your needs.



Figure 5:Send the query to the bot

Respond to bot :

After sending the queries to the bot, the bot will respond to the query. After getting the response to the query the bot asking for anything else to be solved and replies visit again. A healthy heart, lungs and muscle tissues are a part of this Exercise is a further function that has a few effects on physical fitness There is mental clarity and emotional stability thru our movements It impacts our frame and thoughts. The blessings of physical interest. Future scope of the project could be AI Based Healthcare Mobile App system using NLP can also include an mobile assistant in it which will be more functions will be added and can be accessed by many users. Which will also reduce the time and will also be accurate in the health details of patients given to the doctors. We can add biometric system for more secure authentication process.

Virtual fact offers human beings an interesting new manner to workout Fear of vigorous exercising: Makes you sweat without leaving domestic. Motion tracking generation can you can dispose of form mistakes and help muscle developers the knowledge and

assessment of formal mistakes are made by means of the common person I provide.

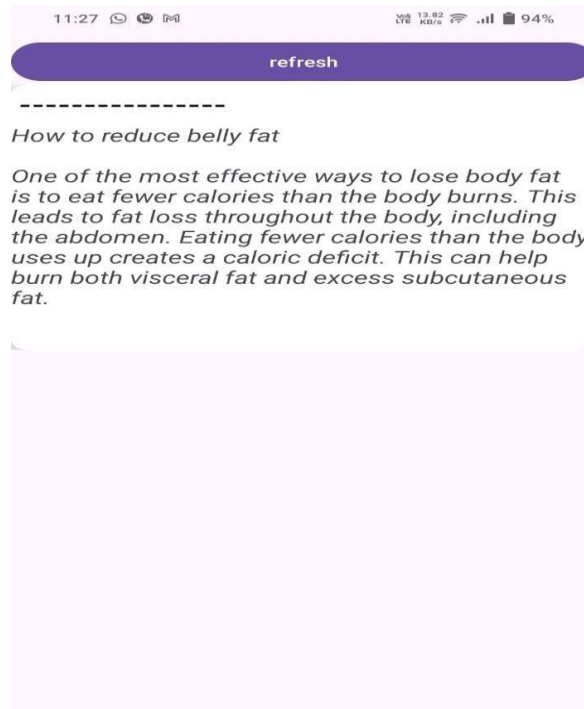


Figure 6: Bot respond to the user/solving the query.

CHAPTER 6

UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to expressthe design of software projects.

GOALS:

- The Primary goals in the design of the UML are as follows:
 1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
 2. Provide extendibility and specialization mechanisms to extend the core concepts.
 3. Be independent of particular programming languages and development process.
 4. Provide a formal basis for understanding the modeling language.
 5. Encourage the growth of OO tools market.
 6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
 7. Integrate best practices.

6.1 Data flow diagram:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

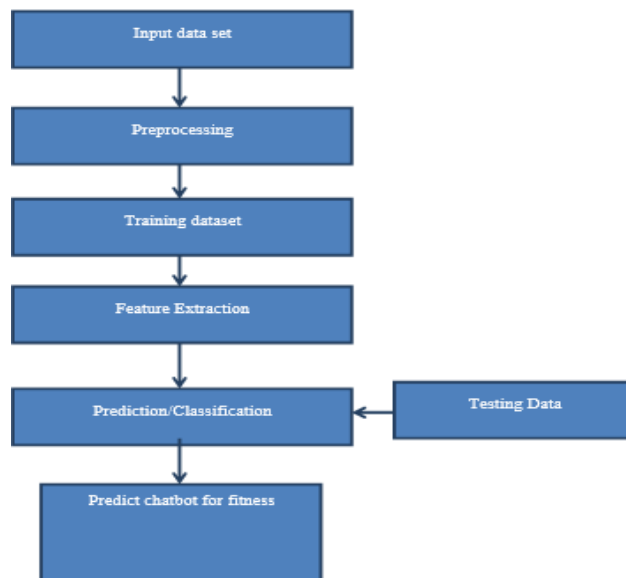


Figure 7:User input to the predicted chatbot

6.2 Use Case Diagram:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted

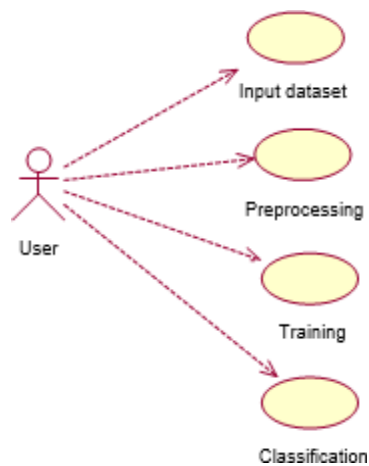


Figure 8:Use case diagram

6.3 Class Diagram:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

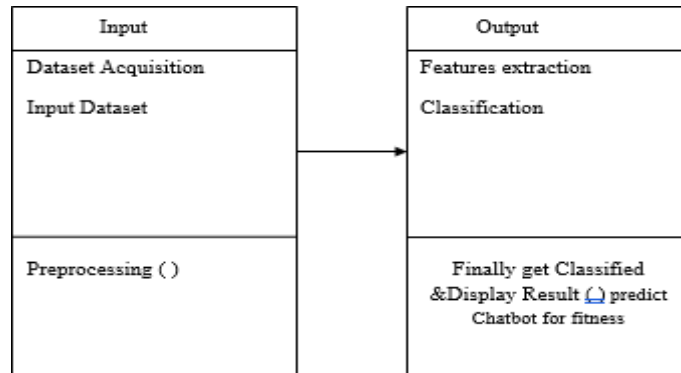


Figure 9: Static structure diagram

6.4 Sequence Diagram:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

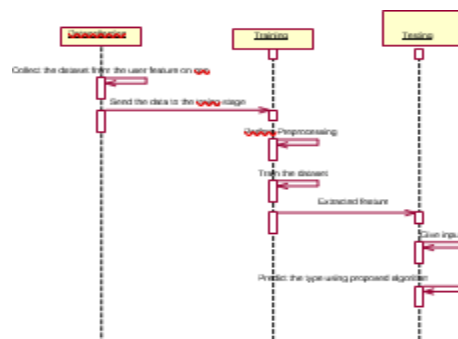


Figure 10: Interaction diagram

6.5 Activity Diagram:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

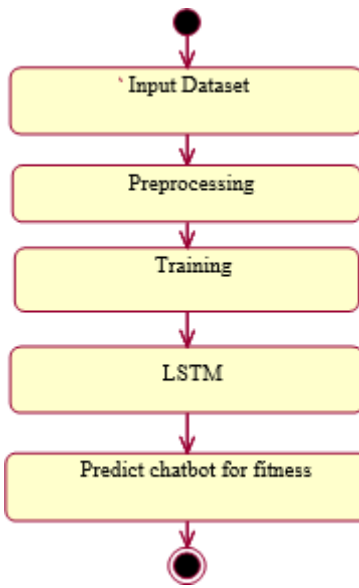


Figure 11: Graphical representations of workflows.

Conclusion:

In this project, we are implementing an AI Based Healthcare Mobile App system using NLP which is easy to use and more secure than the current system it will cure the diseases and helps to maintain proper health in the current system. It will reduce the possibility of diseases. It will provide an accurate information about the health symptoms and medicines to the patients.

The government will also keep the track of the medicines supplied to the medicals and hospitals. However, this system has some drawback like network issues in rural areas might cause a delay in delivering an OTP. The implementation of the system is not costly.

Future scope:

Future scope of the project could be AI Based Healthcare Mobile App system using NLP can also include an mobile assistant in it which will be more functions will be added and can be accessed by many users. Which will also reduce the time and will also be accurate in the health details of patients given to the doctors. We can add biometric system for more secure authentication process.

REFERENCES

- [1] Vaibhav Tode, Himanshu Gadge, Prateek Kachare and Sudarshan Madane, CureBot -An Artificially Intelligent Interactive Bot for Medical Diagnostics International Research journal of Engineering and Technology (IRJET)., Vol.7, no.12 (Dec 2020).
- [2] Satyendra Praneel Reddy Karri and Dr Santosh kumar, Deep Learning Techniques for Implementation of Chatbots, 2020 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, INDIA, (2020) January 22-24.
- [3] Dipesh Kadariya, Revathy Venkataramanan, Hong Yung Yip, Maninder Kalra, Krishnaprasad Thirunarayanan and Amit Sheth, kBot: Knowledge-Enabled personalized Chatbot for Asthma Self-Management, 2019 IEEE International Conference on Smart Computing (SMARTCOMP), Washington, DC, USA,(2019) June 12-15.
- [4] R. Dharwadkar, N.A. Deshpande, A medical chatbot. Int. J. Comput. Trends Technol. (IJCTT). **60**(1) (2018).
- [5] S. Divya, V. Indumathi, S. Ishwarya, M. Priyasankari, S.K. Devi, A self-diagnosis medical chatbot using artificial intelligence. J. Web Dev. Web Designing **3**(1) (2018).
- [6] N. Jyothirmayi, A. Soniya, Y. Grace, C. Reddy Kumar Kishor, B.V. Murthy Ramana, Survey on chatbot conversational system. J. Appl. Sci. Comput. **6**(1) (2019).
- [7] Milla T Mutiwokuziva, Melody W Chanda, Prudence Kadebu, A neural network based chatbot, 2nd International Conference on Communication and Electronics Systems (ICCES 2021).
- [8] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev. in press.
- [9] Rizvi, S.; Sohail, I.; Saleem, M.M.; Irtaza, A.; Zafar, M. A Smart home appliances power management System for the handicapped, elder and Blind People. In Proceedings of the 4th International Conference on Computer and Information Sciences (ICCOINS), Kuala Lumpur, 2018.

[10] Donghua Zhou, Construction of the Evaluation Index System of Scientific Fitness Literacy of the Elderly Group Based on the Integration of Sports and Medicine Under the Big Data Analysis Algorithm, 2022 International Conference on Wearables, Sports and Lifestyle Management (WSLM)

[11] Lin Xiao, Sports Score Management and Physical Fitness Analysis System Based on Data Mining 2023 4th IEEE Global Conference for Advancement in Technology (GCAT)

[12] Ge Zhibin, Research on School Sports Training under the Background of Big Data 2021 International Conference on Information Technology and Contemporary Sports (TCS).

[13] Li Xuan, Research on Reconstruction of the Concept of "Physical Fitness" in the Context of New Crown Epidemic 2020 International Conference on Modern Education and Information Management (ICMEIM)

[14] Zhongpei Hu, Development of Score Management Software for Dance Sport Competition, 2021 6th International Symposium on Computer and Information Processing Technology (ISCIPT)

[15] Ningning He, Construction of Sports Information Service System for the Disabled from the Perspective of National Fitness, 2021 3rd International Conference on Internet Technology and Educational Informization (ITEI)

[16] Fanrong Kong; Guangpeng Li; Hengchao Zhou, Design and Implementation of Sports Fitness Testing Equipment Based on Knowledge Graph Structure Optimization Algorithm, 2023 International Conference on Data Science and Network Security (ICDSNS)

[17] Lin Ma Research on Physical Fitness Test and Examination System of Sports Athletes Based on Computer Imaging Technology 2023 IEEE 6th International Conference on Information Systems and Computer Aided Education (ICISCAE).

[18] Dan Wang; Yingying Zheng, Digital and Intelligent Image Processing by Artificial Intelligence and Internet of Things Technology in Sports Fitness Detection DOI:10.1109/ACCESS.2022.3218912

[19] Ping Jia, Research on the Construction of Integrated Sports and Medicine and Health Management Platform under the background of internet era 2021 International Conference on Information Technology and Contemporary Sports(TCS)

[20] Cui Liang-zhong; Zhou Gang; Yang Suting, Research on Sports Training Prediction Model Based on Selective Ensemble Learning, 2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)

APPENDIX

A.SOURCE CODE

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

import android.content.Context;
import android.content.SharedPreferences;
import android.content.SharedPreferences.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 SharedPreference() {
        super();
    }

    public void save(Context context, String text) {
        SharedPreferences settings;
        Editor editor;

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

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

        editor.commit(); //4
    }

    public String getValue(Context context) {
        SharedPreferences 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 clearSharedPreference(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();
    }
}

```

```

package com.example.fitnessbot;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.TextView;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.Collections;

```

```

/**

```

* Created by User on 2/3/2017.

*/

```
public class CustomAdapter extends BaseAdapter {
    Context c;
    ArrayList<Datacraft> data;

    public CustomAdapter(Context c, ArrayList<Datacraft> data) {
        this.c = c;

        this.data = data;
    }

    @Override
    public int getCount() {

        return data.size();
    }

    @Override
    public Object getItem(int position) {

        return data.get(position);
    }

    @Override
    public long getItemId(int position) {
        data.removeAll(Collections.singleton(null));
        return position;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {

        if(convertView==null)
        {

            convertView= LayoutInflater.from(c).inflate(R.layout.model,parent,false);
        }
        TextView TltTxt= (TextView) convertView.findViewById(R.id.modelTitle);
        TextView descTxt= (TextView) convertView.findViewById(R.id.modelNote1);

        final Datacraft s= (Datacraft) this.getItem(position);
```

```

descTxt.setText(s.getQuestlon()+"\n\n"+s.getAns()+"\n\n");

Toast.makeText(c,"ccc",Toast.LENGTH_SHORT).show();
/*descTxt.setText(s.getNoteOne());
dateTxt.setText(s.getNoteTwo());*/

//ONITECLICK
convertView.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Toast.makeText(c,"ccc",Toast.LENGTH_SHORT).show();
    }
});
return convertView;
}
}

package com.example.fitnessbot;

public class Datacraft {

    private String sendMsg;
    private String Ans;
    private String Questlon;

    public Datacraft(){}

    public String getAns() {
        return Ans;
    }
    public void setAns(String Ans) {
        this.Ans = Ans;
    }
    public String getQuestlon() {
        return Questlon;
    }
    public void setQuestlon(String Questlon) {
        this.Questlon = Questlon;
    }
}

package com.example.fitnessbot;

```

```

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.navigation.fragment.NavHostFragment;

import com.example.fitnessbot.databinding.FragmentFirstBinding;

public class FirstFragment extends Fragment {

    private FragmentFirstBinding binding;

    @Override
    public View onCreateView(
        LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState
    ) {

        binding = FragmentFirstBinding.inflate(inflater, container, false);
        return binding.getRoot();

    }

    public void onViewCreated(@NonNull View view, Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);

        binding.buttonFirst.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                NavHostFragment.findNavController(FirstFragment.this)
                    .navigate(R.id.action_FirstFragment_to_SecondFragment);
            }
        });
    }

    @Override
    public void onDestroyView() {
        super.onDestroyView();
        binding = null;
    }
}

```

```

}
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 SharedPreferences sharedPreferences;
    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);
        sharedPreferences = new SharedPreferences();
        text = sharedPreferences.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);
            }

        }
    });
}
}
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);
    }

    @Override
    public boolean onSupportNavigateUp() {
        NavController navController = Navigation.findNavController(this,
R.id.nav_host_fragment_content_main);
        return NavigationUI.navigateUp(navController, appBarConfiguration)
            || super.onSupportNavigateUp();
    }
}
package com.example.fitnessbot;

import androidx.appcompat.app.AppCompatActivity;

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.Toast;

public class Reg extends AppCompatActivity {
    EditText et1,et2,et3;
    String st1,st2,st3;
    Button btn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        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 SharedPreferences
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();
            }
        }

    );

}
}
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);
        }
    });
}
}
package com.example.fitnessbot;

import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;
import androidx.navigation.fragment.NavHostFragment;

import com.example.fitnessbot.databinding.FragmentSecondBinding;

public class SecondFragment extends Fragment {

    private FragmentSecondBinding binding;

    @Override
    public View onCreateView(
        LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState
    ) {

```

```

        binding = FragmentSecondBinding.inflate(inflater, container, false);
        return binding.getRoot();
    }

    public void onViewCreated(@NonNull View view, Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);

        binding.buttonSecond.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                NavHostFragment.findNavController(SecondFragment.this)
                    .navigate(R.id.action_SecondFragment_to_FirstFragment);
            }
        });
    }

    @Override
    public void onDestroyView() {
        super.onDestroyView();
        binding = null;
    }
}

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 SharedPreferences sharedPreferences;
    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).addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        fbhdata.clear();
        for (DataSnapshot d : dataSnapshot.getChildren()) {
            Datacraft user = d.getValue(Datacraft.class);
            fbhdata.add(user);
        }
    }

    @Override
    public void onCancelled(DatabaseError databaseError) {
    }
});

return fbhdata;
}
}

```