QRTBOT- CHATBOT APPLICATION FOR PEOPLE IN QUARANTINE

A Project Report

Submitted by

JOSHUA JOSEPH	**JEC17CS059**
JOSIN GEORGE	**JEC17CS060**
SANGEETHA C P	**JEC17CS086**
SHILPA SIVADAS	**JEC17CS093**

to

APJ Abdul Kalam Technological University

in partial fulfillment of the requirements for the award of the Degree of

Bachelor of Technology (B.Tech)

in

COMPUTER SCIENCE & ENGINEERING

Under the guidance of

MR. UNNIKRISHNAN P

CREATING TECHNOLOGY
LEADERS OF TOMORROW
ESTD 2002

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Jyothi Engineering College

NAAC Accredited College with NBA Accredited Programmes*

Approved by AICTE & affiliated to APJ Abdul Kalam Technological University

A CENTRE OF EXCELLENCE IN SCIENCE & TECHNOLOGY BY THE CATHOLIC ARCHDIOCESE OF TRICHUR

JYOTHI HILLS, VETTIKATTIRI P.O, CHERUTHURUTHY, THRISSUR. PIN-679531 PH : +91- 4884-259000, 274423 FAX : 04884-274777

NBA accredited B.Tech Programmes in Computer Science & Engineering, Electronics & Communication Engineering, Electrical & Electronics Engineering and Mechanical Engineering valid for the academic years 2016-2022. NBA accredited B.Tech Programme in Civil Engineering valid for the academic years 2019-2022.

June 2021

DECLARATION

We the undersigned hereby declare that the project report "QRTBOT- Chatbot application for people in quarantine", submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under supervision of Mr. Unnikrishnan P. This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in this submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously used by anybody as a basis for the award of any degree, diploma or similar title of any other University.

Name of Students	**Signature**
JOSHUA JOSEPH (JEC17CS059)	
JOSIN GEORGE (JEC17CS060)	
SANGEETHA C P (JEC17CS086)	
SHILPA SIVADAS (JEC17CS093)	

Place:

Date:

Jyothi Engineering College

NAAC Accredited college with NBA Accredited programmes

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CREATING TECHNOLOGY
LEADERS OF TOMORROW
ESTD 2002

CERTIFICATE

This is to certify that the report entitled " **QRTBOT- CHATBOT APPLICATION FOR PEOPLE IN QUARANTINE** " submitted by JOSHUA JOSEPH(JEC17CS059), JOSIN GEORGE(JEC17CS060), SANGEETHA C P(JEC17CS086), and SHILPA SIVADAS(JEC17CS093) to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree in Bachelor of Technology in **Computer Science & Engineering** is a bonafide record of the project work carried out by them under my/our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Prof. Dr Saju P John

Associate Professor
Internal Supervisor & Head of the Department

ACKNOWLEDGEMENT

We take this opportunity to thank everyone who helped us profusely, for the successful completion of our project work. With prayers, we thank **God Almighty** for his grace and blessings, for without his unseen guidance, this project would have remained only in our dreams.

We thank the **Management** of Jyothi Engineering College and our Principal, **Dr. Sunny Joseph Kalayathankal** for providing all the facilities to carry out this project work. We are grateful to the Head of the Department **Prof. Dr Saju P John** for his valuable suggestions and encouragement to carry out this project work. Our sincere thanks to **Dr. Vinith R**, former Head of the Department for permitting us to make use of the facilities available in the department to carry out the project successfully.

We would like to express our whole hearted gratitude to the project guide **Mr. Unnikrishnan P** for his encouragement, support and guidance in the right direction during the entire project work.

We thank our Project Coordinators **Dr. Swapna B Sasi** & **Mr. Shaiju Paul** for their constant encouragement during the entire project work. We extend our gratefulness to all teaching and non teaching staff members who directly or indirectly involved in the successful completion of this project work.

Finally, we take this opportunity to express our gratitude to the parents for their love, care and support and also to our friends who have been constant sources of support and inspiration for completing this project work.

JOSHUA JOSEPH (JEC17CS059)
JOSIN GEORGE (JEC17CS060)
SANGEETHA C P (JEC17CS086)
SHILPA SIVADAS (JEC17CS093)

VISION OF THE INSTITUTE

Creating eminent and ethical leaders through quality professional education with emphasis on holistic excellence.

MISSION OF THE INSTITUTE

- To emerge as an institution par excellence of global standards by imparting quality Engineering and other professional programmes with state-of-the-art facilities.
- To equip the students with appropriate skills for a meaningful career in the global scenario.
- To inculcate ethical values among students and ignite their passion for holistic excellence through social initiatives.
- To participate in the development of society through technology incubation, entrepreneurship and industry interaction.

VISION OF THE DEPARTMENT

Creating eminent and ethical leaders in the domain of computational sciences through quality professional education with a focus on holistic learning and excellence.

MISSION OF THE DEPARTMENT

- To create technically competent and ethically conscious graduates in the field of Computer Science & Engineering by encouraging holistic learning and excellence.
- To prepare students for careers in Industry, Academia and the Government.
- To instill Entrepreneurial Orientation and research motivation among the students of the department.
- To emerge as a leader in education in the region by encouraging teaching, learning,industry and societal connect

PROGRAMME EDUCATIONAL OBJECTIVES

PEO 1: The graduates shall have sound knowledge of Mathematics, Science, Engineering and Management to be able to offer practical software and hardware solutions for the problems of industry and society at large.

PEO 2: The graduates shall be able to establish themselves as practising professionals, researchers or Entrepreneurs in computer science or allied areas and shall also be able to pursue higher education in reputed institutes.

PEO 3: The graduates shall be able to communicate effectively and work in multidisciplinary teams with team spirit demonstrating value driven and ethical leadership.

PROGRAMME SPECIFIC OUTCOMES

Graduate possess -

PSO 1: An ability to apply knowledge of data structures and algorithms appropriate to computational problems.

PSO 2: An ability to apply knowledge of operating systems, programming languages, data management, or networking principles to computational assignments.

PSO 3: An ability to apply design, development, maintenance or evaluation of software engineering principles in the construction of computer and software systems of varying complexity and quality.

PSO 4: An ability to understand concepts involved in modeling and design of computer science applications in a way that demonstrates comprehension of the fundamentals and trade-offs involved in design choices.

PROGRAMME OUTCOMES

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

COURSE OUTCOMES

COs	Description
C410.1	The students will be able to analyse a current topic of professional interest and present it before an audience.
C410.2	Students will be able to identify an engineering problem, analyse it and propose a work plan to solve it.
C410.3	Students will have gained thorough knowledge in design, implementations and execution of Computer science related projects.
C410.4	Students will have attained the practical knowledge of what they learned in theory subjects.
C410.5	Students will become familiar with usage of modern tools.
C410.6	Students will have ability to plan and work in a team.

CO MAPPING TO POs

	POs											
COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
C410.1	3	2	3	3	3	3	2	3	3	2	3	2
C410.2	2	3	3	3	3	3	2	3	2	3	3	3
C410.3	3	2	3	3	3	3	3	3	2	3	3	3
C410.4	3	3	3	3	3	3	3	2	3	3	3	3
C410.5	2	3	3	3	2	3	3	2	2	3	2	2
C410.6	2	3	3	2	2	3	2	3	2	3	2	2
Average	2.5	2.67	3	2.83	2.67	3	2.5	2.67	2.33	2.83	2.67	2.67

CO MAPPING TO PSOs

	PSOs			
COs	PSO1	PSO2	PSO3	PSO4
C410.1	3	3	1	3
C410.2	2	3	2	3
C410.3	3	1	3	2
C410.4	2	1	3	2
C410.5	3	1	2	2
C410.6	3	2	3	3
Average	2.67	1.833	2.33	2.5

ABSTRACT

COVID-19 was first discovered in December 2019 and has continued to rapidly spread across countries worldwide infecting thousands and millions of people. The virus is deadly, and people who are suffering from prior illnesses or are older than the age of 60 are at a higher risk of mortality. Medicine and Healthcare industries have surged towards finding a cure, and different policies have been amended to mitigate the spread of the virus. The project aims at introducing a medical chatbot called QRTBOT for COVID-19 assistance during quarantine using Machine Learning and Natural Language Processing. The chatbot is built to be a conversational agent that motivates users to discuss about their health issues and returns the diagnosis of disease through a series of queries. It also identifies the emotional state of the user through the text. The main goal of this project is to determine if the user has a high chance of being COVID 19 Positive and keep the users updated regarding this infectious disease. It also helps the people to stay away from loneliness which further doesn't lead to depression and keep them mentally fit.

Keywords: Natural Language Processing, Chatbot, COVID-19, Symptom prediction, Sentiment Analysis, Machine Learning.

CONTENTS

List of Figures **xiii**

List of Abbrevations **xv**

1 INTRODUCTION **1**

1.1 Overview 1

1.2 Objectives 2

1.3 Data Description 2

1.4 Organization of the project 2

2 LITERATURE SURVEY **3**

2.1 Chatbots in the fight against the COVID-19 pandemic 3

2.2 Medbot: Conversational artificial intelligence powered chatbot for delivering tele-health after covid-19 6

2.3 Compassion Driven Conversational Chatbot Aimed for better Mental Health 9

2.3.1 Buddy : The 3AM Friend 10

2.4 Chatbot for Disease Prediction and Treatment Recommendation using Machine Learning 12

3 PROBLEM STATEMENT **16**

4 PROJECT MANAGEMENT **17**

4.1 Introduction 17

4.1.1 Initiation 17

4.1.2 Planing and design 18

4.1.3 Execution 18

4.1.4 Monitoring & controlling 18

4.2 System Development Life Cycle 18

4.2.1 Spiral Model 19

5 METHODOLOGY **21**

5.1 Modules 22

5.1.1 SYMPTOM PREDICTION 22

5.1.2 EMOTION ANALYSIS 25

5.2 System Requirements and Specifications 27

5.2.1 PyCharm . . . 27
5.2.2 Windows 10 . . . 27
5.2.3 Python 3.6.7 . . . 27
5.2.4 SCIKIT learn . . . 27
5.2.5 Pandas . . . 28
5.2.6 NLTK . . . 28
5.2.7 Flask . . . 28
5.2.8 SQLyog . . . 28
5.2.9 PyMySQL . . . 28
5.3 Architecture . . . 29
5.4 Data Flow Diagrams . . . 29
5.4.1 DFD -Level 0 . . . 29
5.4.2 DFD Level-1 Module 1 . . . 30
5.4.3 DFD Level-1 Module 2 . . . 31
5.4.4 DFD Level-2 Training . . . 31
5.4.5 DFD Level-2 Testing . . . 32
5.4.6 DFD Level-2 Emotion Prediction . . . 32
5.4.7 DFD Level-2 Symptom Prediction . . . 32
5.5 UML Diagram . . . 33
5.6 Implementation . . . 34
5.6.1 Start with PyCharm . . . 36
5.6.2 Import Libraries . . . 36
5.6.3 Importing the Dataset . . . 36
5.6.4 Preprocessing . . . 36
5.6.5 Train Test Split . . . 36
5.6.6 Feature Scaling . . . 36
5.6.7 Training . . . 37
5.6.8 Predicting the Test set results . . . 37
5.6.9 Evaluating the Algorithm . . . 37
5.6.10 Import Libraries . . . 37
5.6.11 Compiling a Regular Expression Object . . . 37
5.6.12 Conversion of Text to Vector using TF-IDF Vectorizer: . . . 38
5.6.13 Determines the probability of sentence using VADER . . . 38

6 Results & Discussion **44**
6.1 Covid Symptom Analysis 44
6.1.1 Computing and printing the evaluation metrics of binary classification 45
6.2 Emotion Analysis 45
7 Conclusion & Future Scope **47**
7.1 Conclusion 47
7.2 Future Scope 47

LIST OF FIGURES

Figure No.	Title	Page No.
2.1	CDC coronavirus chatbot	5
2.2	The Architecture of the Conversational Bot	7
2.3	UML diagram showing the use cases of the Bot	9
2.4	Sample images of how chatbot “Buddy” holds up conversation and interacts.	12
2.5	Working model of the proposed system	13
2.6	User login module	14
2.7	User register module	15
4.1	Spiral Model	20
5.1	Chatbot Architecture	29
5.2	DFD Level-0	29
5.3	DFD Level-1 Module 1	30
5.4	DFD Level-1 Module 2	31
5.5	DFD Level-2 Training	31
5.6	DFD Level-2 Testing	32
5.7	DFD- Level 2 Emotion Prediction	32
5.8	DFD- Level 2 Symptom Prediction	32
5.9	UML Interaction Diagram	33
5.10	Proposed Implementation Flow Chart 1	34
5.11	Proposed Implementation Flow Chart 2	35
5.12	Registration Page	40
5.13	HTML5 Validation	40
5.14	Login Page	41
5.15	Health Status Update	41
5.16	Covid-19 Symptom Analysis	42
5.17	Emotion Analysis	42

5.18 Precautions Page 43
5.19 FAQ Page 43
6.1 Covid Symptom Analysis Results 44
6.2 Covid Symptom Analysis Results 45
6.3 Results Of Binary Classification 45
6.4 Emotion Analysis Result 46

List of Abbreviations

COVID : Novel Corona Virus Disease
ML : Machine Learning
NLP : Natural Language Processing
SVM : Support Vector Machine
CDC : Centers for Disease Control and Prediction
WHO : World Health Organization
API : Application Programming Interface
GCP : Google Cloud Platform
VUI : Voice User Interface
SSML : Speech Synthesis Markup Language
NLU : Natural Language Understanding
NLG : Natural Language Generation
LSTM : Long Short Term Memory
RNN : Recurrent Neural Network
KNN : K − Nearest Neighbour
POS : Part of Speech
SDLC : System Development Life Cycle
NLTK : Natural Language Toolkit
SQL : Structured Query Language
CRF : Conditional Random Field
TF − IDF : Term Frequency Inverse Document Frequency
VADER : Valence Aware Dictionary and sEntiment Reasoner
HTML : Hypertext Markup Language
IDE : Integrated Development Environment
DFD : Data Flow Diagram

CHAPTER 1

INTRODUCTION

1.1 Overview

Coronavirus disease (COVID-19) is an infectious disease caused by a newly discovered coronavirus. Most people who fall sick with COVID-19 will experience mild to moderate symptoms and recover without special treatment. The huge number of deaths caused by the novel pandemic COVID-19, which can affect anyone of any sex, age and socio-demographic status in the world, presents a serious threat for humanity and society. At this point, there are two types of citizens, those oblivious of this contagious disaster's danger that could be one of the causes of its spread, and those who show erratic or even turbulent behavior since fear and anxiety invades our surroundings because of confinement and panic of being affected. It was first reported in December 2019 in Wuhan City in China . Up to April 2020, more than 144,000 people have died globally from the COVID-19, while more than 2 millions infections have been confirmed in dozens of countries, according to the World Health Organization, as a result the COVID-19 is now declared a pandemic. In fact, while the number of people who are being treated for COVID-19 is increasing by the day, some citizens are not aware of the real threat of this outbreak which explains its quickly spread all over the world, however others, panicked and desperate, fell headlong into the trap of this grim and even worse committing suicide.

This project aims at proposing a medical chatbot that helps users get engaged during quarantine days. It creates awareness among the users about the risk factors of the COVID-19 and the precautions that must be taken. Through a series of queries using decision tree algorithm the chatbot predicts if the users may have contracted COVID-19. It also analyse the emotional state of the user at the point of time through their chat replies and perform corresponding actions based on their current mood.

1.2 Objectives

The main objective of this project is to introduce an interactive chatbot that help users get aware of the symptoms of novel coronavirus and check if they have high chances of being infected by COVID-19 using Machine Learning. It also analyzes the mental state of the user.

1.3 Data Description

The data for this project is taken from an open source platform known as Kaggle and is available at https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge. The data-set as a whole is generally divided into two categories: training and test. As is the case with a usual machine learning problem, we would be training the model using training dataset and evaluating the performance with the test dataset.

1.4 Organization of the project

The report is organised as follow:

- **Chapter 1:Introduction** Gives an introduction ”Chatbot facility for people who are being quarantined ”.
- **Chapter 2:Literature Survey** Summarizes the various existing techniques that helps in achieving the desired result.
- **Chapter 3: Problem Statement** Discuses about the need for the proposed system
- **Chapter 4:Project Management** Contains the effective project management model to be used for the project.
- **Chapter 5:Methodology** Describes the various steps involved to produce this project.
- **Chapter 6:Results and Discussion**Describes the result of experiment conducted.
- **Chapter 7:Conclusion** Concludes with the future scope of implementation.
- **References** Includes the references for the project.

CHAPTER 2

LITERATURE SURVEY

2.1 Chatbots in the fight against the COVID-19 pandemic

During the novel coronavirus (COVID-19) pandemic, institutions like the Centers for Disease Control and Prevention (CDC) and the World Health Organization (WHO) have begun utilizing chatbots to share information, suggest behavior, and offer emotional support.The CDC has named theirs "Clara".[1]Chatbots are software programs that talk with people through voice or text in their natural language. Some well-known examples include "Alexa" from Amazon, "Siri" from Apple, and "Cortana" from Microsoft. They often come pre-installed on smartphones or home-based smart speakers. In recent years, chatbot use for health-related purposes has increased considerably, from supporting clinicians with clinical interviews and diagnosis to aiding consumers in self-managing chronic conditions. While promising, the use of chatbots may pose safety risks. Chatbots have varied widely in their responses to questions about physical health, suicide, intimate partner violence, substance abuse, and other sensitive conversations. In one study, about a third (29%) of chatbot responses to health questions could have caused harm, and about half of those (16%) could have resulted in death if acted upon. The COVID-19 pandemic puts in stark relief the potential for chatbots to help save lives.

- **Special Features of Pandemics**:

  Pandemics have unique characteristics that make them amenable to tailored interventions deliverable via chatbots. In particular,pandemics differ from other natural disasters in three key ways. First, individual actions can significantly worsen outcomes in a pandemic, given that a single person may infect many others depending on their behavior. Second, the fear of infecting others, especially loved ones or healthcare workers, makes infectious diseases more insidious through disease-related stigma. As a result, people can feel personally responsible for bad outcomes during a pandemic and also hide symptoms from others.[5]Third, the physical gatherings typically used to connect with others in difficult times (e.g., family meals, community centers, sports, spiritual and religious events) are exactly what we are supposed to avoid during a pandemic, worsening the risk for future mental health problems. Chatbots have unique affordances, outlined below, which may mitigate short- and long- term disease burden during infectious disease pandemics

- **Information dissemination**

During a pandemic, people do not know what to do. Doing too little (e.g., not following prophylactic measures) can increase everyone's risk of infection. Doing too much (e.g., going to the emergency room for mild symptoms) can overburden the healthcare system, wasting precious resources. Thus, reliable information sources are crucial to prevent a "misinfodemic": the spread of a disease facilitated by viral misinformation.[6] For instance, during the Zika outbreak in 2016, misleading posts spread faster and were more popular than accurate posts on the large social-media site, Facebook. Because chatbots provide a single answer to most questions, they are able to present concise information from credible sources, which may be less overwhelming than social media or web search engines' long list of results. This matters because false news spreads online both faster and further than accurate news18. Chatbots, in contrast to newspapers and online information sources, can often hear and respond in natural language, improving access for people who cannot read or have difficulty using the internet. They can be available any time of the day to answer questions with up-to-date information, and unlike human experts, can concurrently speak with millions of people at the same time in local languages and dialects.

- **Symptom monitoring**

  During a pandemic, both individuals and institutions want to know how and where infections are spreading. Individuals want to avoid getting sick, and institutions such as hospitals or local governments need data-informed policies to increase capacity (i.e., ordering more testing kits) and to plan social interventions (e.g., closing businesses). However, efforts to quickly and accurately gather population level infection rates are stymied by individuals' fear that disclosing symptoms may harm their professional and social lives. Chatbots may be uniquely well suited for symptom screening in a pandemic because people with stigmatized conditions often avoid seeking health care and education. Prior research suggests people are more willing to disclose sensitive personal symptom information to a chatbot than to a human.[7] This means that people may be more forthcoming with chatbots than other humans, providing timelier and more accurate personal triage and population-level infection rate estimates. Healthcare organizations, large corporations like Apple, Amazon, Facebook, Microsoft, and Tencent, governmental agencies like the CDC, and non-governmental organizations like the WHO have launched or helped develop COVID-19 focused chatbots on platforms available to billions of users, likely with the aim of increasing accessibility.

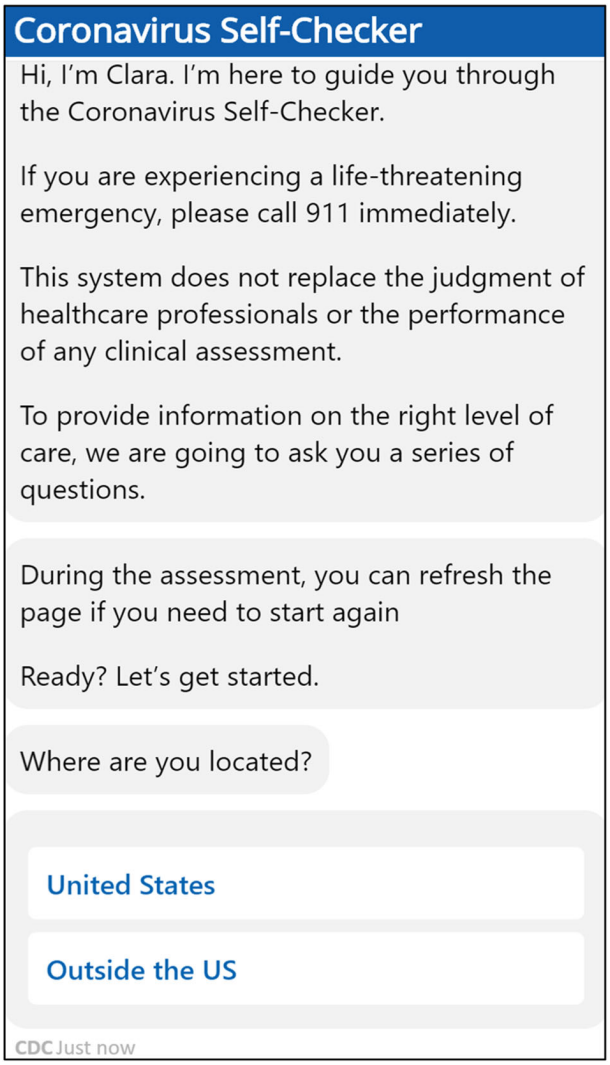


Figure 2.1: CDC coronavirus chatbot

- **Challenges**

  Chatbots may be uniquely useful in a pandemic, but challenges in information dissemination, symptom monitoring, behavior change, and mental health support are worthy of attention. Providing reliable evidence-based information is critical in a pandemic and two issues have material impact: conflicting advice between global and local authorities, and misinformation18. Chatbot developers must decide whose voice to amplify and should provide reliable information from global sources like the WHO, while also coordinating with regional authorities. Both a feature and a challenge of chatbots is their ability to link users to third-party services (e.g., "skills") that then gather and share data with unknown or unexpected consequences. If deployed for symptom screening, which is currently happening for COVID-19, constitutional and regulatory boundaries are tested by sharing health-related information between companies and governments. This concern is not theoretical, as both the United States

and Israel have reportedly explored using digital contact tracing to understand infection vectors. Finally, although chatbots have demonstrated feasibility in behavior change and mental health treatment, they are untested in pandemics and have demonstrated limits in health crisis detection and response.

These challenges, if only addressed in real time during a crisis, may lead to erroneous outputs from a lack of testing. With more than a billion voice searches per month, any health-related mistakes, such as misidentifying key symptoms, would be amplified with extensive harmful repercussions. . Additionally, medical and public health experts must inform what chatbots say, and how they say it. Translating medical information into advice for the public requires expertise and evaluation to prevent unintended consequences. Without proper design and deployment, and ongoing monitoring, chatbots may confuse rather than help users.

2.2 Medbot: Conversational artificial intelligence powered chatbot for delivering tele-health after covid-19

One of the major challenges that India as a country faces is to cater to good quality and affordable healthcare to its growing population.[3] The World Health Report issued by WHO has ranked India's healthcare system at 112 out of 190 countries. This inaccessibility of healthcare facilities especially in rural India and the intricacy in accessing means of transport further causes patients to postpone their treatment, or opt for medical facilities that may be closer but at the same time are not cost-efficient and well-matched to their medical needs. To seek more efficient ways to provide timely medical care, access and quality treatment to the patient, the role of Telemedicine comes into play which connects patients with healthcare providers and healthcare information.

Due to the recent "COVID-19" pandemic, social distancing will stay in India for a long time, especially for patients with chronic diseases, thereby imposing a hindrance for the population to access healthcare facilities.

The data released by the National Health Mission, amid COVID-19 shows that there has been a fall in other acute illnesses being reported during the lockdown in India. This data indicates that a reduced hospitalization case indicates a lack of access to healthcare, rather than a lack of illness. In this alarming situation, telemedicine acts as a boon for people. By using conversational artificial intelligence, healthcare providers can diagnose and treat patients without the need for a personal visit, whilst promoting social distancing and reducing the risk of COVID-19 transmission.

Tele-Health is the distribution of health-related services via electronic and telecommunication technologies. It enables long-distance patients to get care, advice, reminders, education, monitoring, and remote admissions from clinicians. A chatbot is a conversational agent that

communicates with users using natural language. Though there exist some applications that serve as virtual healthcare consultants, none of them provides generic healthcare information, preventive measures, home remedies, and consultation for India-specific context with multi-lingual support. India being a country with a diverse population speaking different languages, access to healthcare at present has multiple barriers including language, lack of healthcare professionals, and lack of access to facilities in rural and remote areas and costs associated with medical consultation. Therefore, an application "Aapka Chikitsak" is developed to provide users healthcare consultation, counseling and information with multi-lingual support (for now, English and Hindi) to improve the healthcare and well-being of the growing population in India and continue provision of healthcare access at ease post the lockdown as well.

In recent years, serverless architectures (Functions-as-a-Service) are gaining traction as an alternative way of providing backend services without requiring a dedicated infrastructure. Serverless allows its users to deploy their stateless functions into platform infrastructures. This stateless behavior makes every invocation independent of the previous runs. For our application, Firebase Cloud Functions and Google Cloud Platform as our backend infrastructure is provided.[9]

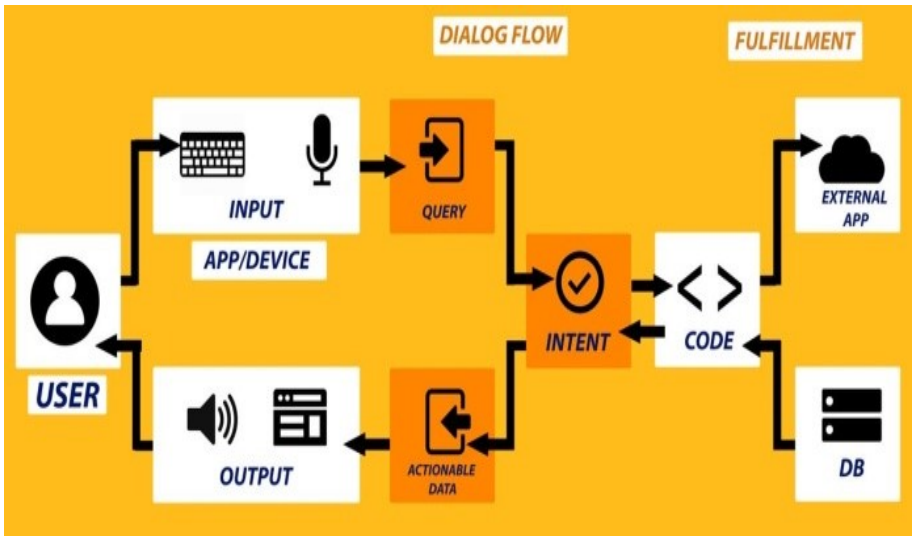


Figure 2.2: The Architecture of the Conversational Bot

The conversation of our bot has been framed and designed in a way to mimic human behavior to develop a user-friendly chat system that allows them to feel at ease, overcoming the bias of machine interaction. While designing the conversational flows, multitudinous utterances that can be used by the users while interacting with the chat system is considered. Our application uses the Dialogflow Conversation API for creating an automated conversational chat system to hold a conversation with the user by understanding the natural language. Dialogflow employs Google's ML expertise, its products like Google Cloud Speech-to-Text and runs on GCP (Google Cloud Platform) that allows scaling of applications to hundreds

of millions of users. When a user chats with our application, there can be a broad variety of utterances that they can utilize aiming at the same purpose. In particular, Dialogflow provides a high-level dialogue flow to identify user queries by mapping them to intents that have been trained with an extensive amount of training phrases pool. Training phrases are certain model phrases that users would use, referred to as end-user expressions. An Intent is a collection or a category of such related end-user expressions during one conversation turn. In our conversational design, it created 255 intents and each of them has been trained with multiple user utterances that were collected during user-testing. By recognizing the intent of the query, Dialogflow chooses the specific actions to fulfill the intent. Entities in Dialogflow are meaningful sequences of characters or lexemes. It is used to extract specific keywords in a user's query. Entities allow us to map the synonyms to its reference value. A typical voice-user interface (VUI) makes possible spoken human chat with computers via speech recognition to understand spoken words and answer questions, and specifically uses text to speech to create a reply. Speech Synthesis Markup Language (SSML) is used to make the voice experience more interactive and robust. Several points are kept while designing the voice user interface of our conversational system. Our bot has its personality so that it sounds more human-like, convincing, and user-friendly. The flow of the conversation consists of short and concise replies in layman language that aids in saving time and makes the interaction more human-like. It includes various health tips and suggestions to increase interactivity and encourage greater engagement. Our bot is devised with multilingual support which has aided us to target both rural and urban areas in India. It structures the VUI of our conversational bot in such a way, to give responses with the focus on having a doctor-patient-like conversation. Our bot has its naturalness as it duly greets users and is trained to give suggestions, health tips, location-based food recommendations according to the patient's concerns and queries.

- Providing information about the majority of prevalent diseases in India along with their possible symptoms and applicable preventive measures.
- Home remedies and quick remedies for common illnesses.
- Interactive counseling sessions for emotional support and expectant mothers.
- Local food recommendations and recommended diet depending upon the geographic location of the user.
- Vaccine administration information.

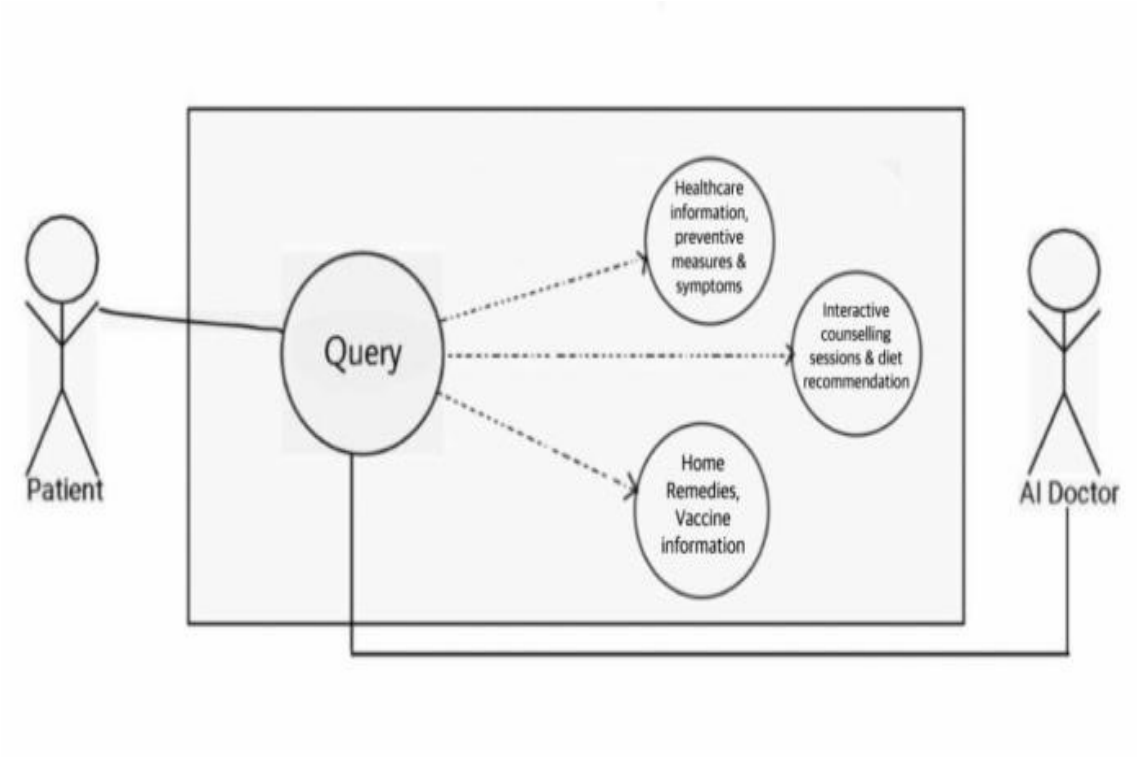


Figure 2.3: UML diagram showing the use cases of the Bot

2.3 Compassion Driven Conversational Chatbot Aimed for better Mental Health

In 2017, there were 197·3 million people with mental disorders in India, comprising 14·3% of the total population of the country. Mental disorders contributed 4·7% of the total DALYs in India in 2017, compared with 2·5% in 1990.[8] Mental illnesses are health conditions that involve change in thinking through one's behavior and emotion. These illnesses are associated with stress and/or problems due to ongoing work, financial, society or family activities. As reported latest in 2014, the number of mental health professionals in India was as low as "one in 100,000 people". The average suicide rate in India is 10.9 for every lakh people and the majority of people who commit suicide are below 44 years of age. Mental illness, also called mental health disorders, refers to a wide range of mental health conditions — disorders that affect your mood, thinking and behavior. Mental illness is treatable and the stigma around needs to be stopped. Many people suffering from mental illness do not wish to talk about it due to fear of being judged and treated differently. But mental illness is nothing to be ashamed of. Mental health problems include anxiety, depression, OCD, Panic attacks, bipolar disorder, DAD, loneliness, paranoia, PTSD, PMDD etc. and the list goes on. Bottomline is that a large number of population is getting affected due to mental health issues and this needs to be curbed right now and with lack of doctors and not many resources for people how do you take care of this problem.

In this regard there have been applications like Wysa chatbot which is one of a kind chatbot which learns every user's experience and builds reports accordingly. Also the Pacifica app also designed to target anxiety, stress through Cognitive Behavioural Therapy and mindfulness.[2] In this paper, the stigma around mental health and how the chatbot is designed to combat situations is discussed . The design and methodology has been discussed for how the chatbot will be implemented and what differentiates it from previously used chatbots in healthcare. Henceforth, the paper also discusses the architecture for the same and how the user responses are understood and then accordingly attended by the bot

2.3.1 Buddy : The 3AM Friend

The chatbot "Buddy" is a compassion-driven AI chatbot that works as your 3AM friend. It is an artificial intelligence based mobile chatbot app for initiating positive conversations, building mental resilience and focuses on improving mental well-being using a text based interface of its own. It has the ability of holding conversations just like any of your friend would and therefore helps in times when people often feel isolated. Instead of rule based or retrieval based , this chatbot is designed by generative based learning that there is no set of questions or rules that the chatbot will definitely ask , it learns from the older conversations and is capable of answering the conversations which it encounters very first time. This application drives the conversation towards a positive scope and tries to understand and assist the user in handling the situations of stress better. The application also helps them develop self-expression. The main benefit of the application is that it entirely free and available to you 24x7 so you always have someone to share your problems with.

The approach picked for this chatbot uses Deep Learning and Natural Language processing for generating and understanding responses , i.e generative approach.

- **Using Natural Language Understanding and Natural Language Generation**

  The very first task in hand is to understand the intent of the user for which the famous natural language understanding concept has been utilized. The job of NLU unit is to transform the user utterance to some semantic format that can be easily understood by the system which includes two very important tasks : Slot filling and Intent detection. The job of filling the slots and detecting the intent of user is seen as a sequence tagging problem. Just for this reason,the LSTM based recurrent neural network have been used for the implementation of NLU unit. Natural Language Generation is the process of generating text from a meaning representation. For tasks like machine translation, text summarization and dialog systems, NLG systems provide a much critical role. Chatbots that use the rule-based NLG , the outputs are predefined template statements for a given frame thus have limited response without adding anything of its own.

- **Generative Based Approach**

  The generative approach used here doesn't use any predefined templates or sentences rather uses deep learning. Therefore unlike the usual retrieval based where we translate from one language to another, here we translate from an input to an output. Seq2Seq model framework most suited for this type of chatbot eliminates the dialog problem faced in rule-based or retrieval-based approaches.

  With the aid of deep-learning, generative based has yielded great results. The Sequence-to -Sequence model architecture has two RNNs with different sets of arguments or parameters. This particular approach consists of 2 RNNs, the first one is used to encode the input sequence , what we call as the encoder, and the second one is used to decode the encoded input sequence into the output or target sequence, known as the decoder. This idea has provided great results for text summarization and conversation based on questions and answers where the sequence of words matter and the system needs to remember the significance of ordering.

- **Data Processing**

  The data pre-processing is a key part of the entire process because making the data ready for modeling plays important role on how efficient the system will turn out to be. Reading the words, converting to index of each word, word to index dictionaries, filter too long and too short sequences, bag of words, array dealt with it, tokenizing with "End of sentence" and "Start of sentence" to make sentences more clear are all done in pre-processing phase. The data pre-processing has changed it from raw lines of conversations to zero-padded numpy arrays of indices, and further required dictionaries of word2index and unit index2word to keep counts. It is important to note here that the LSTM takes only batches of inputs and not a single input at a time. With a few helper functions, we gather random examples from dataset, in batches and feed to model. The most important class Seq2Seq that does all the training, evaluation, graph building, saving model takes several parameters as input like batch size, num-layers, emb-dim, learning rate, epochs etc.

  The LSTM cell that is the most important part of the entire model has a keep-prob placeholder which is used to control dropout rate and prevent overfitting. The basic LSTM cell used is wrapped with a Dropout Wrapper. The LSTM cells here are stacked LSTM architecture.Next we used a function embedded-rnn-seq2seq from seq2seq module, to create a seq2seq model, which does word embedding internally. For testing also, we use the same model that has almost the same parameters as training but with the feed previous switch enabled. Thereby the decoder has to use the output of previous step as input to its present timestep, while during training, the input to decoder is

taken from real output sequence. The function sequence loss has been used to get the expression for loss. The next step is training with a dropout selected as 0.5. The data was split into training, testing and validation and further the results were analyzed.

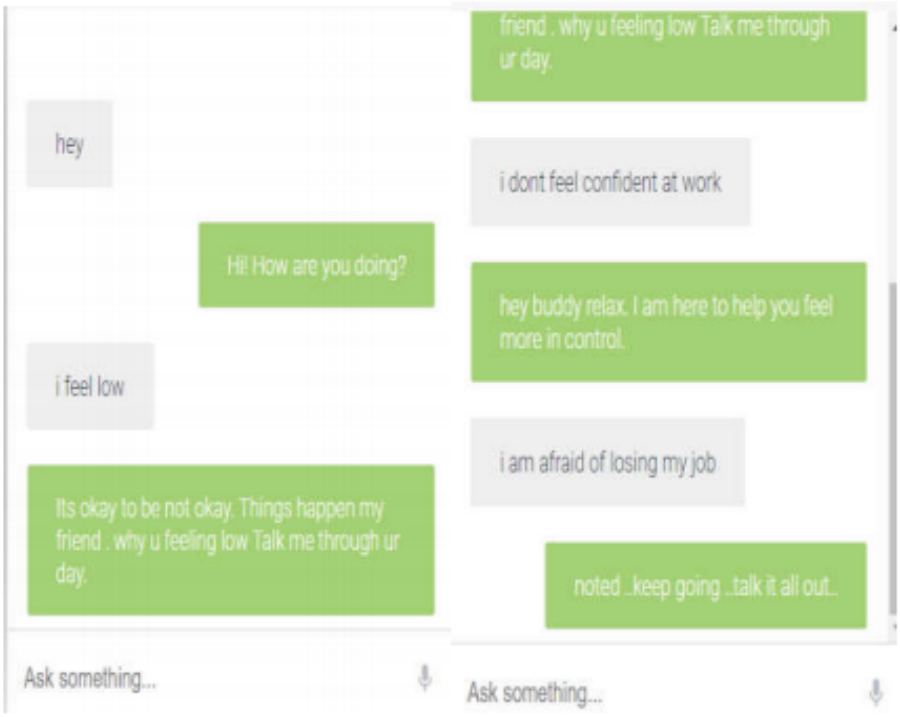


Figure 2.4: Sample images of how chatbot “Buddy” holds up conversation and interacts.

2.4 Chatbot for Disease Prediction and Treatment Recommendation using Machine Learning

A prosperous society is when its entire people are healthy. It is important to maintain the health if one wishes to be happy. Only a healthy body can have a healthy mind and it has a positive impact on the performance of people. Nowadays, people are less aware of their health. In their busy life, they forget to take suitable measures to maintain their health and are less aware of their health status. Most people comprising the working section of the society claim that their hectic schedule gives them no time for periodic medical check-ups and that they disregard any uneasiness shown by their body until it is too severe. Hospitals are the most widely used means by which a sick person gets medical check-ups, disease diagnosis and treatment recommendation. This has been a practice by almost all the people over the world. People consider it as the most reliable means to check their health status.

The proposed system[4] creates an alternative to this conventional method of visiting a hospital and making an appointment with a doctor to get diagnosis. A medical chatbot is built to be a conversational agent that motivates users to discuss about their health issues and based

on the symptoms provided by them; chatbot returns the diagnosis. This chatbot system will be able to identify symptoms from user interaction. Using these extracted symptoms, chatbot predicts the disease and recommends treatment. The machine learning algorithm employed here is K-nearest neighbor algorithm (KNN). This clearly shows that a medical chatbot can somewhat accurately diagnose patients with simple symptom analysis and a conversational approach done with the help of natural language processing.

The main objective of the proposed system is to have the importance of health in life reach out to people and encourage people to follow measures to maintain health by making the chatbot available to all. Chatbot and health have a history of working well together. It creates a good human-like conversational environment for interaction between the user and the system.

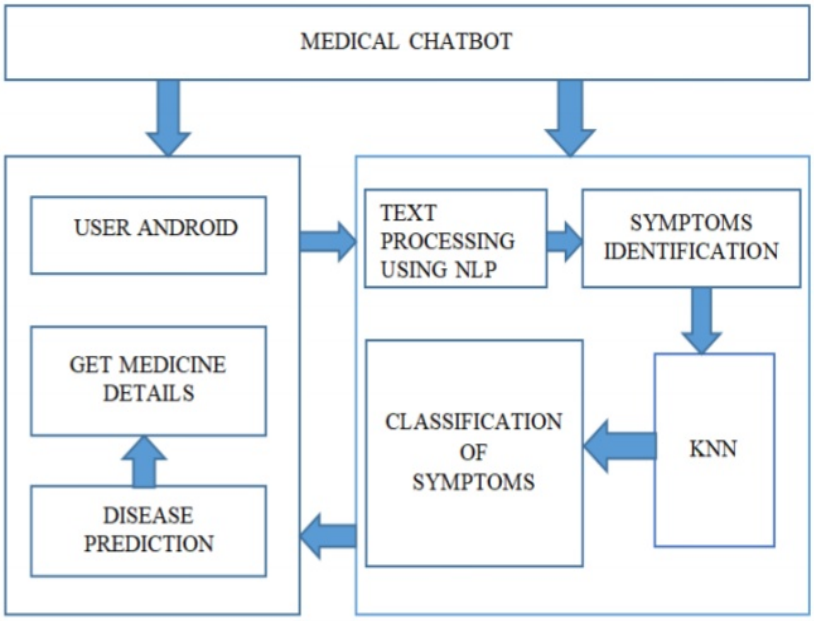


Figure 2.5: Working model of the proposed system

In this system, the user talks about their health and it is a great way for the users to regulate the healthy lifestyle. An important aspect of this system is that talking with a non-human entity provides a sense of security especially when it comes to mental health as it remains as a confidential meeting with the diagnosis being available only to the user. This system is meant to help and deliver immediate actions where humans cannot reach due to timing or budget as it is readily available and free of cost. It allows the user to have free medical check-up based on the symptoms where the user's health issue is easily identifiable.

The chatbot allows user to login to the system. User registers on chatbot application. They need to submit some personal details which will be confidential. User then interacts with the system and the words are recognized by the use of natural language processing and the system recognizes symptoms of the user. There is an admin who controls the chatbot application. The admin views the details of all the users and can even manually add, delete or update symptoms and diseases. The chatbot is trained on symptoms-disease dataset. From the symptoms identified by the user, KNN algorithm can predict the disease, depending on the dataset. The system recognizes the disease and finally recommends the suitable treatment needed for the same.

The chatbot helps to encourage patients to discuss about their medical issues and provides a suitable diagnosis and recommends treatment. Here the text processing is done using natural language processing. Initially the user enters their symptoms through text. It uses the machine learning algorithm KNN. KNN identifies the symptoms from the interaction with the user. KNN maps the symptoms to the particular disease and finally recommends the suitable treatment to the user.

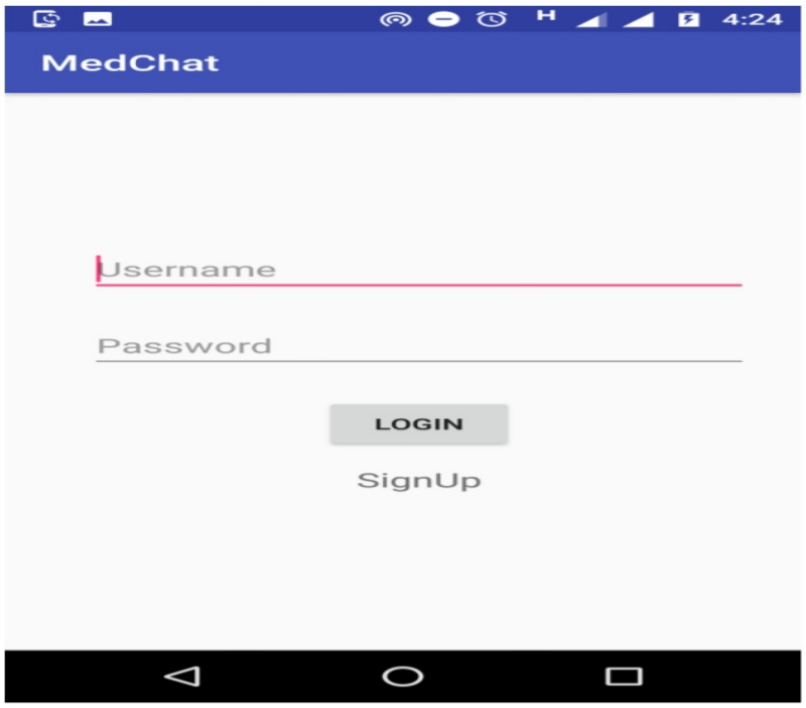


Figure 2.6: User login module

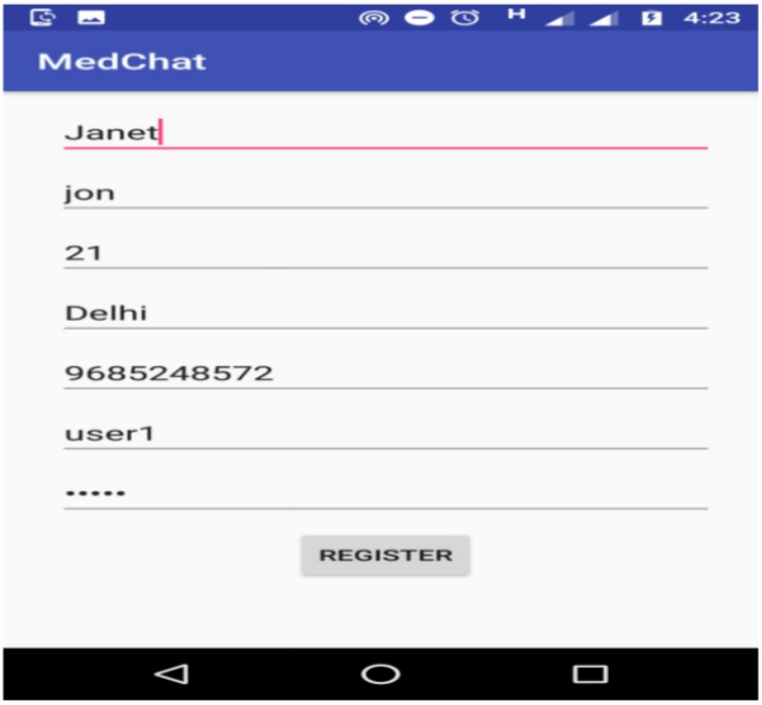


Figure 2.7: User register module

Chatbot can act as a doctor. The chatbot acts as a user application. The user of this application can specify their symptoms to the chatbot and in turn, chatbot will specify the health measures to be taken. General information about symptom and diseases are available in the dataset and thus the chatbot instance can provide information about disease and treatment to the user. After analyzing the symptoms of the different users, it finally predicts the disease to the user and provides with a link where details about the treatment is visible.A smart medical chatbot can be useful to patients by identifying the symptoms as described by them, giving proper diagnosis and providing with suitable treatment for the disease. In the busy life, it is rare for people to frequently visit hospitals for check-ups. Chatbot is of great importance in such situations as they provide diagnostic assistance with a single click of button. Chatbot doesnt require the help of any physician to give proper health measures to the users and this is one of the major advantages of chatbot. Moreover, the cost effectiveness in using chatbot is a major attractiveness to users. The chat with users is completely personal and this helps users to be more open with their health matters and paves way for chatbot to efficiently identify the disease.

CHAPTER 3

PROBLEM STATEMENT

The project ”QRTBOT - Chatbot facility for people in quarantine” aims to create a simple and efficient way for people to find out if they have a chance of being COVID-19 positive and also to provide a social outlet for people who are isolated due to quarantine. In contrast to existing chatbots, our motivation is to not only provide high detection rate of COVID-19 symptoms, but also to provide proper emotional support for people who are in need of it. To accomplish these tasks we make use of machine learning algorithms that can create a system that has the capability to perform symptom analysis as well as sentimental analysis. One of the main advantages of such a system built by machine learning is that the chatbot will work in an open domain, and can thus understand and respond to complex user inputs and queries, which allows us to have better user-chatbot interactions. Thus, we can reduce close contact between people in public by instead encouraging the use of online resources such as QRTBOT.

CHAPTER 4

PROJECT MANAGEMENT

4.1 Introduction

Project management is the discipline of planning, organizing, securing, managing, leading, and controlling resources to achieve specific goals. A project is a temporary endeavor with a defined beginning and end (usually time-constrained, and often constrained by funding or deliverables), undertaken to meet unique goals and objectives, typically to bring about beneficial change or added value. The temporary nature of projects stands in contrast with business as usual (or operations), which are repetitive, permanent, or semi-permanent functional activities to produce products or services. In practice, the management of these two systems is often quite different, and as such requires the development of distinct technical skills and management strategies.

In our project we are following the typical development phases of an engineering project

1. Initiation
2. Planning and Design
3. Execution and Construction
4. Monitoring and Controlling Systems
5. Completion

4.1.1 Initiation

The initiating processes determine the nature and scope of the project. The initiating stage should include a plan that encompasses the following areas :

1. Analysing the business needs/requirements in measurable goals
2. Reviewing of the current operations
3. Financial analysis of the costs and benefits including a budget
4. Stakeholder analysis, including users, and support personal for the project
5. Project charter including costs, tasks, deliverables, and schedule

4.1.2 Planing and design

After the initiation stage, the project is planned to an appropriate level of detail (see example of a flow-chart). The main purpose is to plan time, cost and resources adequately to estimate the work needed and to effectively manage risk during project execution. As with the initiation process, a failure to adequately plan greatly reduces the project's chances of successfully accomplishing its goals.

- Determining how to plan
- Developing the scope statement
- Selecting the planning team
- Identifying deliverables and creating the work breakdown structure
- Identifying the activities needed to complete those deliverables
- Developing the schedule
- Risk planning

4.1.3 Execution

Executing consists of the processes used to complete the work defined in the project plan to accomplish the project's requirements. The execution process involves coordinating people and resources, as well as integrating and performing the activities of the project in accordance with the project management plan. The deliverables are produced as outputs from the processes performed as defined in the project management plan and other frameworks that might be applicable to the type of project at hand.

4.1.4 Monitoring & controlling

Monitoring and controlling consists of those processes performed to observe project execution so that potential problems can be identified in a timely manner and corrective action can be taken, when necessary, to control the execution of the project. The key benefit is that project performance is observed and measured regularly to identify variances from the project management plan.

4.2 System Development Life Cycle

The Systems development life cycle (SDLC), or Software development process in systems engineering, information systems, and software engineering, is a process of creating or altering information systems, and the models and methodologies that people use to develop these systems. In software engineering, the SDLC concept underpins many kinds of software

development methodologies. These methodologies form the framework for planning and controlling the creation of an information system.

The SDLC phases serve as a programmatic guide to project activity and provide a flexible but consistent way to conduct projects to a depth matching the scope of the project. Each of the SDLC phase objectives is described in this section with key deliverables, a description of recommended tasks, and a summary of related control objectives for effective management. The project manager must establish and monitor control objectives during each SDLC phase while executing projects. Control objectives help to provide a clear statement of the desired result or purpose and should be used throughout the entire SDLC process.

4.2.1 Spiral Model

We have used the Spiral model in our project. The Spiral model incorporates the best characteristics of both- waterfall and prototyping model. In addition, the Spiral model also contains a new component called Risk Analysis, which is not there in the waterfall and prototype model. In the Spiral model, the basic structure of the software product is developed first. After the basic structure is developed, new features such as user interface and data administration are added to the existing software product. This functionality of the Spiral model is similar to a spiral where the circles of the spiral increase in diameter. Each circle represents a more complete version of the software product. The spiral is a risk-reduction oriented model that breaks a software project up into main projects, each addressing one or major risks. After major risks have been addressed the spiral model terminates as a waterfall model. Spiral iteration involves six steps:

1. Determine objectives, alternatives and constraints.
2. Identify and resolve risks.
3. Evaluate alternatives.
4. Develop the deliverables for the iteration and verify that they are correct.
5. Plan the next iteration.
6. Commit to an approach for the next iteration.

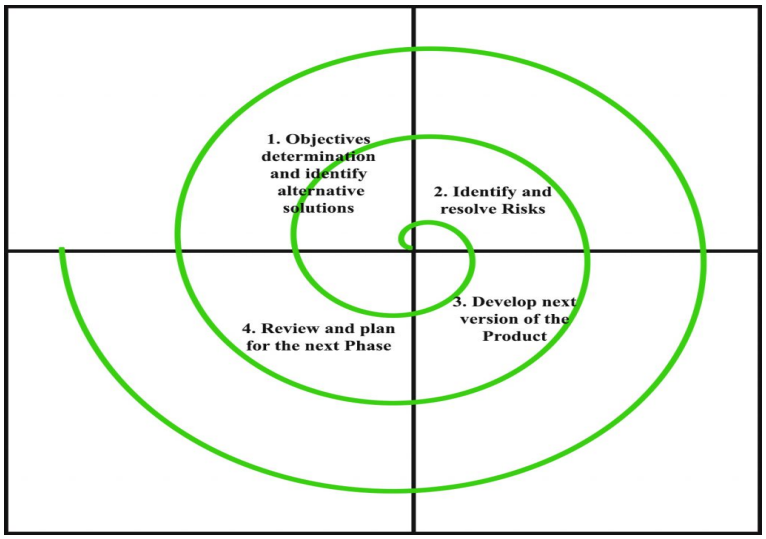


Figure 4.1: Spiral Model

CHAPTER 5

METHODOLOGY

"QRT Chatbot" is a simple and efficient way for people to find out if they have a chance of being COVID-19 positive and also to provide a social outlet for people who are isolated due to quarantine. In contrast to existing chatbots, our motivation is to not only provide high detection rate of COVID-19 symptoms, but also to provide proper emotional support for people who are in need of it. The user can get clear informations about everything related to COVID-19 and also the precautions that he/she should take during this pandemic. To accomplish these tasks we make use of machine learning algorithms that can create a system that has the capability to perform symptom analysis as well as sentimental analysis.

Project consist of mainly two modules. Symptom Prediction and Emotion Analysis. QRT BOT, however, is not intended to be used for diagnosis or treatment purposes. The questions walk users through symptoms and then suggests whether they have to take a COVID test or not. User registers on chatbot application. They need to submit some personal details which will be confidential. User then interacts with the system and the words are recognized by the use of natural language processing and the system recognizes symptoms of the user. The main algorithm used for Symptom Prediction is K-Nearest Neighbors (KNN) algorithm. It is the nonparametric algorithm. The learning and prediction analysis is performed based on the given problem or dataset. The KNN classification model, the prediction is purely based on neighbor data values without any assumption on the dataset. The user can also reduce their stress and mental strain by having a chat with our chatbot. Our chatbot analyses the current emotion through which the user is going on and acts accordingly. This task is done with the help of Term Frequency-Inverse Document Frequency(TF-IDF) algorithm and Valence Aware Dictionary and Sentiment Reasoner (VADER). Sentiment analysis helps a chatbot to understand the emotions and state of mind of the users by analyzing their input text or voice.

HTML 5 Validation

Before submitting data to the server, it is important to ensure all required form controls are filled out, in the correct format. This is called client-side form validation, and helps ensure data submitted matches the requirements set forth in the various form controls. Client-side validation is an initial check and an important feature of good user experience; by catching invalid data on the client-side, the user can fix it straight away.

If it gets to the server and is then rejected, a noticeable delay is caused by a round trip to the server and then back to the client-side to tell the user to fix their data. You'll get messages such as:

- "This field is required" (You can't leave this field blank).
- "Please enter your phone number in the format xxx-xxxx" (A specific data format is required for it to be considered valid).
- "Please enter a valid email address" (the data you entered is not in the right format).
- "Your password needs to be between 8 and 30 characters long and contain one uppercase letter, one symbol, and a number." (A very specific data format is required for your data)

When you enter data, the browser and/or the web server will check to see that the data is in the correct format and within the constraints set by the application. If the information is correctly formatted, the application allows the data to be submitted to the server and saved in a database; if the information isn't correctly formatted, it gives the user an error message explaining what needs to be corrected, and lets them try again

5.1 Modules

5.1.1 SYMPTOM PREDICTION

This system requires the need for an algorithm that can help in predicting whether to take a test as entered by the user. The system uses a machine learning algorithm. For the chatbot to provide the user with a response to the messages received, it needs KNN.

K-Nearest Neighbors (KNN) algorithm

The K-nearest neighbors (KNN) algorithm is a type of supervised machine learning algorithms. KNN is extremely easy to implement in its most basic form, and yet performs quite complex classification tasks. It is a lazy learning algorithm since it doesn't have a specialized training phase. Rather, it uses all of the data for training while classifying a new data point or instance. KNN is a non-parametric learning algorithm, which means that it doesn't assume anything about the underlying data. This is an extremely useful feature since most of the real world data doesn't really follow any theoretical assumption e.g. linear-separability, uniform distribution, etc. In KNN, the 'K' represents the number of nearest neighbor data values. Based on 'K', i.e., the number of nearest neighbors, the decision is made by the KNN algorithm on classifying the given dataset.

Data Acquisition Module:

The Data Acquisition Module's mission is to collect the user's confidential data and create a dataset that contains these user's information:

- First Name
- Last Name
- Phone Number
- Email
- Age
- Gender
- Username
- Password
- Chronic diseases (Alzheimer disease and dementia, Arthritis, Asthma, Heart disease, Cancer, Diabetes, etc) and disease duration

When the user sends an input message, COVID-Chatbot must transform unstructured text to a structured representation composed of entities and intents which called the natural language processing (NLP), through several successive steps such as Tokenization , Training data, Testing data, etc. And finally, intent classification is done using K-Nearest Neighbors (KNN) algorithm because the algorithm requires no training before making predictions, new data can be added seamlessly.

Data Preprocessing Module:

An efficient model is developed through intensive training by providing a large number of datasets. We import the dataset and all the nessecary libraries to work with our dataset. The step of data preprocessing plays a very important role in contributing to the accuracy of any training model.

1. **Train Test Split**:

   To avoid over-fitting, we will divide our dataset into training and test splits, which gives us a better idea as to how our algorithm performed during the testing phase. This way our algorithm is tested on un-seen data, as it would be in a production application. We splits the dataset into 80% train data and 20% test data.

2. **Feature Scaling**:

   Before making any actual predictions, it is always a good practice to scale the features so that all of them can be uniformly evaluated. Since the range of values of raw data varies widely, in some machine learning algorithms, objective functions will not work properly without normalization. Therefore, the range of all features should be normalized so that each feature contributes approximately proportionately to the final distance.

3. **Training and Predictions**:

   We train the KNN algorithm and make predictions with it. The first step is to import the KNeighborsClassifier class from the sklearn.neighbors library. Then this class is initialized with one parameter, i.e. n_neigbours. This is basically the value for the K. There is no ideal value for K and it is selected after testing and evaluation, however to start out, 4 is chosen as value for KNN algorithm. The final step is to make predictions on our test data.

4. **Evaluating the Algorithm**:

   For evaluating an algorithm, confusion matrix, precision, recall and f1 score are the most commonly used metrics. The confusion_matrix and classification_report methods of the sklearn.metrics can be used to calculate these metrics.

5. **Tokenization**:

   The key functionality of tokenization is to convert the text received from the user to tokens. Tokenization occurs in such a way that different words will be converted to different tokens. These tokens are then used as the input for other types of analysis or tasks, like parsing. After forming the tokens from the words, they will undergo lemmatization and stemming. Stemming and lemmatization are performed to make text processing easier. They represent the key steps in natural language processing. Stemming follows rule-based method. It looks upon prefixes or suffixes in tokens and they are removed. This is how it reduces words to its root form. When the words are reduced to its root form, a bag of words is later formed. Similarly the dataset contents which include symptoms and diseases are also processed in the same way and converted to bag of words. This is a vector comprising of 0s and 1s and each of them will be given an answer id.

Identification and Classification Module:

This module identifies and classifies the data given. The binary classification is used for predicting whether the user is contracted with COVID. In binary classification, it identifies

the symptoms and predicts the severity of disease. The user chats with the chatbot application in the same way one chats with other humans. It is a website where the user first login to the system, once registered. It is through this chat, the chatbot finds the symptoms of the user by asking the user a set of 14 questions. For this to happen smoothly, the chatbot will be trained with a dataset containing COVID symptoms. When the user sends messages, text processing will be done. Text processing is done using natural language processing (NLP). NLP makes human to communicate with the machine easily. When a response is received, chatbot tries to converge it that available in the dataset which it is already trained on and it will be one among the k nearest neighbors. This is a vector comprising of 0s and 1s and each of them will be given an answer id. This is done with the help of KNN. If the user is identified with no issues, the chatbot rechecks the symptoms. This is done by calculating a score where the score depends upon the sum of 1s in the list(user's responses) divided by 14 and also refers to the Health Status Update page and checks if it contains any entries which refers to the user's medical history. If the score is greater than the threshold value 0.5 and if the user has updated any chronic disease, then the user has a chance of having COVID-19 otherwise not.

5.1.2 EMOTION ANALYSIS

While working in a sentiment analysis program, certain words are more important than others, and it is crucial to assign this importance. To address this, TF-IDF (Text Frequency - Inverse Document Frequency) is utilized. It is necessary to distinguish between positive, negative, and neutral sentiments, or even retrieve scores associated with a given opinion based only on text. We use VADER (Valence Aware Dictionary and Sentiment Reasoner) to analyse the emotion through which the user going from his chat replies.

TF-IDF (Text Frequency - Inverse Document Frequency)

TF-IDF is a statistical measure that evaluates how relevant a word is to a document in a collection of documents. This is done by multiplying two metrics: how many times a word appears in a document, and the inverse document frequency of the word across a set of documents. TF-IDF for a word in a document is calculated by multiplying two different metrics:

- The term frequency of a word in a document. There are several ways of calculating this frequency, with the simplest being a raw count of instances a word appears in a document. Then, there are ways to adjust the frequency, by length of a document, or by the raw frequency of the most frequent word in a document.
- The inverse document frequency of the word across a set of documents. This means, how common or rare a word is in the entire document set. The closer it is to 0, the more common a word is. This metric can be calculated by taking the total number of

documents, dividing it by the number of documents that contain a word, and calculating the logarithm.

Multiplying these two numbers results in the TF-IDF score of a word in a document. The higher the score, the more relevant that word is in that particular document.

VADER (Valence Aware Dictionary and sEntiment Reasoner)

VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. VADER uses a combination of A sentiment lexicon is a list of lexical features (e.g., words) which are generally labeled according to their semantic orientation as either positive or negative. VADER not only tells about the Positivity and Negativity score but also tells us about how positive or negative a sentiment is. is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. VADER uses a combination of A sentiment lexicon is a list of lexical features (e.g., words) which are generally labeled according to their semantic orientation as either positive or negative. VADER not only tells about the Positivity and Negativity score but also tells us about how positive or negative a sentiment is.

Reading the data:

The first step involves extracting the data from db.sql. The dataset used is from db.sql, which is a file containing a large number of user responses and the reply of the chatbot for responses with positive, neutral and negative emotion.

Text to Vector:

The user responses are split into words and these words are then kept in vector form. Similarly, the data in the dataset are also split into words and then kept in vector to check the sentence similarity using TF-IDF.

TF-IDF:

TF-IDF is used for assigning importance to the different words in the text based on the frequency of their occurrence in the text. The threshold value is 0.3. If the cosine similarity is greater than 0.3, the user response is similar to the one in the dataset otherwise not.

Vader_Lexicon:

VADER (Valence Aware Dictionary and Sentiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media, and works well on texts from other domains. Here, SentimentIntensityAnalyzer() is an object and polarity_scores is a method which will give us scores of the following categories:

- Positive

- Negative
- Neutral
- Compound

The compound score is the sum of positive, negative and neutral scores which is then normalized between -1 (most extreme negative) and +1 (most extreme positive). The more compound score closer to +1, the higher the positivity of the text and vice versa. Depending upon the emotions in user's response the chatbot replies.

5.2 System Requirements and Specifications

5.2.1 PyCharm

PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers.PyCharm is cross-platform, with Windows, macOS and Linux versions.In addition to Python, PyCharm provides first-class support for various Python web development frameworks, specific template languages, JavaScript, CoffeeScript, TypeScript, HTML/CSS, AngularJS, Node.js, and more.

5.2.2 Windows 10

Windows 10 is produced by Microsoft which consists of a series of personal computer operating systems and was created as part of its Windows NT family of operating systems. It is preceded by Windows 8.1, and was released on July 15, 2015 (to manufacturing), and on July 29, 2015(to retail).

5.2.3 Python 3.6.7

Python is the most popular dynamic object-oriented programming language that can be used for many kinds of software development. It offers strong support by providing a smooth platform for integration with other languages and tools, and also comes with extensive standard libraries. Its simplicity makes it possible to be learned in a few days. It encourages higher quality, more maintainable code.

5.2.4 SCIKIT learn

Scikit-learn is a platform that provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. The library is built upon and supported by the SciPy (Scientific Python) that must be installed before you can use sci-kit-learn.

5.2.5 Pandas

Pandas is a software library mainly used for data manipulation and analysis in Python. In particular, it offers data structures along with operations that involve manipulating numerical tables and time series.

5.2.6 NLTK

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming semantic reasoning, etc.NLTK is available for Windows, Mac OS X, and Linux. Best of all, NLTK is a free, open source, community-driven project.

5.2.7 Flask

Flask is a popular Python web framework, meaning it is a third-party Python library used for developing web applications.Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications.

5.2.8 SQLyog

SQLyog is a GUI tool for the RDBMS MySQL.SQLyog is a fast, easy to use and compact graphical tool for managing your MySQL databases.

5.2.9 PyMySQL

PyMySQL is an interface for connecting to a MySQL database server from Python. It implements the Python Database API v2. 0 and contains a pure-Python MySQL client library.

5.3 Architecture

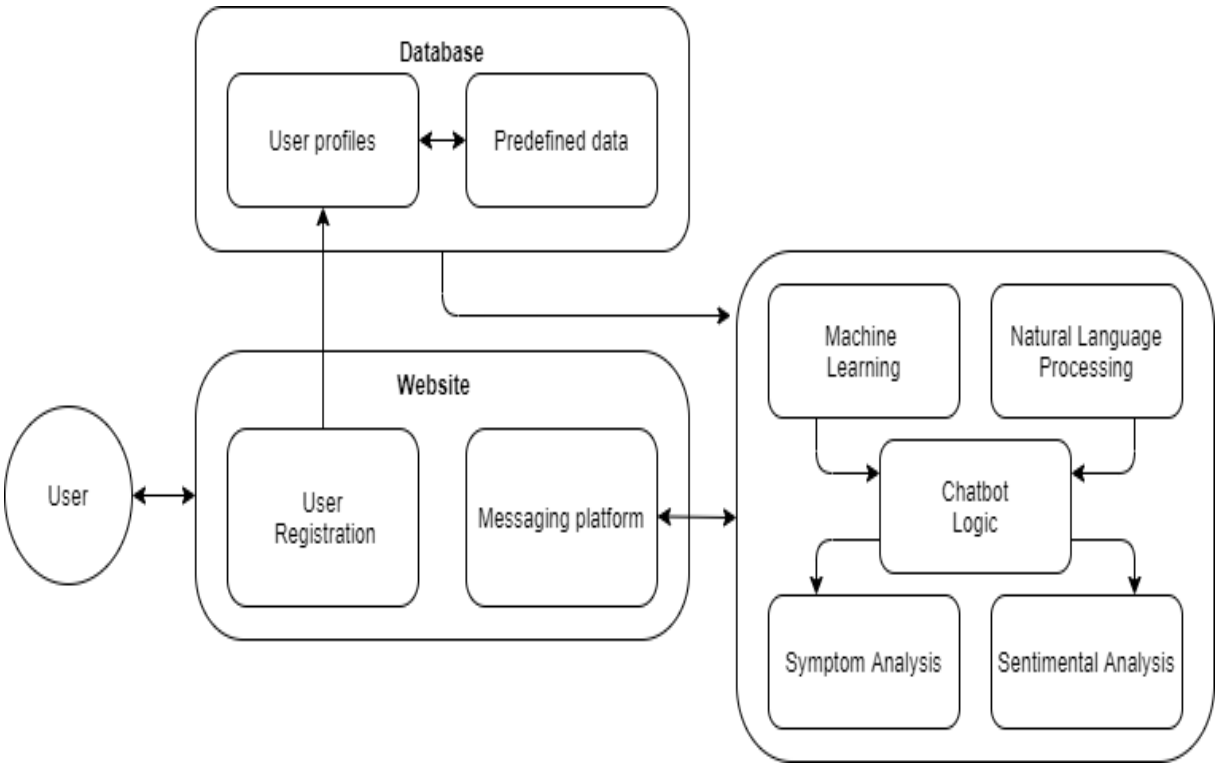


Figure 5.1: Chatbot Architecture

5.4 Data Flow Diagrams

5.4.1 DFD -Level 0

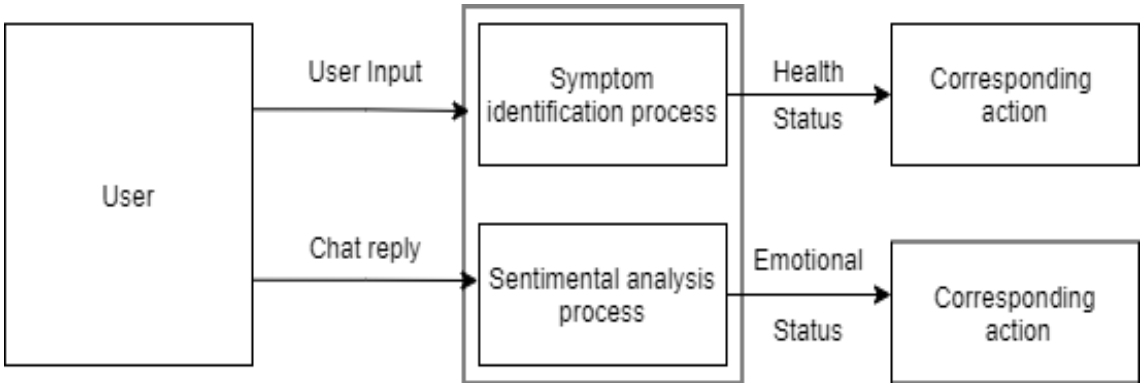


Figure 5.2: DFD Level-0

5.4.2 DFD Level-1 Module 1

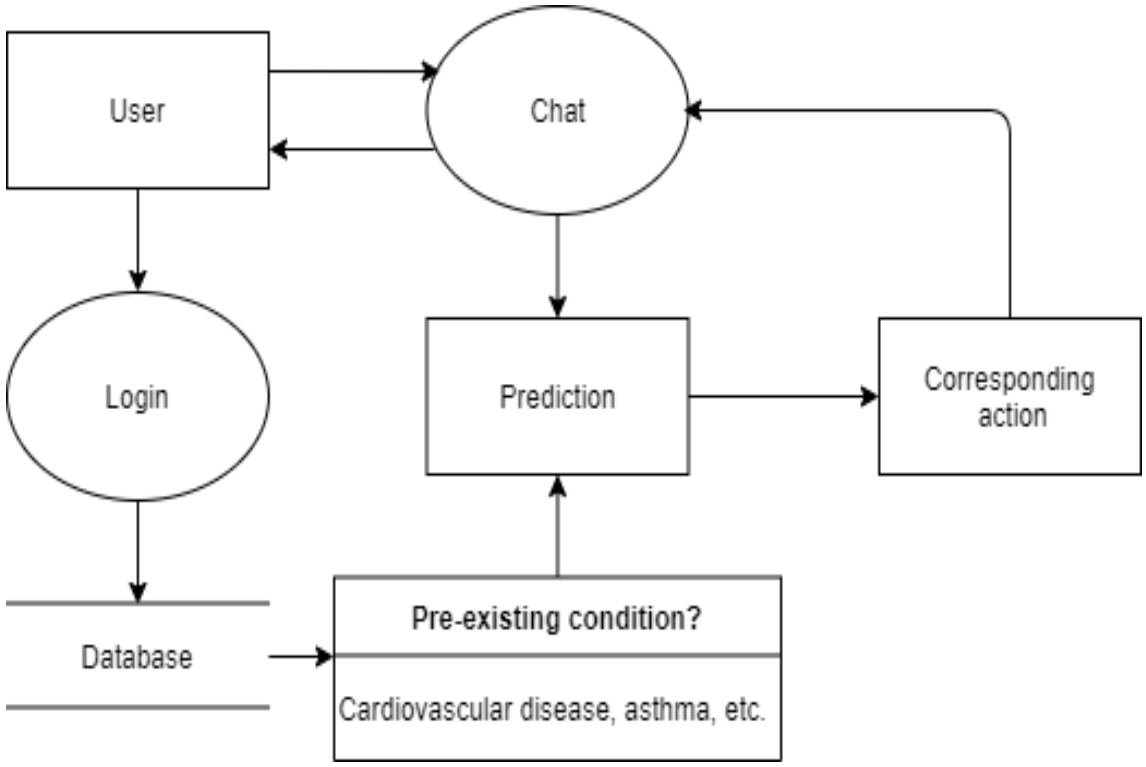


Figure 5.3: DFD Level-1 Module 1

5.4.3 DFD Level-1 Module 2

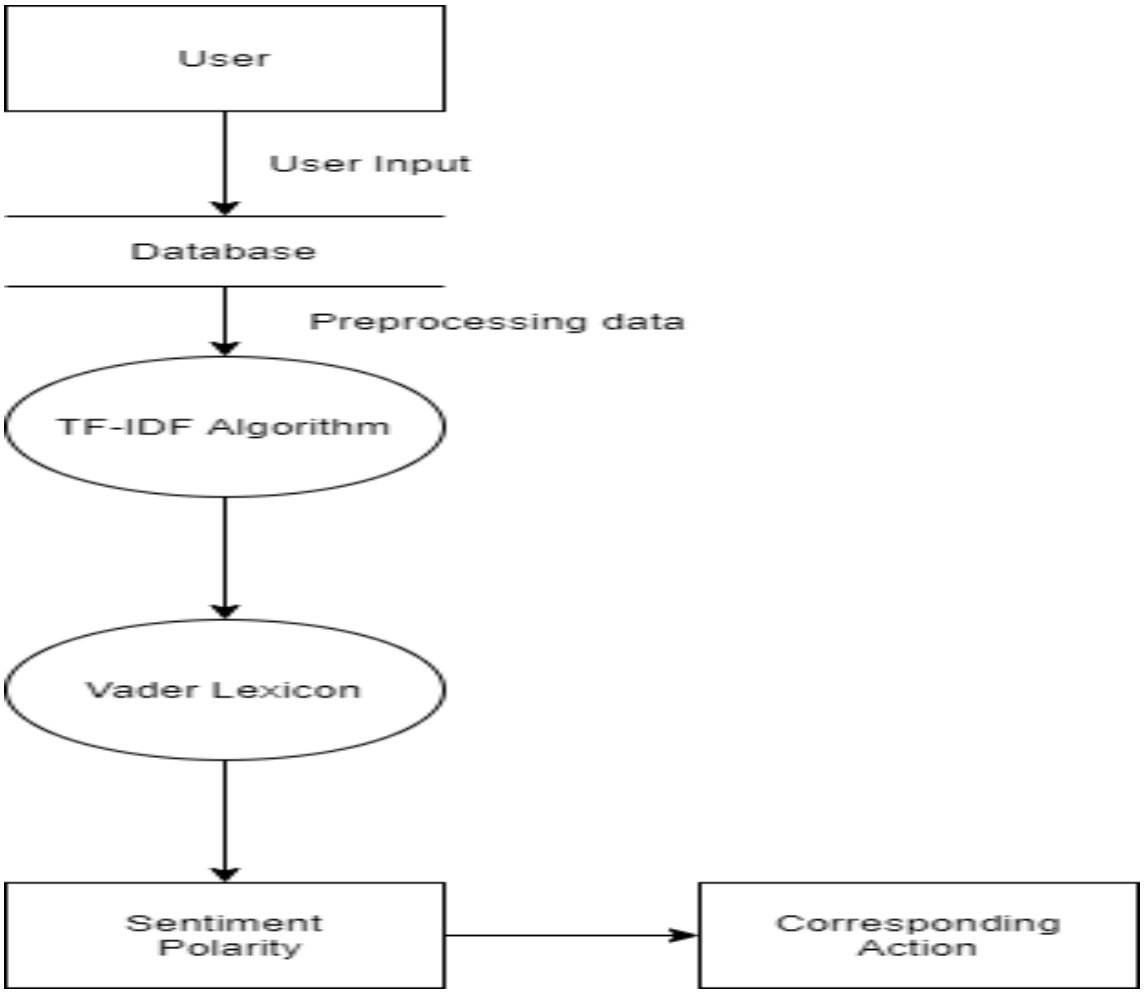


Figure 5.4: DFD Level-1 Module 2

5.4.4 DFD Level-2 Training

Figure 5.5: DFD Level-2 Training

5.4.5 DFD Level-2 Testing

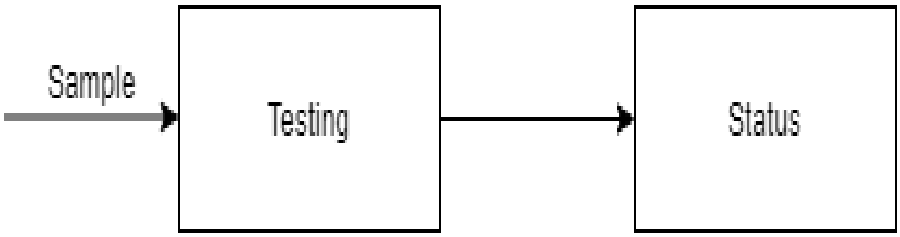


Figure 5.6: DFD Level-2 Testing

5.4.6 DFD Level-2 Emotion Prediction

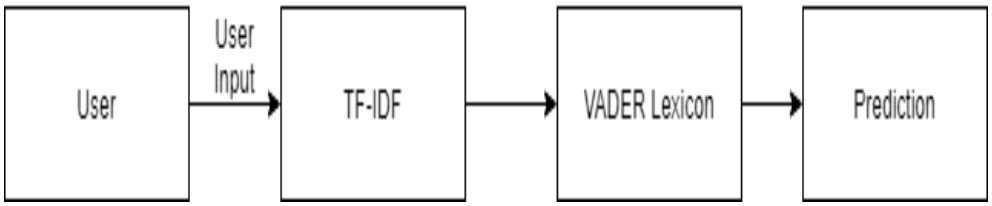


Figure 5.7: DFD- Level 2 Emotion Prediction

5.4.7 DFD Level-2 Symptom Prediction

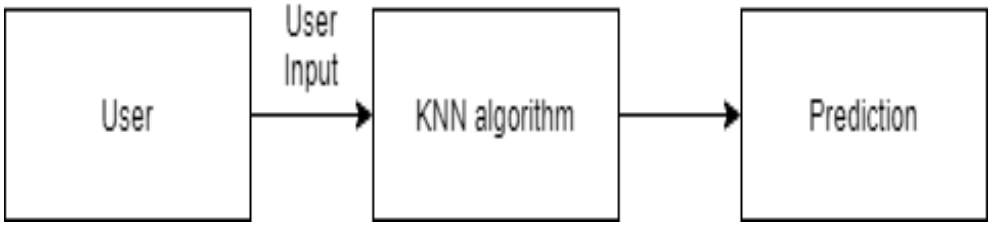


Figure 5.8: DFD- Level 2 Symptom Prediction

5.5 UML Diagram

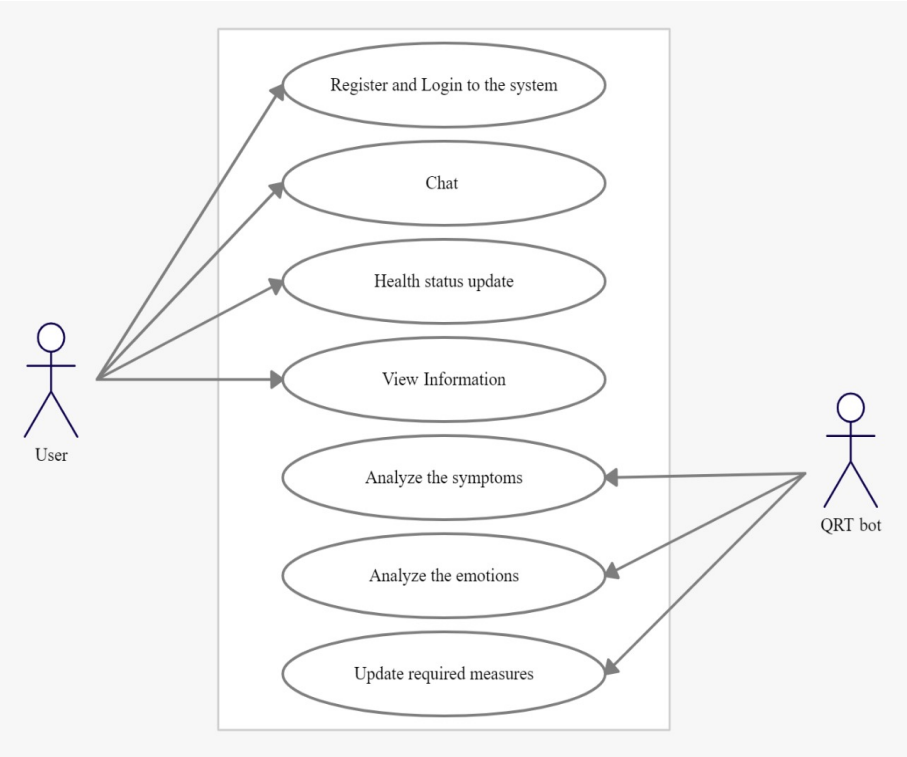


Figure 5.9: UML Interaction Diagram

5.6 Implementation

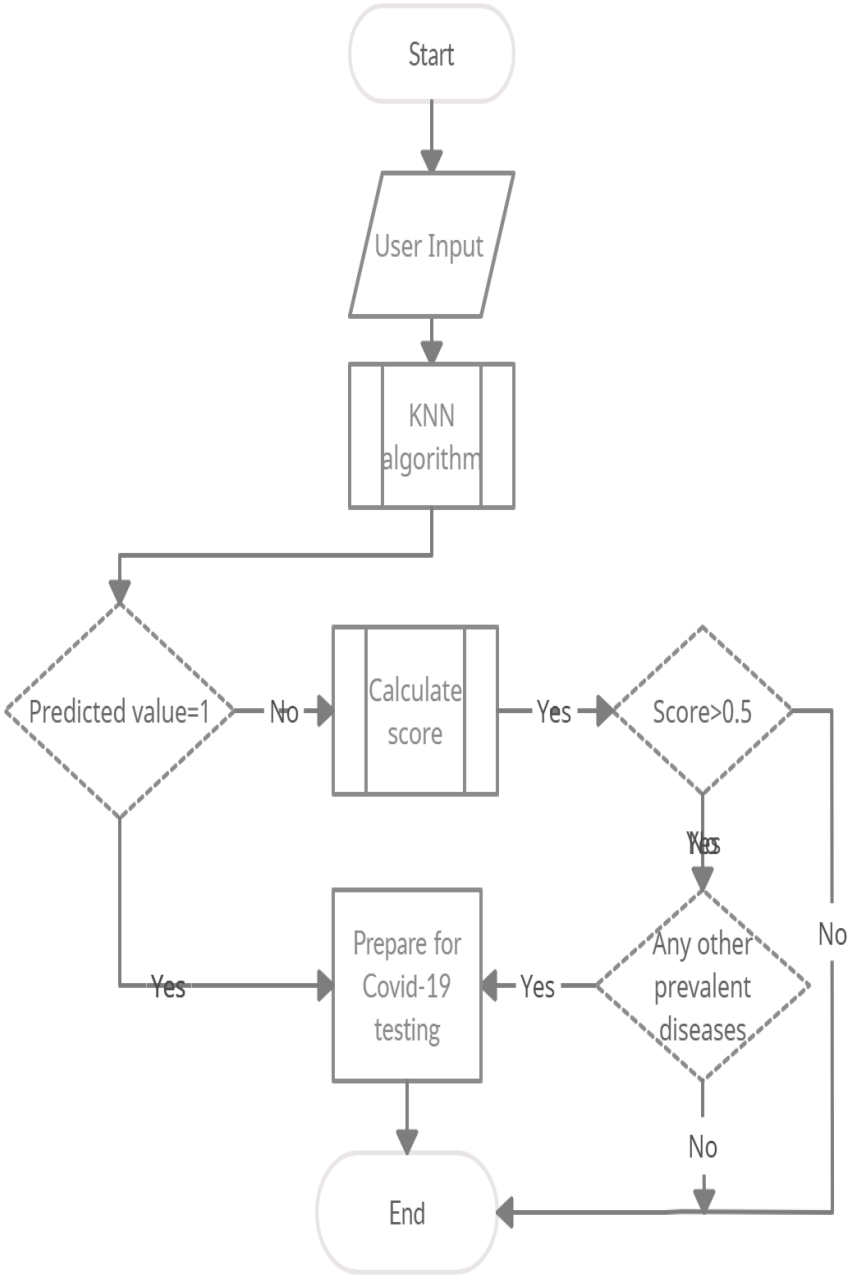


Figure 5.10: Proposed Implementation Flow Chart 1

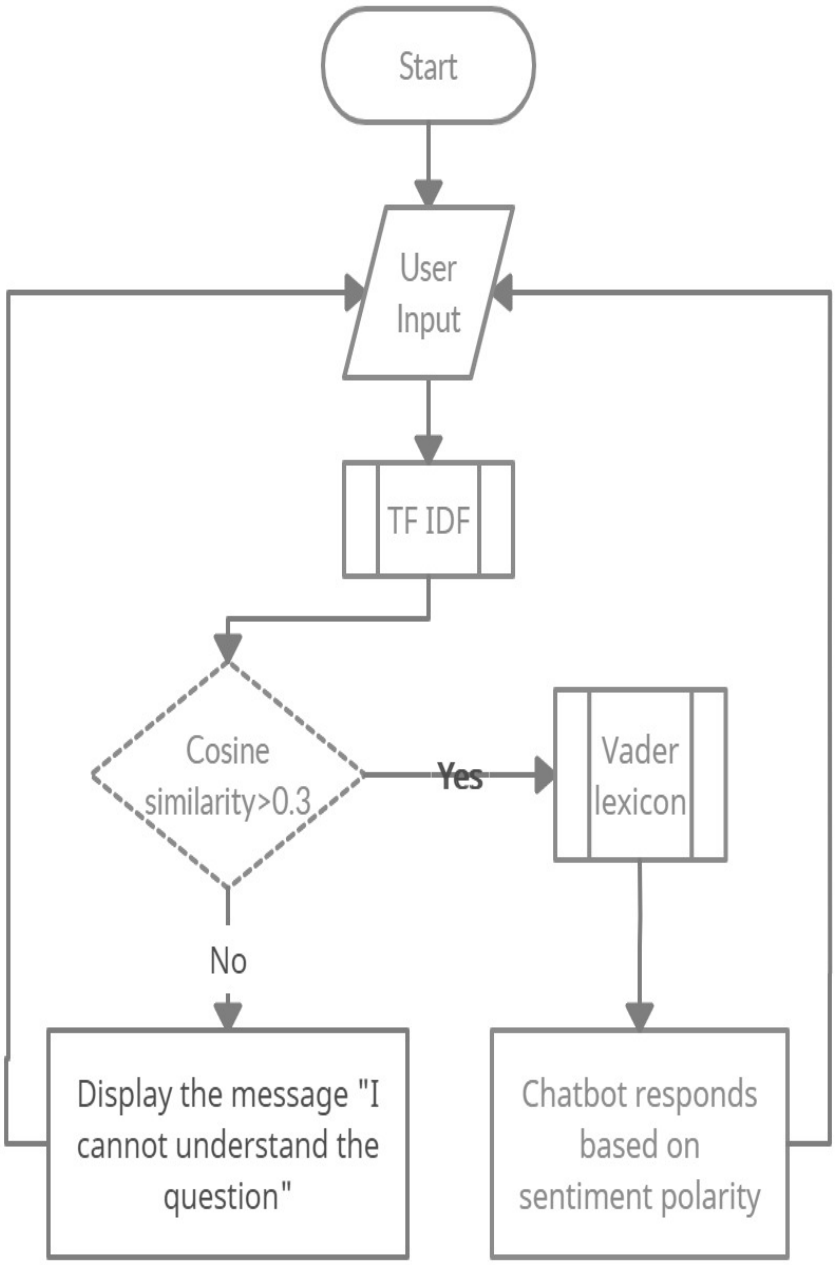


Figure 5.11: Proposed Implementation Flow Chart 2

SYMPTOM PREDICTION:

5.6.1 Start with PyCharm

PyCharm integrates with IPython Notebook, has an interactive Python console, and supports multiple scientific packages including Pandas, Matplotlib, Scikit-Learn, NumPy, etc.PyCharm provides smart code completion, code inspections, on-the-fly error highlighting and quick-fixes, along with automated code refactorings and rich navigation capabilities.

5.6.2 Import Libraries

Used libraries are listed below:

- numpy
- sklearn
- pandas
- matplotlib

5.6.3 Importing the Dataset

```
import pandas as pd
a=pd.read_csv('static/dataset/covid_19_result.csv')
```

5.6.4 Preprocessing

```
X = a.iloc[:, 0:15].values
y = a.iloc[:, 15].values
```

Splits the dataset into its attributes and labels.The X variable contains the attributes while y contains the labels.

5.6.5 Train Test Split

```
    from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2
```

'train-test-split'is used here to split the data in 80:20 ratio i.e. 80% of the data will be used for training the model while 20% will be used for testing the model that is built out of it.

5.6.6 Feature Scaling

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X_train)
```

```
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

Here, StandardScaler from sklearn is used to scale the attributes in order to resolve the magnitude problem

5.6.7 Training

```
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors=4)
classifier.fit(X_train, y_train)
```

The first step is to import the KNeighborsClassifier class from the sklearn.neighbors library. In the second line, this class is initialized with one parameter, i.e. n-neighbors.There is no ideal value for K and it is selected after testing and evaluation.

5.6.8 Predicting the Test set results

```
y_pred = classifier.predict(X_test)
```

5.6.9 Evaluating the Algorithm

```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

For evaluating an algorithm, confusion matrix, precision, recall and f1 score are the most commonly used metrics. The confusion-matrix and classification-report methods of the sklearn.metrics can be used to calculate these metrics.

EMOTION ANALYSIS:

5.6.10 Import Libraries

Used libraries are listed below:

- pymysql
- re

5.6.11 Compiling a Regular Expression Object

```
import re
def cb(qus):
    WORD = re.compile(r'\w+')
```

The re module is Python's standard library that handles all things regular expression.The re.compile() method is used to match a regular expression on many different strings,it is a good idea to construct a regular expression as a python object.

5.6.12 Conversion of Text to Vector using TF-IDF Vectorizer:

```
from collections import Counter
def text_to_vector(text):
    words = WORD.findall(text)
    return Counter(words)

def get_cosine(vec1, vec2):
    intersection = set(vec1.keys()) & set(vec2.keys())
    numerator = sum([vec1[x] * vec2[x] for x in intersection])
    sum1 = sum([vec1[x] ** 2 for x in vec1.keys()])
    sum2 = sum([vec2[x] ** 2 for x in vec2.keys()])
    denominator = math.sqrt(sum1) * math.sqrt(sum2)
    if not denominator:

        return 0.0
    else:
        return float(numerator) / denominator
```

collections.Counter()- A counter is a container that stores elements as dictionary keys, and their counts are stored as dictionary values.The findall() method returns all non-overlapping matches of pattern in string, as a list of strings. The string is scanned left-to-right, and matches are returned in the order found.Tf-idf is a transformation which is applied to texts to get two real-valued vectors. You can then obtain the cosine similarity of any pair of vectors by taking their dot product and dividing that by the product of their norms. That yields the cosine of the angle between the vectors.Here, the get-cosine function measures the similarity between two vectors by calculating the cosine of the angle between the two vectors

5.6.13 Determines the probability of sentence using VADER

```
def sent(k):
    import nltk
    from nltk.sentiment.vader import SentimentIntensityAnalyzer
    pstv=0
    ngtv=0
    ntl=0
```

```
sid = SentimentIntensityAnalyzer()
ss = sid.polarity_scores(k)
a = float(ss['pos'])
c = float(ss['neg'])
b = float(ss['neu'])
print(ss)
if c >b or  c>a:
    res = "negative"
else:
    res="positive"
return  res
```

VADER uses a list of lexical features (e.g. word) which are labeled as positive or negative according to their semantic orientation to calculate the text sentiment. Vader sentiment returns the probability of a given input sentence to be positive, negative, and neutral.The Compound score is a metric that calculates the sum of all the lexicon ratings which have been normalized between -1(most extreme negative) and +1 (most extreme positive).

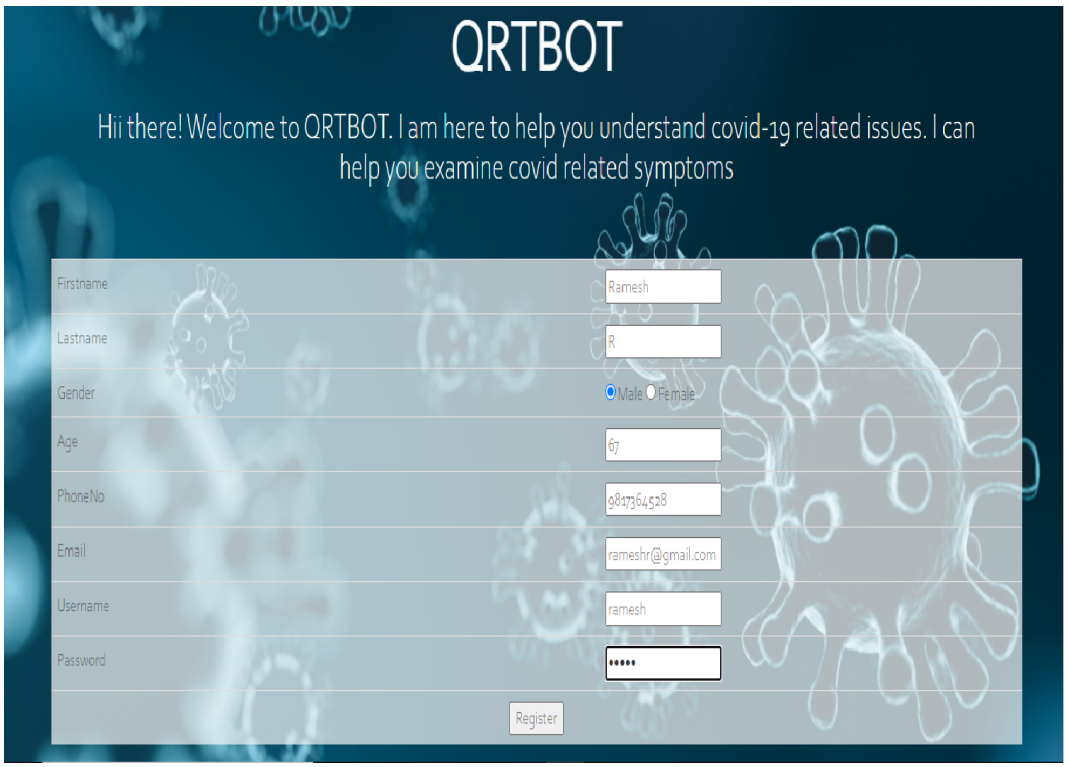


Figure 5.12: Registration Page

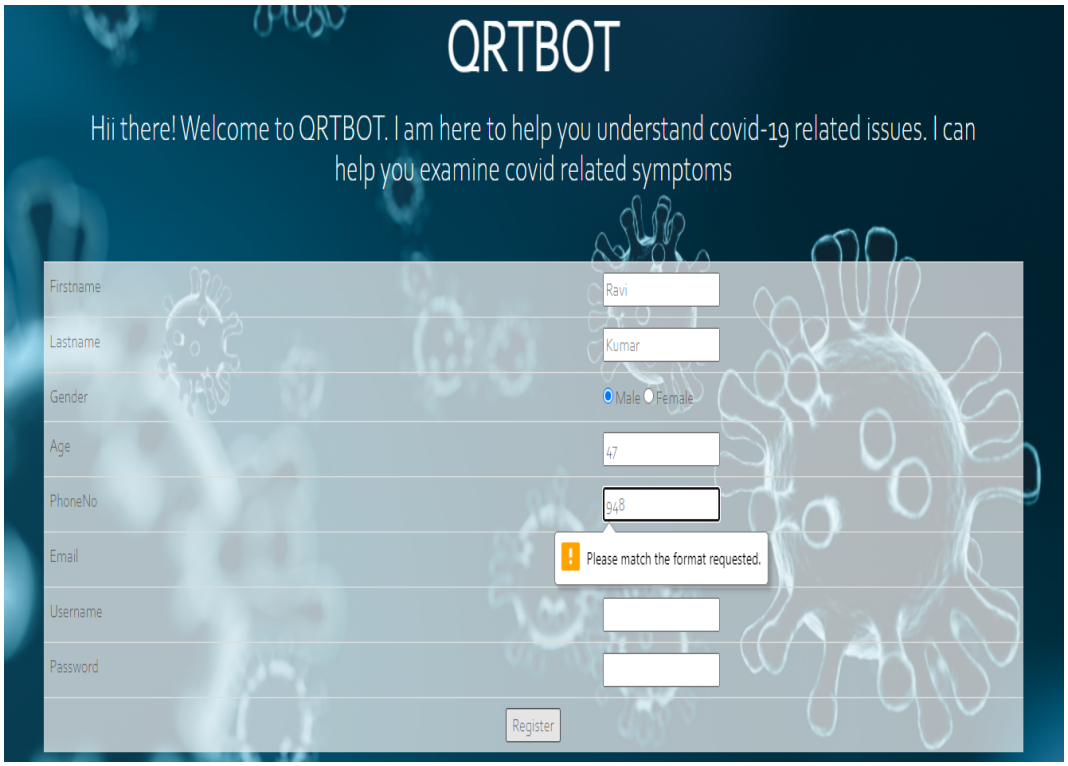


Figure 5.13: HTML5 Validation

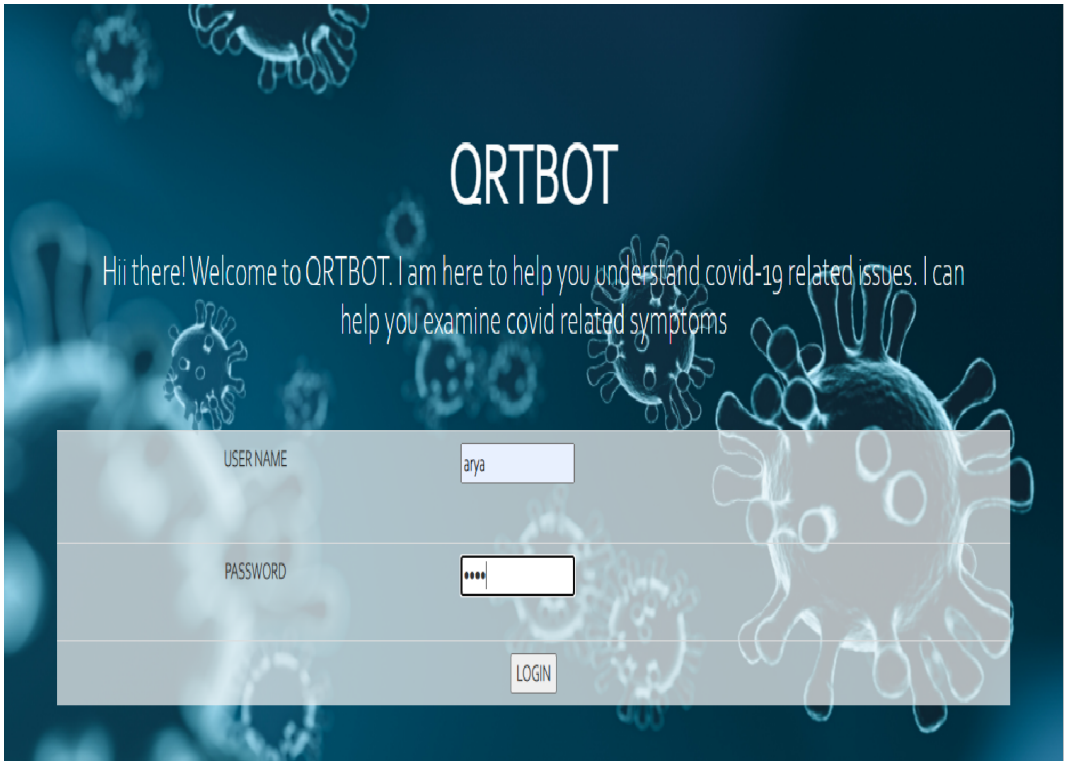


Figure 5.14: Login Page

Figure 5.15: Health Status Update

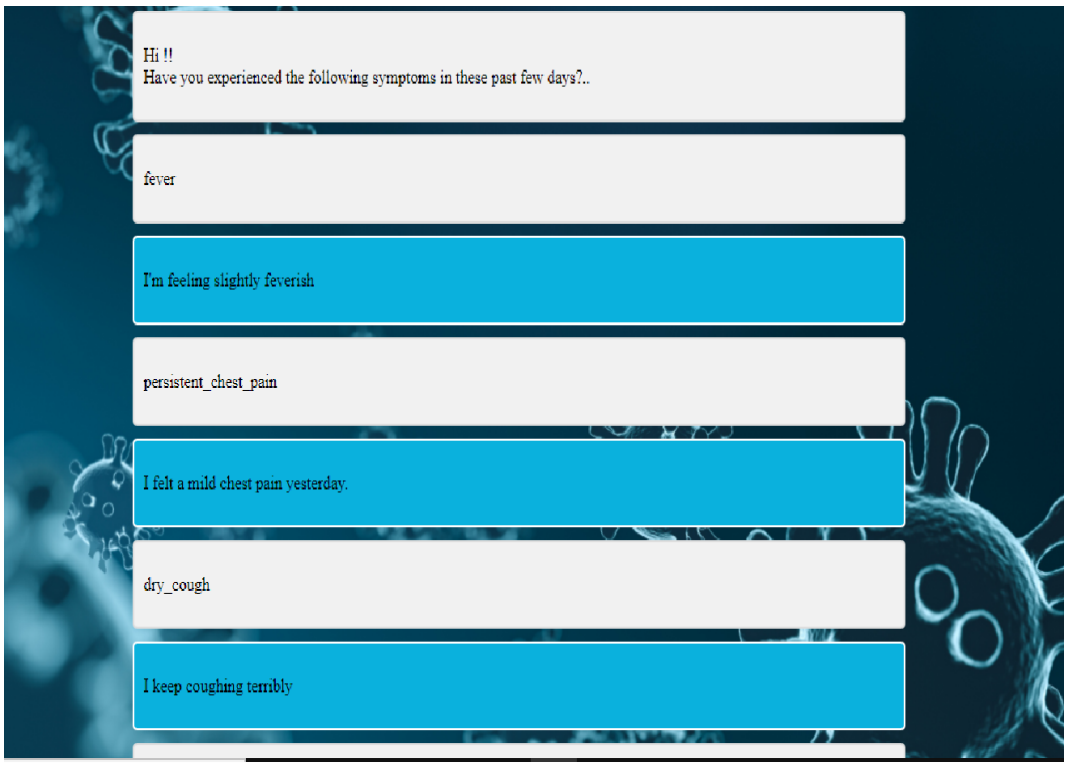


Figure 5.16: Covid-19 Symptom Analysis

Figure 5.17: Emotion Analysis

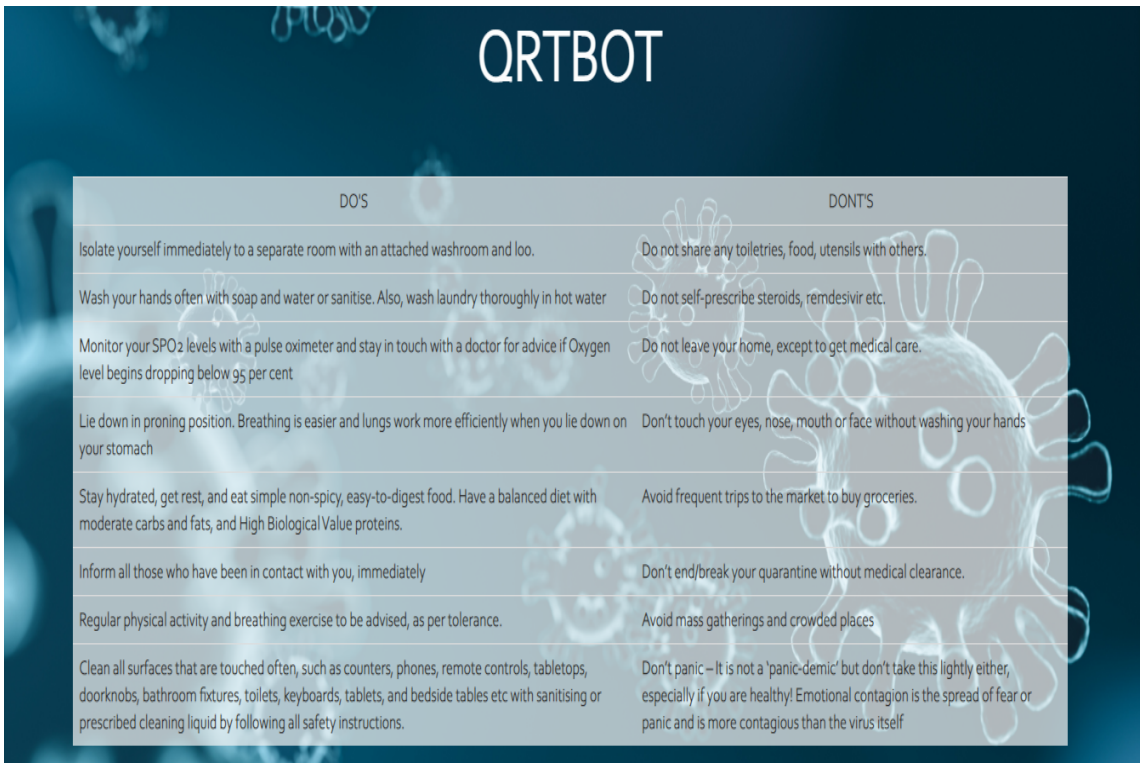


Figure 5.18: Precautions Page

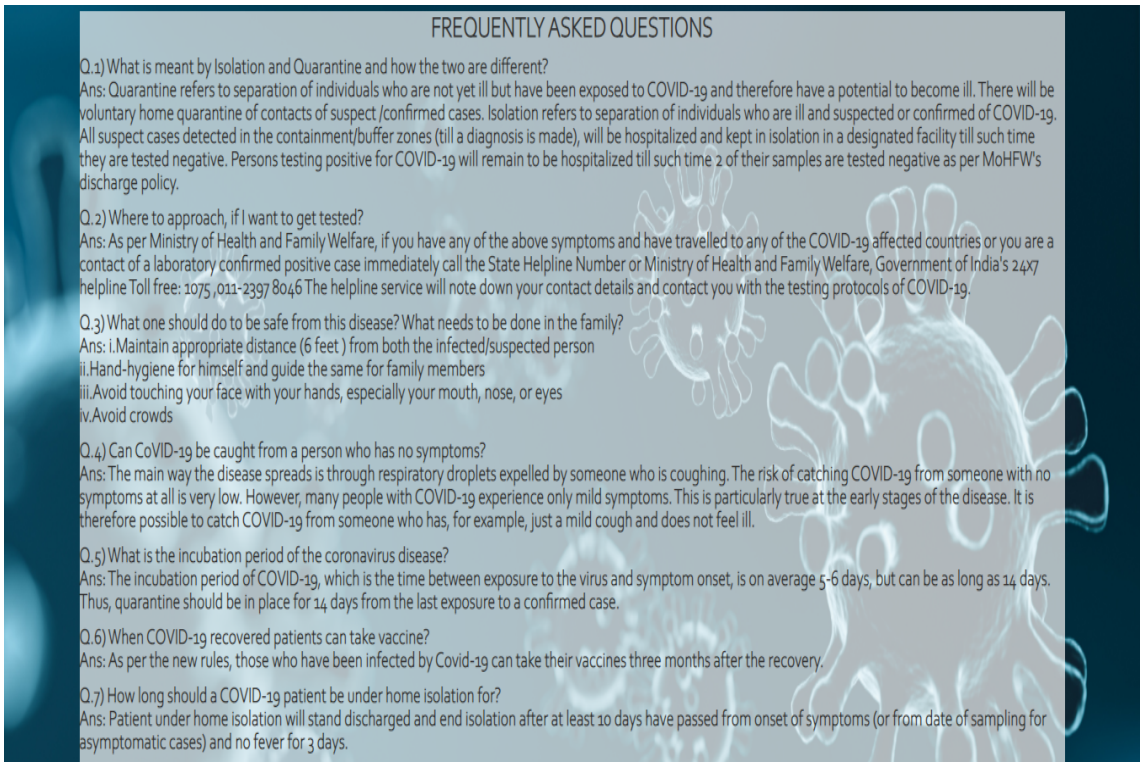


Figure 5.19: FAQ Page

CHAPTER 6

RESULTS & DISCUSSION

6.1 Covid Symptom Analysis

The binary classification is used for predicting whether the user has a high chance of being Covid-19 Positive.Here, it identifies the symptoms by asking the user a set of 14 questions and predicts the severity with the help of KNN algorithm.It also calculates a score where the score depends upon the sum of 1s in the list(user's responses) divided by 14 and also refers to the Health Status Update page and checks if it contains any entries which refers to the user's medical history.

If the score is greater than 0.5 and if the user has updated any chronic disease then it will display the following as shown in Fig 3. else it will display the message as shown in Fig 3.

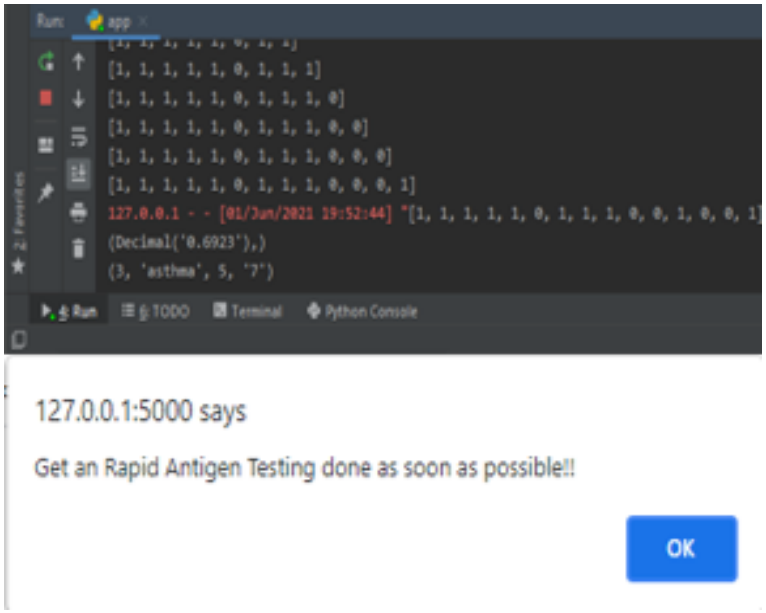


Figure 6.1: Covid Symptom Analysis Results

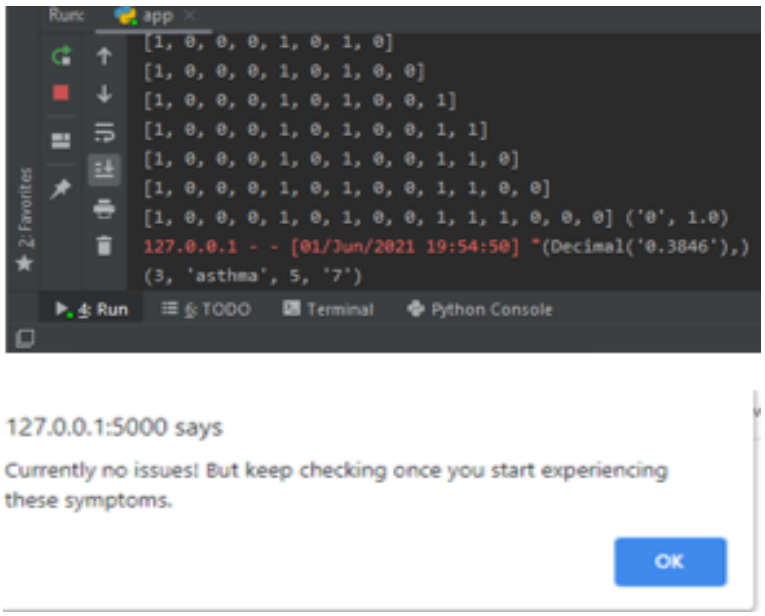
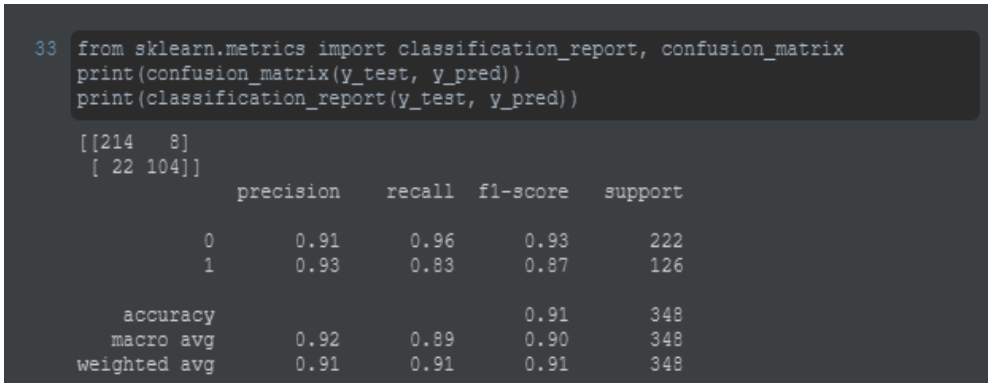


Figure 6.2: Covid Symptom Analysis Results

6.1.1 Computing and printing the evaluation metrics of binary classification

The evaluation metrics are evaluated using the normal formulas which we use to compute them manually. An accuracy of 91% was obtained for the trained model after it was tested on the test dataset.



```
33 from sklearn.metrics import classification_report, confusion_matrix
   print(confusion_matrix(y_test, y_pred))
   print(classification_report(y_test, y_pred))

   [[214   8]
    [ 22 104]]
                  precision    recall  f1-score   support

               0       0.91      0.96      0.93       222
               1       0.93      0.83      0.87       126

        accuracy                           0.91       348
       macro avg       0.92      0.89      0.90       348
    weighted avg       0.91      0.91      0.91       348
```

Figure 6.3: Results Of Binary Classification

6.2 Emotion Analysis

The multiclass classification is used for identifying the emotional state of the user.It classifies the sentiment polarity as positive, negative, or neutral. The sentence similarity between the user response and the data in the dataset is checked using TF-IDF. The emotion in the sentence is then analysed using VADER which gives us scores of the Positive, Negative, Neutral and Compound as shown in Fig 6.4.

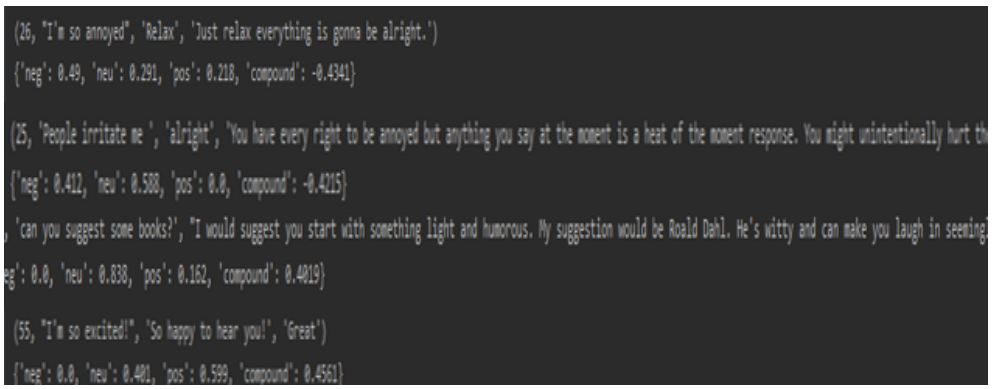


Figure 6.4: Emotion Analysis Result

The compound score is the sum of positive, negative and neutral scores which is then normalized between -1 (most extreme negative) and +1 (most extreme positive). The more compound score closer to +1, the higher the positivity of the text and vice versa. Depending upon the emotions in user's response the chatbot replies.

CHAPTER 7

CONCLUSION & FUTURE SCOPE

7.1 Conclusion

Medical chatbot has a high impact on the health culture of the state. It has improved reliability and is less prone to human errors. The facts that the chatbot is free and can be accessed wherever the user is, be it their working environment, prompt the user to have it and use it. We develop a smart chatbot for COVID-19 Assistance using Machine Learning and Natural Language Processing during quarantine.

We have outlined the design of the proposed project, which aims to predict whether the user has a chance of being affected by COVID-19. It includes two major modules. The first one is Symptom Prediction which uses KNN algorithm and analyses user replies to a questionnaire, to determine if high case of positive COVID-19. The second one is Sentimental Analysis which analyses the user replies to determine emotional state of the user and responds accordingly.

7.2 Future Scope

In the future, we can improve our sentiment analysis model and make it applicable to the human voice. A video call with a specialized doctor can also be made depending on the availability of the user rather than based on the availability of doctors. Along with that, a method to notify the nearest COVID-19 test centre location based on the user's location can be added. It can also inform the user on status of vaccine availability in nearby centres and hospitals.

Bibliography

[1] Miner, Adam S and Laranjo, Liliana and Kocaballi, A Baki. *"Chatbots in the fight against the COVID-19 pandemic"*. npj Digital Medicine, 3(1):1– 4, 2020.

[2] Khadikar, Sunath and Sharma, Pawan and Paygude, Priyanka. *"Compassion Driven Conversational Chatbot Aimed for better Mental Health"*.

[3] Bharti, Urmil and Bajaj, Deepali and Batra, Hunar and Lalit, Shreya and Lalit, Shweta and Gangwani, Aayushi. *"Medbot: Conversational artificial intelligence powered chatbot for delivering tele-health after covid-19"*. 2020 5th International Conference on Communication and Electronics Systems (ICCES), pages 870–875. IEEE, 2020.

[4] Mathew, Rohit Binu and Varghese, Sandra and Joy, Sera Elsa and Alex, Swanthana Susan. *"Chatbot for Disease Prediction and Treatment Recommendation using Machine Learning"*. 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), pages 851–856. IEEE, 2019

[5] Mak, WW and Cheung, F and Woo, J and Lee, D and Li, P and Chan, KS and Tam, CM. *"A comparative study of the stigma associated with infectious diseases (SARS, AIDS, TB)."*. Hong Kong medical journal= Xianggang yi xue za zhi,15:34,2009.

[6] Sarah Merriam,Dr Hannah Behrendt. *"Covid-19:how do we encourage the right behaviours during an epidemic":*. https://www.bi.team/blogs/covid-19-how-do-we-encourage-the-right-behaviours-during-an-epidemic.

[7] Lucas, Gale M and Gratch, Jonathan and King, Aisha and Morency, Louis-Philippe. *"It's only a computer: Virtual humans increase willingness to disclose"*. Computers in Human Behavior, pages 94–100, 2014.

[8] Sagar, Rajesh and Dandona, Rakhi and Gururaj, Gopalkrishna and Dhaliwal, RS and Singh, Aditya and Ferrari, Alize and Dua, Tarun and Ganguli, Atreyi and Varghese, Mathew and Chakma, Joy K and others. *"The burden of mental disorders across the states of India: the Global Burden of Disease Study 1990–2017"*. The Lancet Psychiatry, pages 148-161, 2020.

[9] Bharti, Urmil and Bajaj, Deepali and Budhiraja, Payal and Juyal, Meghna and Baral, Sushmita and others. *"Android Based e-Voting Mobile App Using Google Firebase as BaaS"* International Conference on Sustainable Communication Networks and Application, pages 231-241, 2019.