

**SAVITRIBAI PHULE PUNE UNIVERSITY**

**A PROJECT REPORT ON**

**“Healthbot –An Interactive Bot for Medical  
Diagnostics”**

**SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE**

**BACHELOR OF ENGINEERING**

**(Computer Engineering)**

**BY**

Sakshi Sajjanshetty    B151024332  
Dheeraj Bhor            B151024217  
DhanashreeDhanajkar B151024241  
Abishek Gaikwad        B151024247

**Under The Guidance of**  
Prof Ms. Kalpna Saharan

**DEPARTMENT OF COMPUTER ENGINEERING**  
**STES' RMD Sinhgad School of Engineering**



**Warje, Pune-411058**  
**Affiliated to Savitribai Phule Pune University**

RMDSSOE, Department of Computer Engineering 2021-22

**A PROJECT APPROVAL SHEET**

A Project Title

**Healthbot –An Interactive Bot for Medical Diagnostics**

Is successfully completed by

<b>Sakshi Sajjanshetty</b>	<b>B151024332</b>
<b>Dheeraj Bhor</b>	<b>B151024217</b>
<b>Dhanashree</b>	<b>B151024241</b>
<b>Abishek Gaikwad</b>	<b>B151024247</b>

At

DEPARTMENT OF COMPUTER ENGINEERING

RMD SINHGAD SCHOOL OF ENGINEERING , PUNE

SAVITRIBAI PHULE PUNE UNIVERSITY,PUNE

ACADEMIC YEAR 2021-2022

Mrs. Kalpana Saharan  
Internal Guide  
Dept. of Computer Engg.

Prof. Vina M. Lomte  
H.O.D  
Dept. of Computer Engg.



**RMD Sinhgad School of Engineering**  
**DEPARTMENT OF COMPUTER ENGINEERING**

**CERTIFICATE**

This is to certify that the Project Entitled

**“Healthbot –An Interactive Bot for Medical  
Diagnostics”**

Submitted by

Sakshi Sajjanshetty	B151024332
Dheeraj Bhor	B151024217
Dhanashree	B151024241
Abishek Gaikwad	B151024247

Is a bonafide work carried out by Students under the supervision of Prof. Kalpana Saharan and it is submitted towards the partial fulfillment of the requirement of Bachelor of Engineering (Computer Engineering).

Mrs. Kalpana Saharan  
Internal Guide  
Dept. of Computer Engg.

Prof. Vina M. Lomte  
H.O.D  
Dept. of Computer Engg.

Dr. V.V.Dixit  
Principle  
RMD Sinhgad School of Engineering

Signature of Internal Examiner

Signature of External Examiner

## **ABSTRACT**

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 (NLP) 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.

## Acknowledgments

It gives us great pleasure in presenting the preliminary project report on 'Healthbot –An Interactive Bot for Medical

I would like to take this opportunity to thank my internal guide Prof. Kalpana saharan for giving me all the help and guidance I needed. I am really grateful to them for their kind support. Their valuable suggestions were very helpful.

We are extremely grateful to Prof. Vina Lomate, Head of the Department of Computer Engineering, for encouragement during the course of the project work.

We also extend our heartfelt gratitude to the staff of Computer Engineering Department for their cooperation and support.

Sajjanshetty Sakshi

Dheeraj Bhor

Dhanajkar Dhanashree

Abhishek Gaikwad

(B.E. Computer Engg.)

# INDEX

<b>1</b>	<b>Synopsis</b>	<b>12</b>
1.1	Project Title .....	13
1.2	Project Option .....	13
1.3	Project Guide .....	13
1.4	Problem Statement .....	13
1.5	Abstract .....	13
1.6	Goals and Objectives.....	14
1.7	Names of Conferences / Journals where papers can be published	14
1.8	Review of Conference/Journal Papers supporting Project idea	14
1.9	Plan of Project Execution .....	16
<b>2</b>	<b>Technical Keywords</b>	<b>17</b>
2.1	Area of Project .....	18
2.2	Technical Keywords.....	18
<b>3</b>	<b>Introduction</b>	<b>19</b>
3.1	Project Idea.....	20
3.2	Motivation of the Project.....	20
3.3	Literature Survey.....	20
<b>4</b>	<b>Problem Definition and scope</b>	<b>22</b>
4.1	Problem Statement .....	23
4.1.1	Goals and objectives .....	23
4.1.2	Statement of scope.....	23
4.2	Major Constraints.....	23
4.3	Methodologies of Problem solving and efficiency issues .....	24
4.4	Outcome .....	24
4.5	Applications.....	24
4.6	Hardware Resources Required .....	25
4.7	Software Resources Required.....	25
<b>5</b>	<b>Project Plan</b>	<b>27</b>
5.1	Project Estimates.....	28
5.1.1	Reconciled Estimates.....	28
5.1.2	Project Resources .....	29
5.1.3	Hardware Requirement .....	29
5.1.4	Software Requirement .....	30
5.2	Risk Management w.r.t. NP Hard analysis .....	30
5.2.1	Risk Identification .....	30
5.2.2	Risk Analysis .....	31
5.2.3	Overview of Risk Mitigation, Monitoring, Management .....	32

5.3	Project Schedule .....	35
5.3.1	Project task set.....	35
5.3.2	Task network.....	36
5.4	Team Organization.....	37
5.4.1	Team structure.....	37
5.4.2	Management reporting and communication .....	37
<b>6</b>	<b>Software requirement specification</b>	<b>38</b>
6.1	Introduction .....	39
6.1.1	Purpose and Scope of Document.....	39
6.1.2	Overview of responsibilities of Developer .....	39
6.2	Usage Scenario .....	39
6.2.1	User profiles .....	39
6.2.2	Use Case View .....	40
6.3	Data Model and Description.....	41
6.3.1	Data Description.....	41
6.3.2	Data objects and Relationships .....	42
6.4	Functional Model and Description.....	43
6.4.1	Data Flow Diagram .....	43
6.4.2	Activity Diagram.....	44
6.4.3	Non-Functional Requirements .....	45
6.4.4	Sequence Diagram.....	46
6.4.5	Design Constraints.....	47
6.4.6	Software Interface Description.....	47
<b>7</b>	<b>Detailed Design Document using Annexure A and B</b>	<b>48</b>
7.1	Introduction .....	49
7.2	Architectural Design .....	51
7.2.1	Overall Description.....	51
7.3	Data design .....	52
7.3.1	Define Intents .....	52
7.3.2	Data Preparation .....	52
7.3.3	Create Bag of Words .....	52
7.3.4	Model training .....	52
7.4	Component Design.....	52
7.4.1	Class Diagram .....	52
<b>8</b>	<b>Project Implementation</b>	<b>54</b>
8.1	Introduction .....	55
8.2	Tools and Technologies Used.....	55
8.2.1	Software used .....	55
8.2.2	Hardware specification .....	57
8.2.3	Components .....	57
8.3	Methodologies .....	58
8.4	Verification and Validation for Acceptance.....	61

8.4.1	Verification .....	61
8.4.2	Validated .....	61
<b>9</b>	<b>Software Testing</b>	<b>62</b>
9.1	Type of Testing Used .....	63
9.2	Test Cases and Test Results .....	63
9.2.1	Unit Testing .....	63
9.2.2	Integration Testing.....	65
9.2.3	System testing.....	67
9.2.4	Functional Testing .....	69
<b>10</b>	<b>Results</b>	<b>72</b>
10.1	Screen shots .....	73
10.2	Outputs .....	74
<b>11</b>	<b>Deployment and Maintenance</b>	<b>76</b>
11.1	Installation and un-installation.....	77
11.1.1	Installation of Python IDE .....	77
11.1.2	Installation of PyCharm .....	81
11.1.3	Installation of Sublime Text 3 .....	85
11.1.4	Installation of Flask .....	87
11.2	User help.....	90
<b>12</b>	<b>Conclusion and Future Scope</b>	<b>91</b>
12.1	Conclusion.....	92
12.2	Future Scope .....	92
12.3	Summary.....	92
<b>13</b>	<b>References</b>	<b>93</b>
<b>14</b>	<b>Plagiarism Report</b>	<b>95</b>
<b>15</b>	<b>Information of Project Group Members</b>	<b>97</b>



# List of Figures

5.1	Task Network .....	36
6.1	Use case diagram .....	40
6.2	Creation of Dataset .....	41
6.3	Data Preparation .....	41
6.4	Packet Entity .....	42
6.5	Data Flow Diagram Level 0 .....	43
6.6	Data Flow Diagram Level 1 .....	43
6.7	Activity diagram (module-1) .....	44
6.8	Activity diagram (module-2) .....	44
6.9	Sequence diagram .....	46
7.1	Architecture diagram .....	51
7.2	Class Diagram .....	53
8.1	Defining Intent .....	58
8.2	Data Preparation .....	58
8.3	Bag of words .....	59
8.4	Model Training .....	59
8.5	SGD optimizer .....	60
8.6	Chatbot model .....	60
8.7	Class prediction .....	60
8.8	Bot response .....	60
9.1	Test case 1 .....	64
9.2	Test case 2 .....	64
9.3	Test case 3 .....	65
9.4	Test case 4 .....	66
9.5	Test case 5 .....	66
9.6	Test case 6 .....	67
9.7	Test case 7 .....	68
9.8	Test case 8 .....	68
9.9	Test case 9 .....	69
9.10	Test case 10 .....	70
9.11	Test case 11 .....	70
9.12	Test case 12 .....	71
10.1	Home/start page of Curebot .....	73
10.2	Conversation with the Curebot .....	74
10.3	Solving query of the user .....	75
11.1	Python IDE installation setup .....	77
11.2	Python IDE installation setup .....	77
11.3	Python IDE installation setup .....	78
11.4	Python IDE installation setup .....	78
11.5	Python IDE installation setup .....	79
11.6	Python IDE installation setup .....	79
11.7	Python IDE installation setup .....	80

11.8	Python IDE installation Complete .....	80
11.9	PyCharm installation setup .....	81
11.10	PyCharm installation setup .....	81
11.11	PyCharm installation setup .....	82
11.12	PyCharm installation setup .....	82
11.13	PyCharm installation setup .....	83
11.14	PyCharm installation Complete .....	83
11.15	Run PyCharm Community Edition .....	84
11.16	Create New PyCharm Project .....	84
11.17	Sublime Text 3 installation setup .....	85
11.18	Sublime Text 3 installation setup .....	85
11.19	Sublime Text 3 installation setup .....	86
11.20	Sublime Text 3 installation setup .....	86
11.21	Sublime Text 3 installation Complete .....	87
11.22	Flask installation Setup .....	87
11.23	Flask installation Setup .....	88
11.24	Flask installation Setup .....	88
11.25	Flask installation Setup .....	88
11.26	Flask installation Setup .....	89
11.27	Flask installation Setup .....	89
11.28	Flask installation Completed .....	90
P.1	Plagiarism with 100% Uniqueness .....	96

# List of Tables

1.1	Project Execution.....	16
4.1	Hardware Requirements.....	25
5.1	Tasks divisions weeks.....	29
5.2	Hardware Requirements.....	29
5.3	Risk Table .....	31
5.4	Risk Probability definitions [?] .....	31
5.5	Risk Impact definitions [?].....	32
5.6	Risk ID 1 .....	33
5.7	Risk ID 2 .....	33
5.8	Risk ID 3 .....	34
5.9	Risk ID 4 .....	34
5.10	Risk ID 5 .....	35
5.11	Team Structure .....	37
8.1	Hardware specification .....	57

# **CHAPTER 1**

## **SYNOPSIS**

## **1.1 Project Title**

Healthbot -An Artificially Intelligent Interactive Bot for Medical Diagnostics.

## **1.2 Project Option**

Un-sponsored

## **1.3 Project Guide**

Prof Mrs.Kalpna Saharan

## **1.4 Problem Statement**

Today in the present era, the major challenges that India as a country is facing is to cater to good quality and affordable healthcare services to its growing population and at the same time, they are not cost efficient. So, the main aim of our project is to design a Conversational Artificial Intelligence Powered Chatbot for Medical Diagnostics which mainly focuses on rural parts as well as poor and needy people of our country.

## **1.5 Abstract**

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.

## **1.6 Goals and Objectives**

- To provide efficient solutions for user queries.
- To build an intuitive, and easy communication with the Bot through speech and user demands.
- The Proposed Bot offers enhanced technologies benefits, reduce user hassle costs, and increased user welfare.
- Chatbots applications streamline interactions between people and services, enhancing user.

## **1.7 Names of Conferences / Journals where papers can be published: -**

- ✓ IEEE/ACM Conference/Journal 1
- ✓ Conferences/workshops in IITs
- ✓ Central Universities or SPPU Conferences
- ✓ IEEE/ACM Conference/Journal 2

## **1.8 Review of Conference/Journal Papers supporting Project idea: -**

The chatbot is a software program that is used to interact with clients using natural language. Humans can replace by chatbots for monotonous jobs of answering queries and giving efficient responses. Chatbots used for acquiring knowledge. It can be implemented on our mobiles and local personal systems and can access the internet. Chatbots communicates with clients in any particular domain with their query as input in general conversational statements. This literature review presents History, Technology and Existing Work of the Chat-Bots. Chatbots have good conversational abilities they are good at emulating a human conversation, and they learn through knowledge. Alan Turing wondered if a computer program could talk to a group of people without realizing that their interlocuter was artificial, this question was named Turing test. The first chatbot with ELIZA name was constructed in 1966. ELIZA simulated a psychotherapist's operation, returning the user's sentences in the interrogative form. ELIZA uses pattern matching and a response selection scheme based on templates. In the history of chatbots ALICE was the first

online chatbot inspired by ELIZA. ALICE was based on pattern-matching. Difference between ALICE and ELIZA was that ALICE was developed using new language AIML created for this purpose. PARRY is considered more advanced than ELIZA as it is supposed to have personality and better controlling structure. Chats-Bots mentioned in this Literature survey is also based on medicinal purpose such as K-Bot. There is a well-recognized need for a shift to proactive asthma care given the impact asthma has on overall healthcare costs. The demand for continuous monitoring of patient's adherence to the medication care plan, assessment of environmental triggers, and management of asthma can be challenging in traditional clinical settings and taxing on clinical professionals. K-Bot is limited to only Asthma and cannot treat on anything else. Deep learning is a subset of machine learning in artificial intelligence that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network. Recurrent Neural Network (RNN) are a type of Neural Network where the output from previous step is fed as input to the current step. ... It uses the same parameters for each input as it performs the same task on all the inputs or hidden layers to produce the output.

Natural language processing (NLP) deals with building computational algorithms to automatically analyze and represent human language. In this paper, survey on Neural Network in Speech Recognition is Referred from the research paper speech recognition using deep neural networks published in 2019. Deep learning algorithms have been mostly used to enhance the capabilities of computers so that it understands what humans can do, which includes speech recognition. The main focus of the review was to shed a light on paper that employ the Deep learning technique for implementation with best use of neural network. Machine learning is defined as the field of study that provides computers with the ability to learn from input data without being explicitly programmed to do so. Automatic speech recognition is the capability of a machine or computer to recognize the content of words and phrases in an uttered language and transform them to a machine-understandable format. Automatic speaker recognition can be defined as the process of recognizing the unknown speaker on the basis of the information embedded in his/her speech signal using machine. Disadvantages of Speech Recognition were that it was difficult to distinguish the language of the user, also it may face difficulty in accent recognition as well. COVID-19 has taken control all over world, millions of people have suffered from Covid and many have resulted in death. There is a survey based on COVID-19 which was implemented using Deep Learning. This survey explores how Deep Learning has battled the COVID-19 pandemic and provides direction for future research on COVID-19. We cover Deep Learning applications in NLP, Computer Vision, Life Science and Epidemiology.

## 1.9 Plan of Project Execution

ID	Task Name	Start	Finish	Duration	Complete
1	Requirement Engineering	21 Jun	26 Jun	4	100%
2	Design & Modeling	28 Jun	6 Jul	38	95%
3	Implementation	20 Aug	24 oct	64	0%
4	Testing	30 Sep	10 Oct	10	0%
5	Deployment	12 Oct	1 Dec	26	0%

Table 1.1: Project Execution



**CHAPTER 2**  
**TECHNICAL KEYWORDS**

## **2.1 Area of Project**

Machine Learning (ML) OR Deep Learning

## **2.2 Technical Keywords**

### **Chatbot:**

A chatbot is a software application used to conduct an on-line chat conversation via text or text-to-speech, in lieu of providing direct contact with a live human agent.

### **COVID-19:**

Coronavirus disease (COVID-19) is an infectious disease caused by a newly discovered coronavirus. Most people infected with the COVID-19 virus will experience mild to moderate respiratory illness and recover without requiring special treatment. Older people, and those with underlying medical problems like cardiovascular disease, diabetes, chronic respiratory disease, and cancer are more likely to develop serious illness.

### **Deep Learning:**

Deep learning is a subset of machine learning in artificial intelligence that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network.

### **Natural Language Toolkit (NLTK):**

The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing for English written in the Python programming language.

### **Natural Language Processing (NLP):**

Natural language processing is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data.

### **Neural Network:**

A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature.

# **CHAPTER 3**

## **INTRODUCTION**

### 3.1 Project Idea

So, the main aim of our project is to design a Conversational Artificial Intelligence Powered Chatbot for Medical Diagnostics which mainly focuses on rural parts as well as poor and needy people of our country. One of the major challenges that India is currently facing is to provide good quality and affordable healthcare for growing population. Nowadays, it is becoming very difficult to provide healthcare facilities as we have seen in COVID-19 critical situations that condition in India was getting worse because of lack of transportation availability, doctors and hospitality and sometimes it causes the people to postpone their treatment as well as there is an increment in death count.

### 3.2 Motivation of the Project

One of the major challenges that India is currently facing is to provide good quality and affordable healthcare for growing population. Nowadays, it is becoming very difficult to provide healthcare facilities as we have seen in COVID-19 critical situations that condition in India was getting worse because of lack of transportation availability, doctors and hospitality and sometimes it causes the people to postpone their treatment as well as there is an increment in death count. For that we have come up with a solution of a Conversational Artificially Intelligent Powered Chatbot which communicates with Patient (End-user) through web-UI, gives precaution and tries to solve their problem with the help symptoms provided by Patient itself.

### 3.3 Literature Survey

- ***kBot: Knowledge-enabled Personalized Chatbot for Asthma Self-Management***

**Authors:** Dipesh Kadariya, Revathy Venkata Ramanan, Hong Yung Yip, Maninder Kalra, Krishna prasad Thirunarayanan, Amit Sheth.

**Publication Year:** 2019

**Description:** kBot is a Chatbot mainly used to help Asthma Patient through meaningful conversation.

**Limitation:** This Chatbot is only used for Asthma Self-Management.

**Location:** India.

- ***Medbot: Conversational Artificial Intelligence powered Chatbot for Delivering Tele-Health for COVID-19***

**Authors:** Urmil Bharti, Deepali Bajaj, Hunar Batra

**Publication Year:** 2020

**Description:** This paper highlights us about the Medbot which is a Telehealth Consultant.

**Limitation:** It is Capable of converting Audio to text but not vice versa.

**Location:** India.

- ***Path-Bot: An Intelligent Chatbot for Guiding Visitors and Locating Venues***

**Authors:** Katlego Mabunda, Abejide Ade-Ibijola

**Publication Year:** 2019

**Description:** This paper presented the development of PathBot as a tool to assist students and visitors to navigate through the campus venues.

**Limitation:** It does not handle rerouting and cases where the user is lost.

**Location:** South Africa

- ***Speech Recognition Using Deep Neural Networks***

**Authors:** Ali Bou Nassif, Ismail Shahin, Imtinan Attili, Mohammad Azzeh, and Khaled Shaalan

**Publication Year:** 2019

**Description:** Automatic speech recognition is the capability of a machine or computer to recognize the content of words and phrases in an uttered language and transform them to a machine- understandable format

**Limitation:** The discrimination between two languages may become challenging as well as accent recognition.

**Location:** UAE

- ***COVID-19: Deep Learning Application***

**Authors:** Connor Shorten, Taghi M Khoshgoftaar, Borko Furht

**Publication Year:** 2021

**Description:** This survey explores how Deep Learning has battled the COVID-19 pandemic and provides direction for future research on COVID-19. We cover Deep Learning applications in NLP, Computer Vision, Life Science and Epidemiology

**Limitation:** Interpretability, Generalization Metrics, Learning from Limited Labeled Data, and Data Privacy

**Location:** USA

**CHAPTER 4**

**PROBLEM DEFINITION AND  
SCOPE**

## **4.1 Problem Statement**

Today in the present era, the major challenges that India as a country is facing is to cater to good quality and affordable healthcare services to its growing population and at the same time, they are not cost efficient. So, the main aim of our project is to design a Conversational Artificial Intelligence Powered Chatbot for Medical Diagnostics which mainly focuses on rural parts as well as poor and needy people of our country.

### **4.1.1 Goals and objectives**

- To provide efficient solutions for user queries.
- To build an intuitive, and easy communication with the Bot through speech and user demands.
- The Proposed Bot offers enhanced technologies benefits, reduce user hassle costs, and increased user welfare.
- Chatbots applications streamline interactions between people and services, enhancing user.

### **4.1.2 Statement of scope**

Our System takes diagnostic information of the Patient (end User) in the form of query through text or voice from the web UI Interface and then converts it into the feed able data to the appropriate trained model which is built at the runtime and gives back the required solution(remedy/treatment) for the same in the form of Response Text.

## **4.2 Major Constraints**

- While storing the data in the database, some irrelevant data may get added in the database. The irrelevant data can be called as dirty bit.
- The interruption in the internet connectivity can stop the functioning of application system.

### **4.3 Methodologies of Problem solving and efficiency issues**

- Define a Sequential model. The network will be defined as sequence of layers, each with its own customizable size and activation function. In this model, first layer will be input layer, which requires to define the size of input. After this more and more layers can be added and customized until we reached the final output layer.
- After creating a structure of network, compile it, which transforms the simple sequence of layers into a complex group of matrix operations that dictate how the network behaves.
- Once it is done, we can train or fit the network.
- After training the model, we can use it to make predication on new data.

### **4.4 Outcome**

- Our System takes diagnostic information of the Patient (end User) in the form of query through text or voice from the web UI Interface and then converts it into the feed able data to the appropriate trained model which is built at the runtime and gives back the required solution(remedy/treatment) for the same in the form of Response Text.
- Our system detects all the medical problems help the user by providing them accurate medicines.
- 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.
- Our system 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.

### **4.5 Applications**

- Our system analyzes the symptoms of the patient and give the correct response of the query.



- The users can use our system 24\*7 for any medical issue.
- Our system gives quick reply to the query.
- Our system Provide Medical Information and gives the accurate medicines.
- Our system used in Hospitals, Clinic to handle the patients.
- Curebot collect Patient Data.

## 4.6 Hardware Resources Required

Sr. No.	Parameter	Minimum Requirement	Justification
1	Storage	4 GB	For smooth performance
2	Processor	i3	Easy for multitasking
3	Internet	LAN/WIFI	For connection to internet

Table 4.1: Hardware Requirements

## 4.7 Software Resources Required

Platform:

1. Operating System: Any 64-Bit (Windows/MAC/Linux) is required.
2. Programming Language: Python 3

It is developed by Guido van Rossum. Python is an interpreted high-level general purpose programming language. Python's design philosophy emphasizes Code readability with its notable use of significant indentation. Its language construct as well as its Object-oriented approach aim to help Programmers write clear, logical code for small and large-scale projects. Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

3. User Interface: HTML5 and CSS3

HTML5 is a markup language used for structuring and presenting content on the World Wide Web. It is the fifth and last major HTML version that is a World Wide Web Consortium recommendation. The current specification is known as the HTML Living Standard. HTML5 was first released in a public-facing form on 22 January 2008 with a major update and "W3C

Recommendation" status in October 2014. Its goals were to improve the language with support for the latest multimedia and other new features; to keep the language both easily readable by humans and consistently understood by computers and devices such as web browsers, Parser etc.

4. Editors: PyCharm/VS Code/Sublime Text3.

PyCharm is an integrated development environment used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains.

5. Web Application: Latest Version of Google Chrome/IE Firefox/Other.

# **CHAPTER 5**

## **PROJECT PLAN**

## 5.1 Project Estimates

### 5.1.1 Reconciled Estimates

#### 5.1.1.1 Cost Estimate

A cost estimation is the process of approximation of the cost of a program, project or operation. The cost estimate is the product of the cost estimating process. The cost estimation of the project depends on some major aspects like construction cost, manufacturing cost, software development cost and maintenance cost. Our system requires hardware component grid and processing board so cost is bit high. Estimated cost of our system will depend on the area of deployment.

#### 5.1.1.2 Time Estimates

Week 1	(02-Aug-2021)	Deciding the project topic
Week 2	(09- Aug-2021)	Discussion regarding the topic with project guide
Week 3	(16- Aug-2021)	Finalize the domain and topic of the project.
Week 4	(23- Aug-2021)	Analyzed the requirements of the project
Week 5	(30-Aug-2021)	Estimated the cost and time requirement of the project
Week 6	(06- Sept-2021)	Started Research about different implementation technique required
Week 7	(13- Sept-2021)	Started the project work with HTML5, CSS3, VS Code
Week 8	(27-Oct-2021)	Started studying about libraries/ Modules required for the project
Week 9	(11- Oct-2021)	Started the project work with installing Python, HTML5, CSS3.
Week 10	(25- Oct-2021)	Continued the work with manuscript of the publishing paper
Week 11	(08- Nov-2021)	Published a Review paper regarding the project
Week 12	(27-Dec-2021)	Prepared a rough model of the project and presented in the college
Week 13	(03-Jan-2022)	Started to prepare different UML Diagram and Architecture
Week 14	(10-Jan-2021)	continued the work on architecture and design of the Project

Week 15	(17-Jan-2022)	Started to prepare Customized dataset
Week 16	(24-Jan -2022)	Started working on actual implementation of the Project
Week 17	(07-Feb-2022)	Started studying about deep learning process
Week 18	(14-Feb-2022)	Started studying about neural networks Concept
Week 19	(28-Feb-2022)	Started building a training model for our project
Week 20	(07-Mar-2022)	Applying model and started working on prediction of classes
Week 21	(14-Mar-2022)	Started working on generating bot response
Week 22	(21-Mar-2022)	Started working on Front end (HTML & CSS)
Week 23	(28-Mar-2022)	Started working on CSS & Updating the User Interface
Week 24	(04-Apr-2022)	Started working on Flask & updating /refactoring the code.
Week 25	(11-Apr-2022)	Started working on Testing of the bot
Week 26	(18-Apr-2022)	Adding more intents for getting more appropriate bot response
Week 27	(25-Apr-2022)	Deployment of the project on Cloud
Week 28	(02-May-2022)	Published an Implementation Paper regarding the Project

Table 5.1: Tasks divisions weeks

### 5.1.2 Project Resources

### 5.1.3 Hardware Requirement

Sr. No.	Parameter	Minimum Requirement	Justification
1	Storage	4 GB	For smooth performance
2	Processor	i3/i5	Easy for multitasking
3	Internet	LAN/WIFI	For connection to internet

Table 5.2: Hardware Requirements

### **5.1.4 Software Requirement**

The software requirements are as follows:

- OS (Linux/Windows/MAC)
- Programming Language (Backend): -Python3,
- Editors: - PyCharm, VS Code, Sublime Text 3.
- Front-end: - Flask (for UI), HTML, CSS

## **5.2 Risk Management w.r.t. NP Hard analysis**

### **5.2.1 Risk Identification**

Risk is an expectation of loss, a potential problem that may or may not occur in the future. It is generally caused due to lack of information, control or time. Loss can be anything, increase in production cost, development of poor-quality software, not being able to complete the project on time. Risk increases with the increase in noise of the environment in which the system is working.

1. Have top software and customer managers formally committed to support the project?

Ans: Yes, here there are no customers and the software managers are the team members itself.

2. Are requirements fully understood by the software engineering team?

Ans: Yes, and the actions are taken accordingly.

3. Do end-users have realistic expectations?

Ans: No, the prototype is generated which if implemented realistically then only it can meet up to expectations.

4. Does the software engineering team have the right mix of skills?

Ans: Yes, we have sound knowledge regarding the software and hardware.

5. Are project requirements stable?

Ans: Yes, the requirements are stable and no dynamic demands are occurring.

6. Is the number of people on the project team adequate to do the job?

Ans: Yes, they are more than enough.

7. Do all customer/user constituencies agree on the importance of the project and on the requirements for the system/product to be built?

Ans: Yes, the project if got implemented then it can very helpful for the patients in this pandemic situation.

### 5.2.2 Risk Analysis

Risk analysis a very important aspect of software risk management. In this phase the risk is identified and then categorized. After the categorization of risk, the level, likelihood (percentage) and impact of the risk is analyzed. Likelihood is defined in percentage after examining what are the chances of risk to occur due to various technical conditions.

ID	Risk Description	Probability	Impact		
			Schedule	Quality	Overall
1	Complexity of technology.	Low	High	Low	High
2	Technical knowledge possessed by testing team.	High	Low	High	High
3	Conflicts within the team.	Low	High	Low	Low
4	Teams being distributed over a large geographical area.	Low	Low	Low	Low
5	Usage of poor-quality testing tools.	Medium	Low	High	High

Table 5.3: Risk Table

Probability	Value	Description
High	The knowledge of tester	> 75%
Medium	Testing tools	26 – 75%
Low	Conflict within team	< 25%

Table 5.4: Risk Probability definitions [?]

Impact	Value	Description
Very high	> 10%	Schedule impact or Unacceptable quality
High	5 –10%	Schedule impact or Some parts of the project have low quality
Medium	< 5%	Schedule impact or barely noticeable degradation in quality Low Impact on schedule or Quality can be incorporated

Table 5.5: Risk Impact definitions [?]

### 5.2.3 Overview of Risk Mitigation, Monitoring, Management

This section in detail describes Risk Mitigation, Monitoring and Management for each of the possible risks. It will talk about ways to avoid, monitor and to have ways to manage the risks.

#### **Mitigation:**

In this section several different software development risks will be identified; a plan will be created to avoid these risks. We will think of the risks possible and of the way to keep the software development process from encountering these risks. It is important to have mitigation plan to avoid risks once and for all. Goal is to attack the risk even before it comes into existence. The plan will help in identifying the possible risks and to monitor them.

#### **Monitoring:**

In this section we will identify the conditions to monitor to determine whether risk is becoming more or less likely.

#### **Management:**

In this section we will identify several different software development risks and will try to create a plan to manage these risks if they do occur.



Risk ID	1
Risk Description	Complexity of technology.
Category	Requirement
Source	Software requirement
Probability	Low
Impact	High
Response	Severe
Strategy	Avoid by opting user friendly techniques
Risk Status	Not Occurred

Table 5.6: Risk ID 1

Risk ID	2
Risk Description	Technical knowledge possessed by testing team.
Category	Testing Field
Source	Software Testing
Probability	High
Impact	High
Response	Mitigate
Strategy	Better testing will resolve this issue.
Risk Status	Identified

Table 5.7: Risk ID 2

Risk ID	3
Risk Description	Conflicts within the team.
Category	General
Source	May occur in any phase of project development
Probability	Low
Impact	Low
Response	Accept
Strategy	Try to accept everyone's point of view
Risk Status	Identified

Table 5.8: Risk ID 3

Risk ID	4
Risk Description	Teams being distributed over a large geographical area.
Category	Technology
Source	May happen there is no connectivity
Probability	Low
Impact	Low
Response	Accept
Strategy	Have good internet connection
Risk Status	Identified

Table 5.9: Risk ID 4

Risk ID	5
Risk Description	Usage of poor-quality testing tools.
Category	Technology
Source	This was identified during early testing.
Probability	Medium
Impact	High
Response	Accept
Strategy	Try to find good testing tools.
Risk Status	Identified

Table 5.10: Risk ID 5

## 5.3 Project Schedule

### 5.3.1 Project task set

Major Tasks in the Project stages are:

- Task 1:  
The first task was to install the required software and packages which are required for project development like Python 3, HTML5, CSS3, PyCharm, VS Code.
- Task 2:  
The second task was to test the project development environment by the running any database connectivity Python program.
- Task 3:  
In third task we split the project in function oriented sub-tasks by which object-oriented language features are implemented for easy development of project.
- Task 4:  
The fourth task is implementing sub-task by coding which are defined in previous step.
- Task 5:  
The fifth task was to integrate the all-sub- program and User interface together for further testing.

Task 6:

- The final task was to test the project in sequence. That all the program & function run successfully.

### 5.3.2 Task network

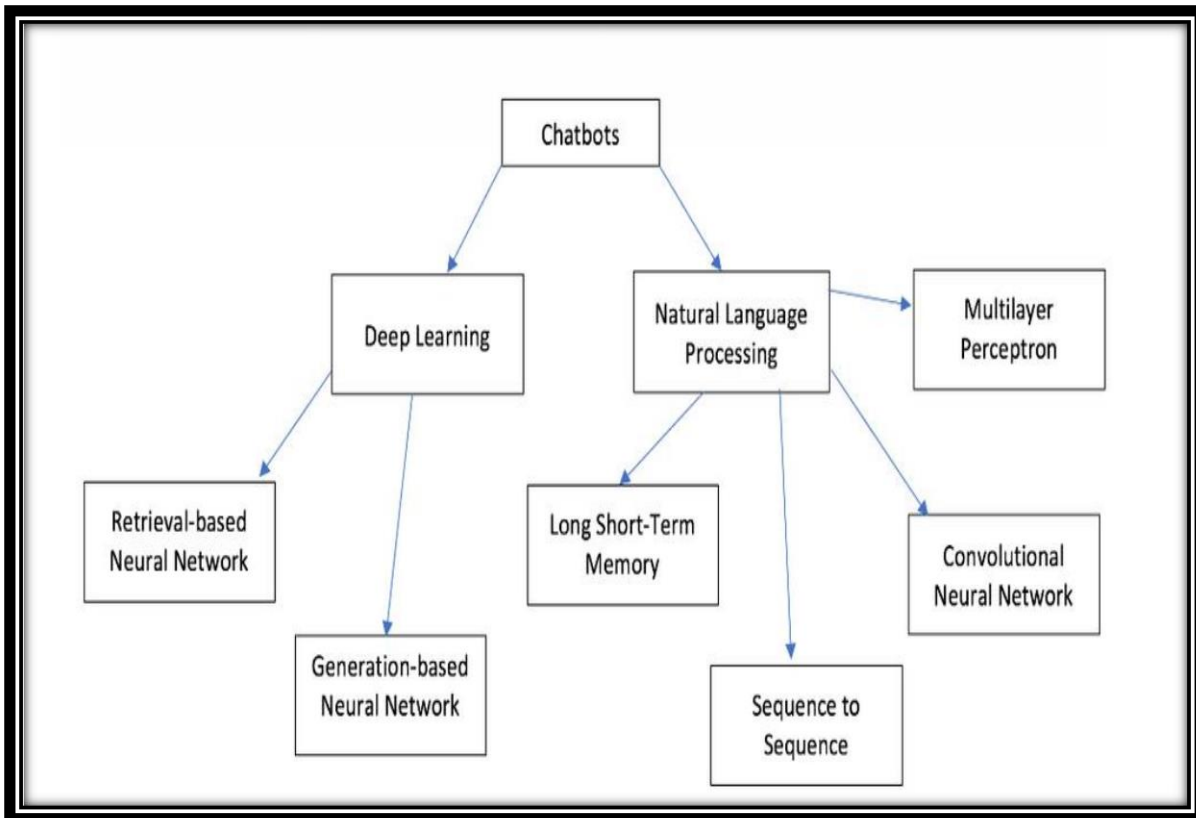


Figure 5.1: Task Network

## 5.4 Team Organization

All the four members were formed as a team and performed most of the tasks as a team.

### 5.4.1 Team structure

Various task and the teams were as follows:

SR. NO	PROJECT WORK	CONTRIBUTOR
1	Developing project idea	Sakshi, Dheeraj, Dhanashree, Abhishek
2	Literature Survey	Dheeraj, Sakshi, Abhishek, Dhanashree
3	System Description and synopsis	Dhanashree, Dheeraj
4	Requirement specification	Dheeraj, Sakshi, Dhanashree, Abhishek
5	Project plan specification	Sakshi, Abhishek
6	Software level coding	Abhishek, Sakshi, Dhanashree, Dheeraj
7	Test cases	Abhishek, Sakshi
8	Publishing paper	Dhanashree, Dheeraj, Sakshi, Abhishek
9	Project Report	Sakshi, Dheeraj

Table 5.11: Team Structure

### 5.4.2 Management reporting and communication

In each week we communicated with our guide and discussed the progress of the project. Various points discussed in the weeks were noted down and a separate log book is maintained.

**CHAPTER 6**

**SOFTWARE REQUIREMENT  
SPECIFICATION**

## **6.1 Introduction**

### **6.1.1 Purpose and Scope of Document**

To provide a detailed overview of our software product, its parameters and goals. This document describes its user interface, hardware and software requirements.

### **6.1.2 Overview of responsibilities of Developer**

The developer has to carry out the various functions. The developer first needs to figure out what all the requirements are. Then the developer has the responsibility to build the application and take various possible inputs from the network. Then the various inputs need to be store in the database.

## **6.2 Usage Scenario**

To model a system the most important aspect is to capture the dynamic behavior. To clarify a bit in details, dynamic behavior means the behavior of the system when it is running /operating. So only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior. So, use case diagrams are consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system. So, to model the entire system numbers of use case diagrams are used.

### **6.2.1 User profiles**

Chatbots improve the way we communicate with customers but also take out the personal touch from chats. The Users tool allows you to get closer to the people that chat with your bot and build relationships. Users can give lots of information on who is talking with your bot. Our System takes diagnostic information of the Patient (end User) in the form of query through text or voice from the web UI Interface and then converts it into the feed able data to the appropriate trained model which is built at the runtime and gives back the required solution(remedy/treatment) for the same in the form of Response Text.

### 6.2.2 Use Case View

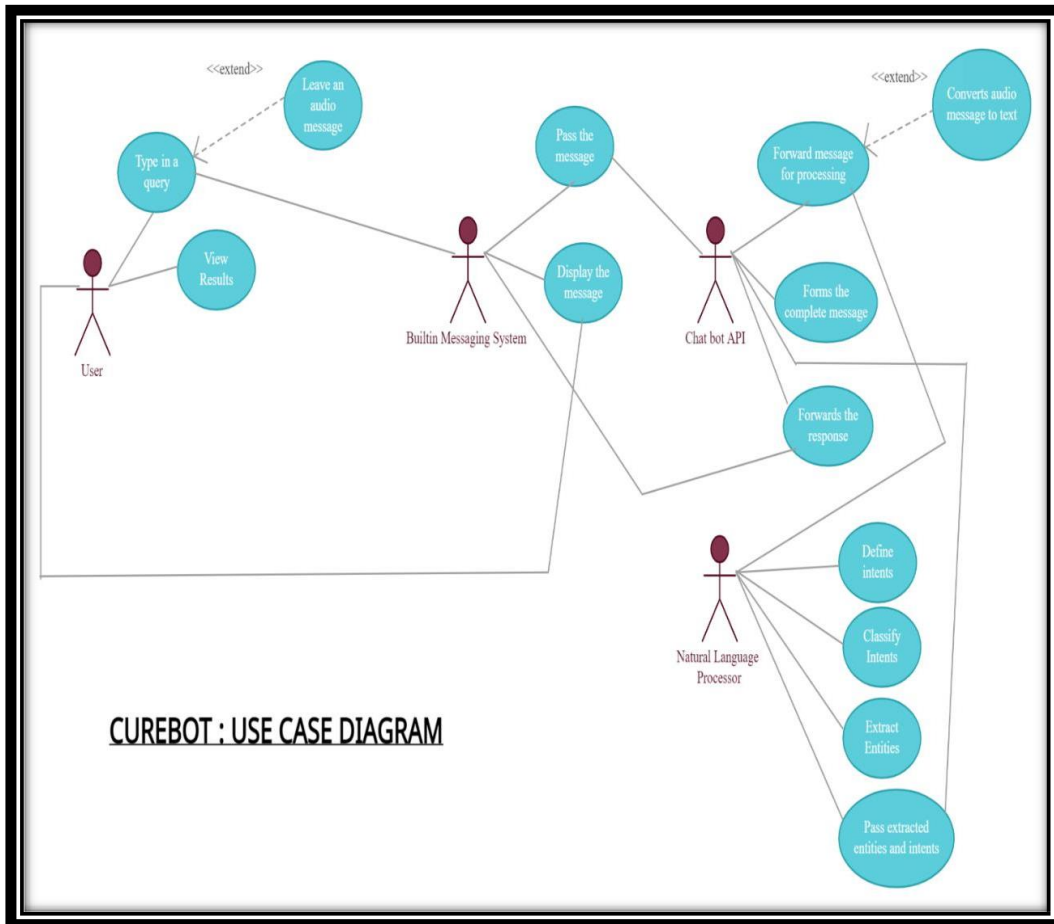


Figure 6.1: Use case diagram



## 6.3 Data Model and Description

### 6.3.1 Data Description

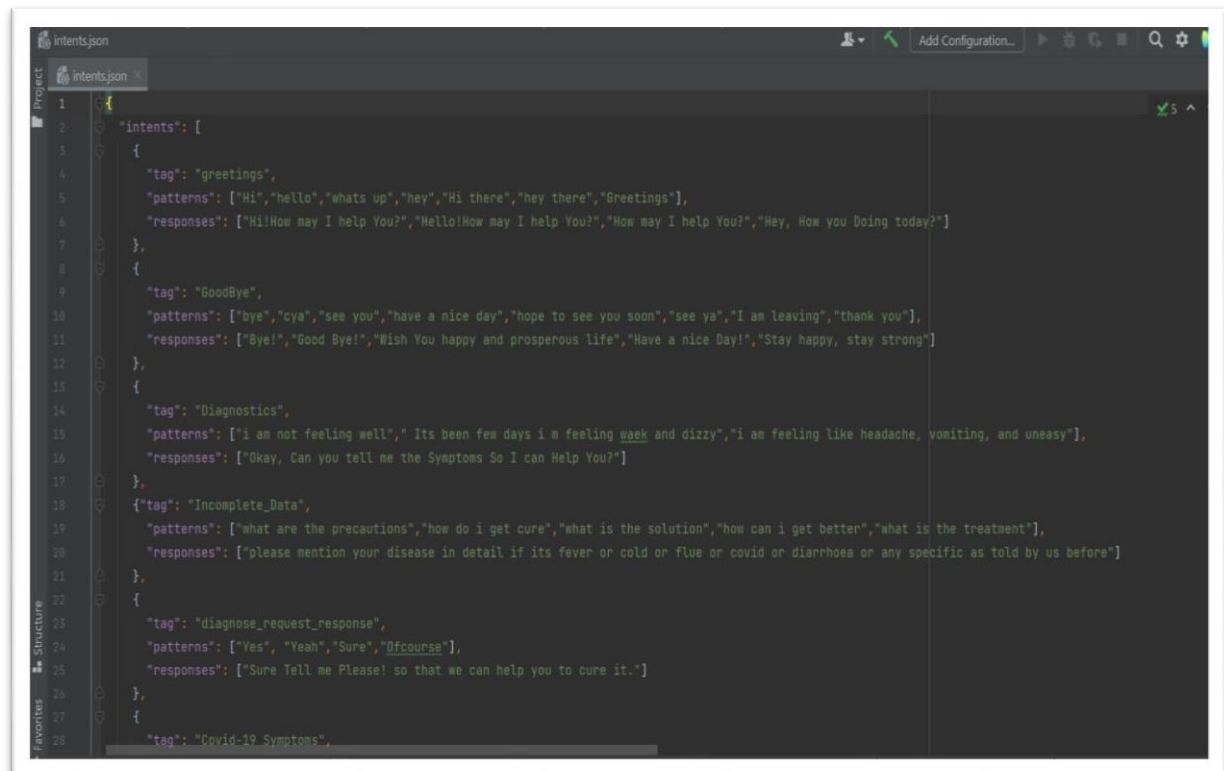


Figure 6.2: Creation of Dataset



Figure 6.3: data preparation

### 6.3.2 Data objects and Relationships

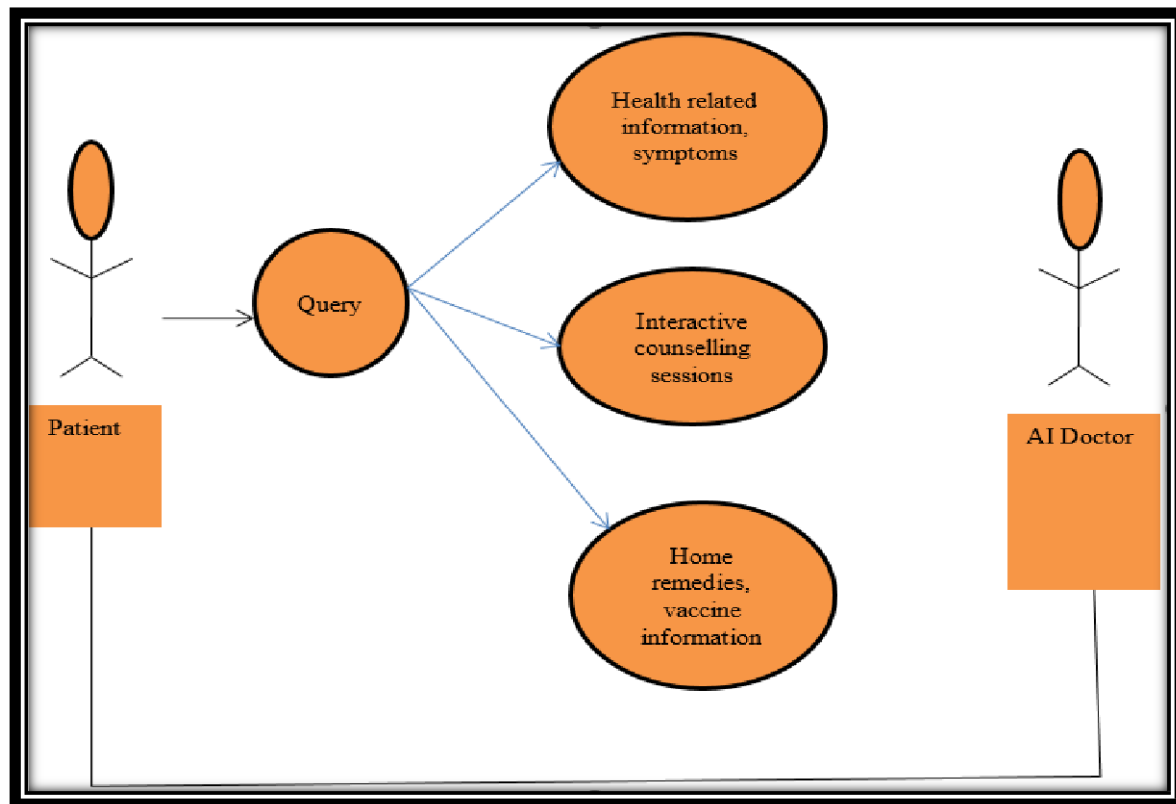


Figure 6.4: Packet Entity

## 6.4 Functional Model and Description

### 6.4.1 Data Flow Diagram

#### 6.4.1.1 Level 0 Data Flow Diagram

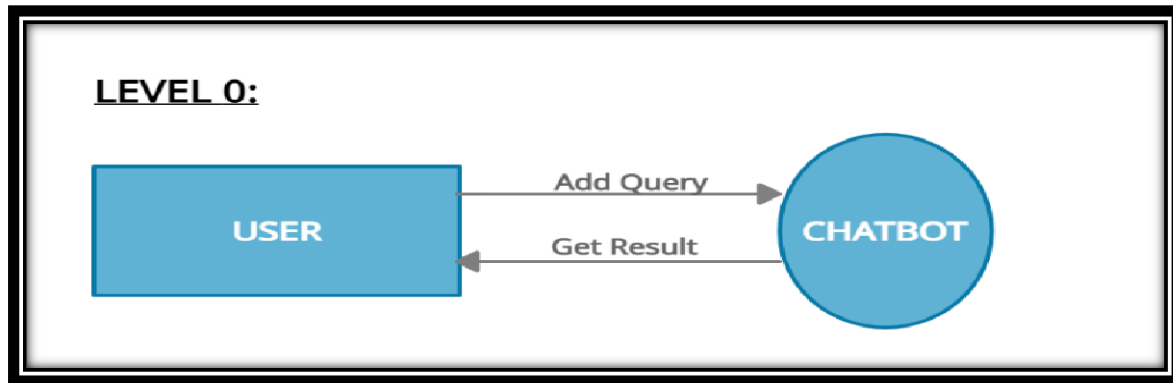


Figure 6.5: Data Flow Diagram Level 0

#### 6.4.1.2 Level 1 Data Flow Diagram

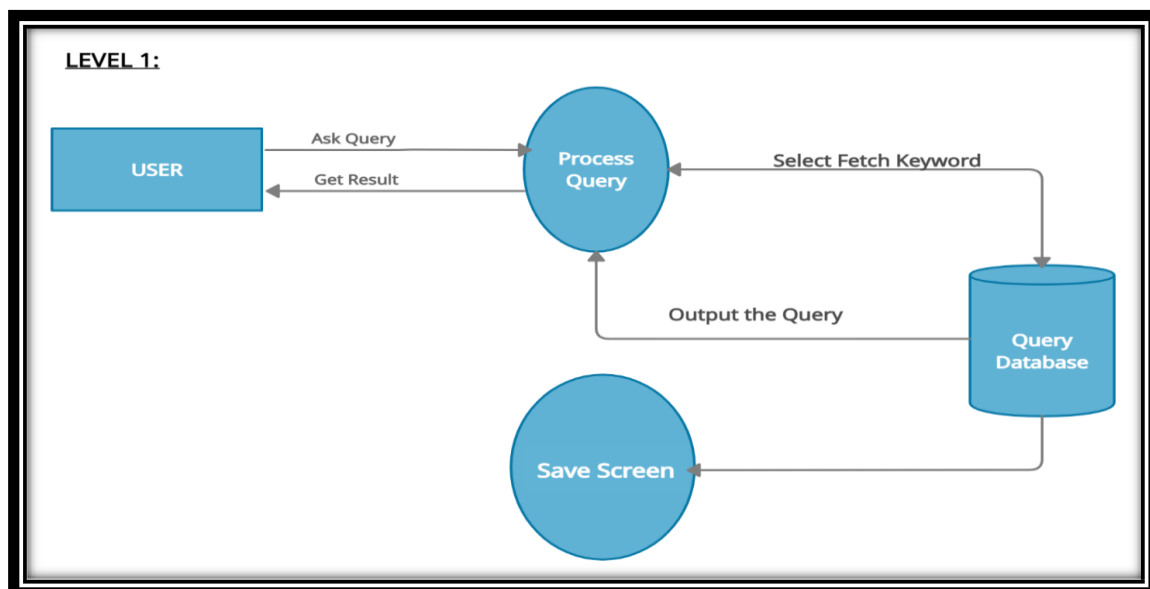


Figure 6.6: Data Flow Diagram Level 1

### 6.4.2 Activity Diagram:

Activity diagram is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. So, the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. Activity diagrams deal with all type of flow control by using different elements like fork, join etc. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another. It does not show any message flow from one activity to another. Activity diagram is some time considered as the flow chart. Although the diagrams look like a flow chart but it is not.

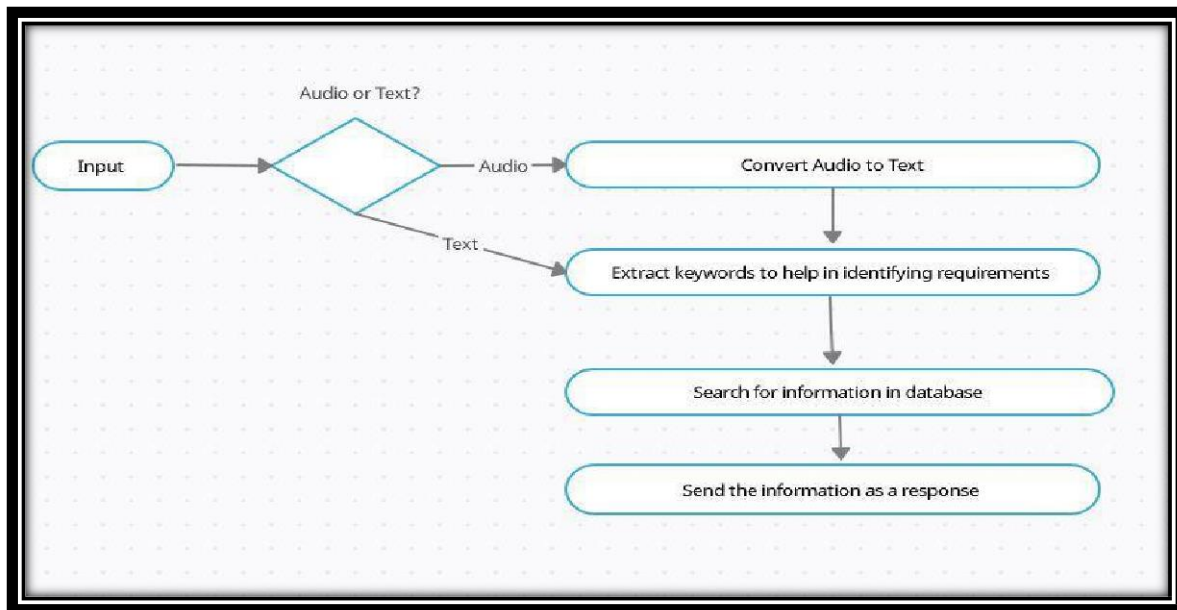


Figure 6.7: Activity diagram: Personal Query Response Activity (module-1)

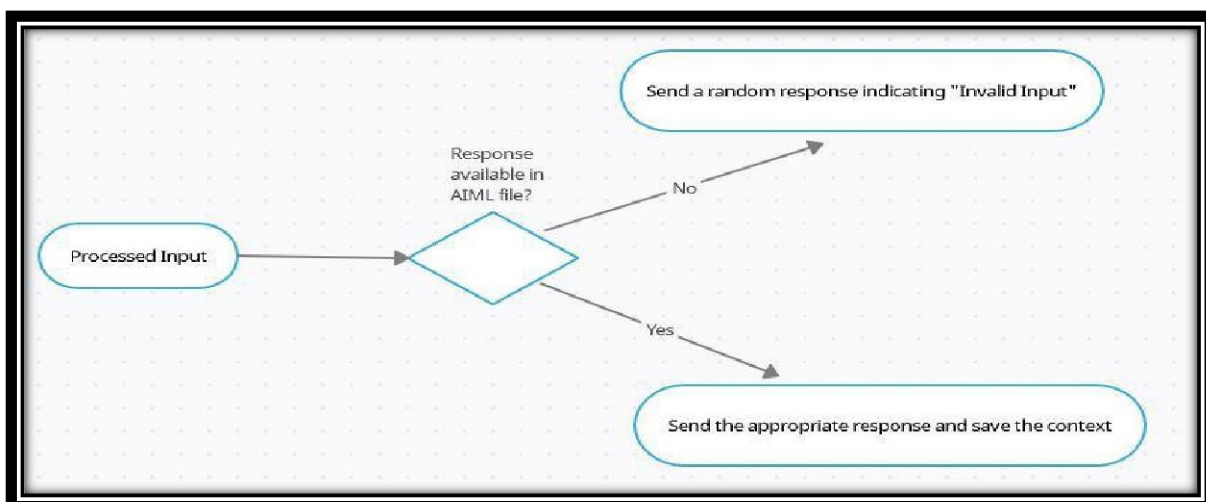


Figure 6.8: Activity diagram: Normal Conversation Response Activity (Module-2)

### 6.4.3 Non-Functional Requirements:

- **Security:** -

Any system developed should have a secured way to access its functionality. A system which can be accessed randomly without any check or verification cannot be relied on for saving your personal information/data. Hence, we have used a login page to validate the users using the software.

- **Reliability:** -

Reliability is defined as the ability of a system or a component to function understated conditions for a specified period of time. Any layman can operate the systems without knowing working details in his day-to-day life. User friendly UI makes it possible.

- **Maintainability:** -

Maintainability is the ability with which a product or an application is maintained in order to isolate and correct or remove the defects occurred in the application. Our project is modular in nature, maintaining the software is very easy. If any problem arises one has to check the related module instead of checking the whole code.

- **Portability:** -

Portability is the ability of an application which can be used in different environments without making any changes to the application or environment itself. Our application can work on multiple versions operating system Windows, MAC, Linux. Hence, portable.

- **Extensibility:** -

Extensibility is the ability of an application which can be further implemented taking future growth into consideration. Our project is divided into modules and hence, the new and better functionality can be added or modified in the applications easily.

- **Reusability:** -

Reusability is the ability of a software or an application in which the existing functionality of that application can be developed and further can be used for different purpose. We have made use of Python which makes our software reusable. The Object- oriented approach helps to make our system highly reusable.

- **Compatibility:** -

Compatibility is the ability of a software to run on different computer architectures without changing the components of that computer.

- **Serviceability:** -

Serviceability is the ability of technical support to install, configure and monitor software products and identify faults or bugs and provide maintenance in pursuit of solving a problem and restoring the product into service. We have made use of Python 3, HTML5, CSS3.

#### 6.4.4 Sequence Diagram:

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

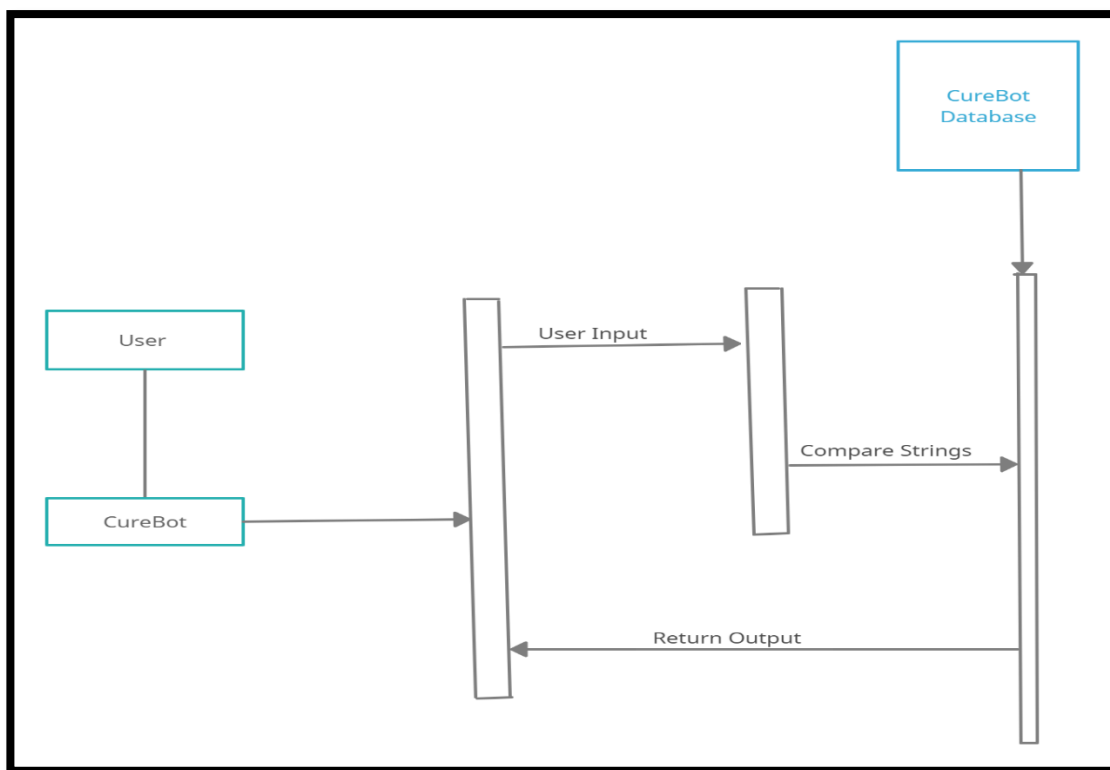


Figure 6.9: Sequence diagram

### **6.4.5 Design Constraints**

In system design, a design constraint refers to some limitation on the conditions under which a system is developed, or on the requirements of the system. The design constraint could be on the systems form, fit or function or could be in the technology to be used, materials to be incorporated, time taken to develop the system, overall budget, and so on. During system design, it is as important to identify each design constraint as it is to elicit requirements since the design constraints place an overall boundary around the system design process.

- While storing the data on the database, some irrelevant data may get added in the database. The irrelevant data can be called as dirty bit.
- The bot cannot do physical operations. It means it cannot help user the way doctor can heal the patient.
- The bot does not work without the user prompting first. The user has to convey his/ her message first.
- The regular Internet connectivity is necessary for the bot to solve the query of the patient.

### **6.4.6 Software Interface Description**

#### **6.4.6.1 Graphical User Interface**

- User Friendly UI for user.
- Interactive access to the application.
- Create a personalized branded experience.
- Communicate with the user in a humanlike way.
- Assist user as a concerned one do.

#### **6.4.6.2 Training Dataset**

- Training data is a collection of labeled information that's used to build a machine learning model.
- Through training data, an AI model learns to perform its task at a high level of accuracy.
- Training data is the textbook that will teach your model to do its assigned task.



**CHAPTER 7**  
**DETAILED DESIGN DOCUMENT**

## 7.1 Introduction

The Chatbot is a software programs that is used to interact with clients using natural language Processing via text or text to speech format. Initially, chatbot created for interact with human for entertainment purpose only. 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. This inaccessibility of healthcare facilities especially in rural areas and the difficulty in accessing means of transport and it causes patients to postpone their treatment, or option for medical facilities which is closer but at the same time are not cost-efficient and well-matched to their medical needs. The efficient ways to provide timely medical care, access and quality of the treatment to the patient, the role of Medical Chatbot comes into play which connects patients with the chatbot and help them to give the correct antibiotics/ medicines and precautions.

**The typical interaction b/w humans and chatbot using Natural Language Processing (NLP) as follows:**

1. A human interact with chatbot
2. The chatbot captures the audio of the human
3. The chatbot converts the Audio to text.
4. The Chatbot Process the text's data.
5. Data/text to audio conversion takes place in the chatbot.
6. The chatbot responds to the human in audio format.

**Deep Learning** is a subset of machine learning in Artificial Intelligence concerned with algorithms capable of learning unsupervised from data which is unstructured and unlabeled. It uses a function of the brain called **neural networks**.

The Steps for creating a chatbot:

1. Libraries & Data which we include
2. Initializing Chatbot Training Model
3. Building the Deep Learning Model
4. Building GUI for the chatbot
5. Running the Chatbot or producing Output.

**Libraries & Data: -**

● **train\_chatbot.py** — the code is used for reading the human language data into the training set. By using a Keras sequential neural network to create the model.

● **chatgui.py** — the code used for cleaning the responses which is based on the predictions from the model and creates a graphical interface for interacting with the chatbot.

- **classes.pkl** — a list of different types of classes of the responses given by chatbot.
- **words.pkl** — a list of the words that could be used for the pattern recognition.
- **intents.json** — a bunch of JavaScript objects that lists different tags and correspond to different types of word patterns.
- **chatbot\_model.h5** — the final model which is created by train\_chatbot.py and used by chatgui.py.

### **Neural Network:**

Neural network is a sequence of algorithms that try to build the relationships between the data and the human brain. Neural networks adapt the changing input so, the network generates the best possible output without needs to redesign again.

### **Sequential model:**

The Sequential model is a stack of different layers where every layer has only one input and one output tensor.

### **Modules:**

#### TensorFlow:

It is an end-to-end Open-Source platform for Machine Learning. It has a different tools, libraries and community resources.

#### Speech\_Recognition:

It is an automatic speech recognition (ASR). It has the power which enables a program to process human audio into a written format.

#### Natural Language Toolkit (NLTK) Libraries:

It contains libraries and programs for Natural language processing. It is the most powerful NLP libraries, which contains packages to make chatbot understand human audio and reply to it with an accurate response.

#### flask module

Flask is a lightweight WSGI web application framework. It is built to get started quick and easy, with the capability to scale up to the complex applications.

#### Keras

It is a deep learning API written in Python language, running on the top of the machine learning platform i.e., Tensor flow. It is used to create layers in Neural Network.

## 7.2 Architectural Design

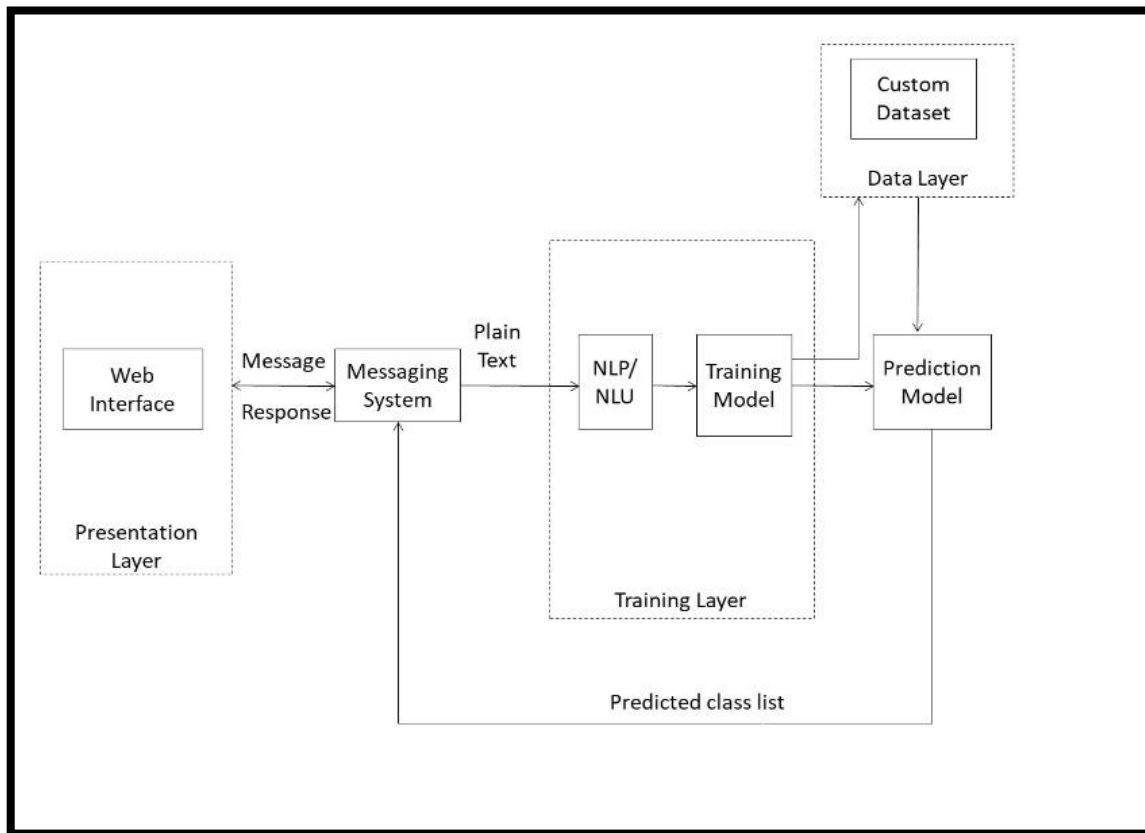


Figure 7.1: Architecture diagram

### 7.2.1 Overall Description

#### 7.2.1.1 Objectives:

- To Provide Home Remedies/Primary treatment for common Diseases.
- To provide information about Majority of Prevalent Diseases in India.
- To provide precautions to be taken for COVID-19.
- To provide information about the Diet plan in case of any Disease.

#### 7.2.1.2 User Perspective:

Our System takes diagnostic information of the Patient (end User) in the form of query through text or voice from the web UI Interface and then converts it into the feed able data to the appropriate trained model which is built at the runtime and gives back the required solution (remedy/treatment) for the same in the form of Response Text.

## 7.3 Data design

### 7.3.1 Define Intents

An “**intent**” is the intention of the user interacting with a chatbot or the intention behind each message that the chatbot receives from a particular user.

### 7.3.2 Data Preparation

The variable “**words**” holds all the tokenized training data (which are sample messages in each intent category/tag), the “**classes**” variable holds all target labels/tags correspond to each training data and the “**documents**” variable holds both training data (words of sample messages tokenized using “**word\_tokenize(pattern)**”) and labels correspond to each training data. After this, eliminate symbols, defined in “**ignore\_letters**” variable, from training data and also eliminate stop words, if required.

### 7.3.3 Create bag of words

Convert text data into numeric format (0 and 1 ). Create bag of words, such as if word is present in “**word\_patterns**” append 1, otherwise append 0 to the list.

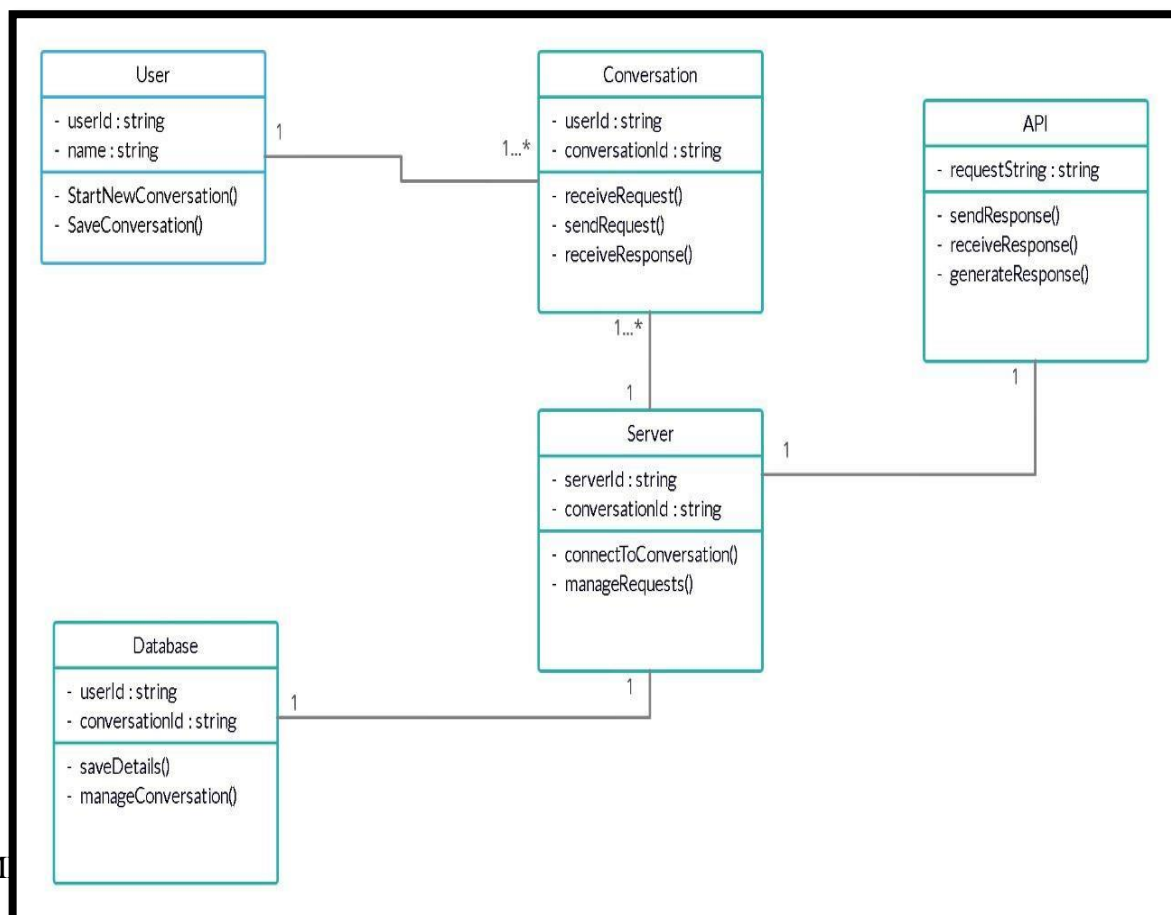
### 7.3.4 Model training

Define Neural Network architecture for proposed model and for that use the “**Sequential**” model class of Keras. Keras is an open source, high level library for developing neural network model.

## 7.4 Component Design

### 7.4.1 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 objects. In the diagram, classes are represented with boxes which contain three parts as follows the top part contains the name of the class, the middle part contains the attributes of the class and the bottom part contains the methods the class can execute.



RM

# **CHAPTER 8**

## **PROJECT IMPLEMENTATION**

## 8.1 Introduction

The project requires multiple tools for implementation but for user end it only requires a Python 3, HTML5 and CSS3 installed PC. The various tools and flow chart of the project is been elaborated as follows:

## 8.2 Tools and Technologies Used

### 8.2.1 Software used

#### 8.2.1.1 Python 3: -

Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. Python 3.0 was released in 2008. Although this version is supposed to be backward incompatible, later on many of its important features have been backported to be compatible with version 2.7.

#### **Characteristic: -**

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

#### **Need: -**

- Used for Backend programming.
- Used to create sequential model.
- Used to predict the class.
- Used to get user input and return the output.

#### 8.2.1.2 PyCharm: -

**PyCharm** is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains (formerly known as **IntelliJ**) **PyCharm** is cross-platform, with Windows, macOS and Linux versions. PyCharm is the most popular IDE used for Python scripting language. This chapter will give you an introduction to PyCharm and explains its features.



PyCharm offers some of the best features to its users and developers in the following aspects –

- Code completion and inspection
- Advanced debugging
- Support for web programming and framework Flask.

**Need:**

- To develop the User Interface with the use of PyCharm Editor.
- It Support the web programming and framework Flask.

**8.2.1.3 Flask: -**

Flask is a micro web framework written in Python. ...Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Flask is part of the categories of the micro-framework. Micro-framework is normally framework with little to no dependencies to external libraries. This has pros and cons. Pros would be that the framework is light, there are little dependency to update and watch for security bugs, cons is that some time you will have to do more work by yourself or increase yourself the list of dependencies by adding plugins

**Need:**

- To render HTML file/code.
- To define http methods like GET/POST.

**8.2.1.4 HTML5: -**

HTML5 is the next major revision of the HTML standard superseding HTML 4.01, XHTML 1.0, and XHTML 1.1. HTML5 is a standard for structuring and presenting content on the World Wide Web. HTML5 is a cooperation between the World Wide Web Consortium (W3C) and the Web Hypertext Application Technology Working Group (WHATWG).

**Features: -**

- New Semantic Elements
- Forms 2.0
- Persistent Local Storage
- WebSocket
- Server-Sent Events
- Canvas
- Audio & Video

**Needs:**

- To develop a frontend.
- To create a form for getting user input.

#### 8.2.1.5 CSS3: -

Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language. CSS3 is collaboration of CSS2 specifications and new specifications, we can call this collaboration is module.

##### **Modules: -**

- Selectors
- Box Model
- Backgrounds
- Image Values and Replaced Content
- Text Effects
- 2D Transformations
- 3D Transformations
- Animations
- Multiple Column Layout
- User Interface

##### **Need:**

- To define styles.

#### 8.2.2 Hardware specification

Sr. No.	Parameter	Minimum Requirement	Justification
1	Storage	4 GB	For smooth performance
2	Processor	i3	Easy for multitasking
3	Internet	LAN/WIFI	For connection to internet

Table 8.1: Hardware specification

#### 8.2.3 Components

##### **Software: -**

- OS(Linux/Windows)
- Programming Language (Backend): -Python3,
- Editors: - VS Code,
- Front-end: - Flask (for UI), HTML, CSS
- Network Required: -
- Ethernet (LAN)
- Wireless N/W (WIFI)

## 8.3 Methodologies

### 8.3.0.1 Project Flow

#### 1. Define Intents:

An “**intent**” is the intention of the user interacting with a chatbot or the intention behind each message that the chatbot receives from a particular user [1]. Define simple intents and set of messages that corresponds to those intents and also map some responses according to each intent category. Create a JSON file named “**intents.json**”, which is in the form of categories provided with some tags. Define set of patterns with the corresponding responses. Following is the sample dataset:

```
{
  "intents": [
    {
      "tag": "greetings",
      "patterns": ["Hi", "hello", "whats up", "hey", "Hi there", "hey there", "Greetings"],
      "responses": ["Hi!", "Hello!", "How may I help You?", "Hey, How you Doing?", "Good Day"]
    },
    {
      "tag": "GoodBye",
      "patterns": ["bye", "cya", "see you", "have a nice day", "hope to see you soon", "see ya", "I am leaving"],
      "responses": ["Bye!", "Good Bye!", "Wish You the Same, Have a nice Day!", "Sad to see you go :("]
    },
    {
      "tag": "Covid-19 Symptoms",
      "patterns": ["I have cough, cold and feeling fever with a sore throat"],
      "responses": ["Your Symptoms Denote to COVID-19. Please Ensure your RIPCR Test from nearest clinic"]
    },
    {
      "tag": "Fever",
      "patterns": ["i have a temperature of 101 degrees", "i am feeling cold", "I have fever"],
      "responses": ["You Have fever"]
    }
  ]
}
```

Figure 8.1: Defining Intent

#### 2. Data Preparation:

Load the json file and extract the required data.

```
intents = json.loads(open('intents.json').read())
|
words = []
classes = []
documents = []
ignore_letters = ['?', ' ', '!', ' ', '!', ' ', '!', ' ']

for intent in intents['intents']:
    for pattern in intent['patterns']:
        word_list = nltk.word_tokenize(pattern)
        words.extend(word_list)
        documents.append((word_list, intent['tag']))
        if intent['tag'] not in classes:
            classes.append(intent['tag'])
```

Figure 8.2: Data Preparation

The variable “**words**” holds all the tokenized training data (which are sample messages in each intent category/tag), the “**classes**” variable holds all target labels/tags correspond to each training data and the “**documents**” variable holds both training data (words of sample messages tokenized using “**word\_tokenize(pattern)**”) and labels correspond to each training data. After this, eliminate symbols, defined in “**ignore\_letters**” variable, from training data and also eliminate stop words, if required.

### 3. Create bag of words

Convert text data into numeric format (0 and 1). Create bag of words, such as if word is present in “**word\_patterns**” append 1, otherwise append 0 to the list.

```
for document in documents:
    bag = []
    word_patterns = document[0]
    word_patterns = [lemmatizer.lemmatize(word.lower()) for word in word_patterns]
    for word in words:
        bag.append(1 if word in word_patterns else bag.append(0))
```

Figure 8.3: Bag of words

### 4. Model training:

Define Neural Network architecture for proposed model and for that use the “**Sequential**” model class of Keras. Keras is an open source, high level library for developing neural network model [2].

The steps for creating a Keras (Sequential) model are the following:

**Step 1:** Define neural network model (i.e., Sequential model). The network is defined as a sequence of layers, each with its own customizable size and activation function. In this sequential model, the first layer is the input layer, which define the size of the input, which is feeded to neural network. After this more and more layers (hidden layers) can be added and customized until reached to the final output layer.

```
model = Sequential()
model.add(Dense(128, input_shape=(len(train_x[0]),), activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(len(train_y[0]), activation='softmax'))
```

Figure 8.4: Model Training

**Step 2:** Define the optimization algorithm that will be used to train the sequential model, for that use “**SGD**” optimizer and also choose the loss function. Then, compile the neural network, which transforms the simple sequence of layers into a complex group of matrix operations that describes the behavior of the network.

```
sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd, metrics=['accuracy'])
```

**Step 3:** Train or fit the neural network and save it in “chatbot\_model.h5” file.

```
hist = model.fit(np.array(train_x), np.array(train_y), epochs=500, batch_size=5, verbose=1)
model.save('chatbot_model.h5', hist)
```

Figure 8.6: Chatbot model

**Step 4:** Now, model/network is trained, and can be used to make predictions.

#### 5. Get User input:

Accept input query/message from user with developed frontend using **Flask** framework.

#### 6. Predict class for user input:

Filter out the class/tag for user input, with specified threshold and make list of all filtered tags. Sort the list in reverse order, so that it returns class/tag, which has the highest probability. Finally, return the list, with classes and their corresponding probability.

```
def predict_class(sentence, model):
    # Filter out predictions below a threshold
    p = bow(sentence, words, show_details=False)
    res = model.predict(np.array([p]))[0]
    ERROR_THRESHOLD = 0.25
    results = [(i, r) for i, r in enumerate(res) if r > ERROR_THRESHOLD]
    # sort by strength of probability
    results.sort(key=lambda x: x[1], reverse=True)
    return_list = []
    for r in results:
        return_list.append({"intent": classes[r[0]], "probability": str(r[1])})
    return return_list
```

Figure 8.7: Class prediction

#### 7. Get Bot response:

Store the class/tag, returned by previous function, having the highest probability. Compare that class/tag, with tags defined in “**intents.json**” file. Return the response corresponding to that class/tag.

```
def getResponse(ints, intents_json):
    tag = ints[0]["intent"]
    list_of_intents = intents_json["intents"]
    for i in list_of_intents:
        if i["tag"] == tag:
            result = random.choice(i["responses"])
            break
    return result
```

Figure 8.8: Bot response

## **8. Display Bot Response:**

Display bot response using **Flask** framework.

# **8.4 Verification and Validation for Acceptance**

## **8.4.1 Verification**

In verification the aim was to ensure that the product is being built according to the requirements and design specifications. In other words, to ensure that work products meet their specified requirements.

1. We evaluated our project with the help Plans, Requirement Specifications, Design Specifications, Code, Test Cases and verified that we were building the product right.
2. In verification process we also performed activities like:
  - Reviews
  - Walkthroughs
  - Inspections

## **8.4.2 Validated**

In validation the aim was to ensure that the product actually meets the user's needs, and that the specifications were correct in the first place. In other words, to demonstrate that the product fulfills its intended use when placed in its intended environment.

1. We performed validation on the project after it was completely built.
2. In order to perform validation, we carried out different types of testing on our project.
  - GUI Testing
  - Component Interface Testing
  - Unit Testing
  - Integration Testing
  - Black Box Testing
  - White Box Testing
  - Alpha Testing
  - Beta Testing



# **CHAPTER 9**

## **SOFTWARE TESTING**

# SOFTWARE TESTING: -

The definition of testing is "the process of questioning a product in order to evaluate it", where the "questions" are things, the tester tries to do with the product, and the product answers with its behavior in reaction to the probing of the tester. Although most of the intellectual processes of testing are nearly identical to that of review or inspection, the word testing is connoted to mean the dynamic analysis of the product putting the product through its paces.

## 9.1 Type of Testing Used

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 an investigation conducted to provide stake- holders with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use.

For testing of the project, the unit and integration testing are been performed.

## 9.2 Test Cases and Test Results

### 9.2.1 Unit Testing

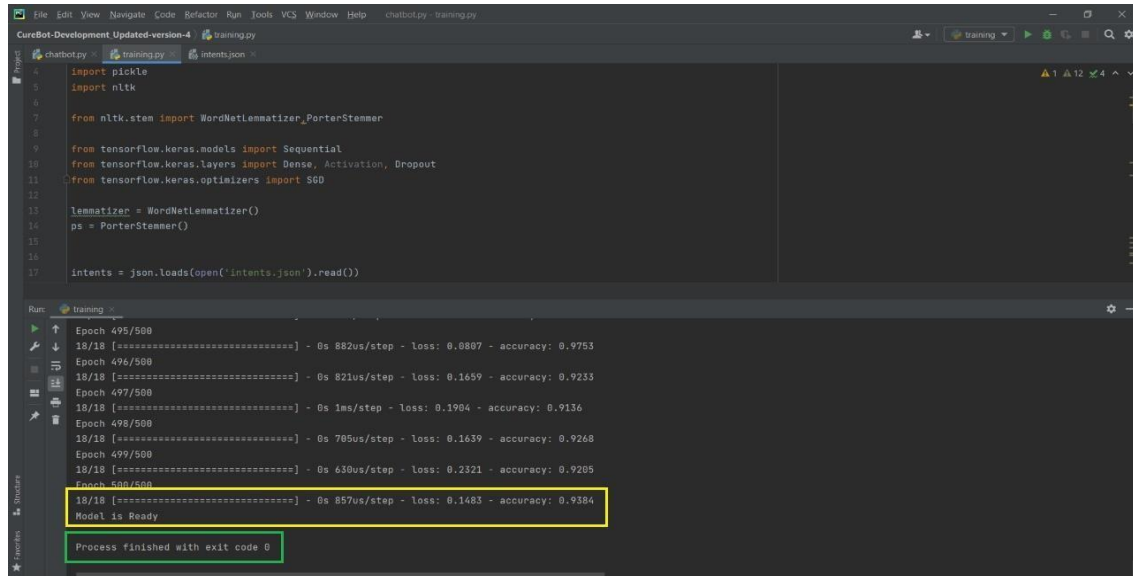
Unit testing, also known as component testing, refers to tests that verify the functionality of a specific section of code, usually at the function level. In an object-oriented environment, this is usually at the class level, and the minimal unit tests include the constructors and destructors.

These types of tests are usually written by developers as they work on code (white-box style), to ensure that the specific function is working as expected. One function might have multiple tests, to catch corner cases or other branches in the code. Unit testing alone cannot verify the functionality of a piece of software, but rather is used to ensure that the building blocks of the software work independently from each other. Unit testing is a software development process that involves synchronized application of a broad spectrum of defect prevention and detection strategies in order to reduce software development risks, time, and costs. It is performed by the software developer or engineer during the construction phase of the software development lifecycle. Rather than replace traditional QA focuses, it augments it. Unit testing aims to eliminate construction errors before code is promoted to QA; this strategy is intended to increase the quality of the resulting software as well as the efficiency of the overall development and QA process.

**Test case 1:** To test the training.py file

**Expected output:** The "training.py" file should run successfully without any error.

**Output observed:**



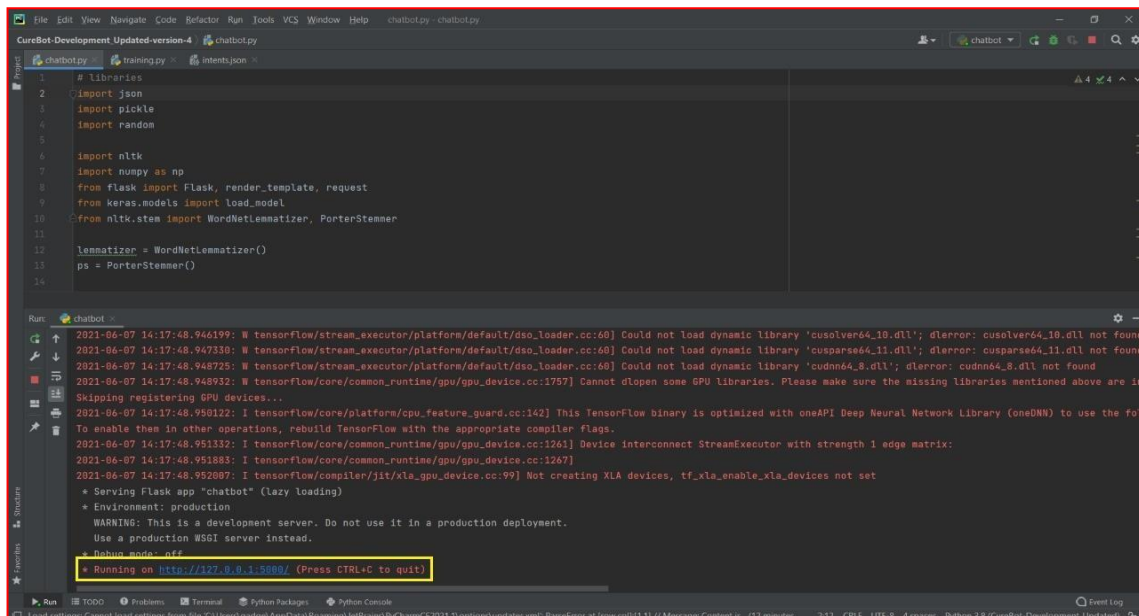
```
File Edit View Navigate Code Refactor Run Tools VCS Window Help chatbotpy training.py
CureBot-Development Updated-version-4 training.py
Project chatbotpy training.py intents.json
1 import pickle
2 import nltk
3
4 from nltk.stem import WordNetLemmatizer, PorterStemmer
5
6 from tensorflow.keras.models import Sequential
7 from tensorflow.keras.layers import Dense, Activation, Dropout
8 from tensorflow.keras.optimizers import SGD
9
10 lemmatizer = WordNetLemmatizer()
11 ps = PorterStemmer()
12
13 intents = json.loads(open('intents.json').read())
14
15
16
17
18 Run training
19 Epoch 495/500
20 18/18 [=====] - 0s 882us/step - loss: 0.0807 - accuracy: 0.9753
21 Epoch 496/500
22 18/18 [=====] - 0s 821us/step - loss: 0.1659 - accuracy: 0.9233
23 Epoch 497/500
24 18/18 [=====] - 0s 1ms/step - loss: 0.1904 - accuracy: 0.9136
25 Epoch 498/500
26 18/18 [=====] - 0s 705us/step - loss: 0.1639 - accuracy: 0.9268
27 Epoch 499/500
28 18/18 [=====] - 0s 630us/step - loss: 0.2321 - accuracy: 0.9205
29 Epoch 500/500
30 18/18 [=====] - 0s 857us/step - loss: 0.1483 - accuracy: 0.9384
31 Model is Ready
32 Process finished with exit code 0
```

**Result:** PASS

**Test case 2:** To test the chatbo.py file

**Expected output:** The "chatbo.py" file should run successfully without any error.

**Output observed:**



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help chatbotpy chatbotpy
CureBot-Development Updated-version-4 chatbotpy
Project chatbotpy training.py intents.json
1 # libraries
2 import json
3 import pickle
4 import random
5
6 import nltk
7 import numpy as np
8 from flask import Flask, render_template, request
9 from keras.models import load_model
10 from nltk.stem import WordNetLemmatizer, PorterStemmer
11
12 lemmatizer = WordNetLemmatizer()
13 ps = PorterStemmer()
14
15
16
17 Run chatbot
18 2021-06-07 14:17:48.946199: W tensorflow/stream_executor/platform/default/dso_loader.cc:60] Could not load dynamic library 'cudnn64_8.dll'; dlopen error: cudnn64_8.dll not found
19 2021-06-07 14:17:48.947338: W tensorflow/stream_executor/platform/default/dso_loader.cc:60] Could not load dynamic library 'cuspars64_11.dll'; dlopen error: cuspars64_11.dll not found
20 2021-06-07 14:17:48.948725: W tensorflow/stream_executor/platform/default/dso_loader.cc:60] Could not load dynamic library 'cudnn64_8.dll'; dlopen error: cudnn64_8.dll not found
21 2021-06-07 14:17:48.948932: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1757] Cannot dlopen some GPU libraries. Please make sure the missing libraries mentioned above are in
22 Skipping registering GPU devices...
23 2021-06-07 14:17:48.950122: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the fol
24 To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
25 2021-06-07 14:17:48.951332: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1261] Device interconnect StreamExecutor with strength 1 edge matrix:
26 2021-06-07 14:17:48.951883: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1267]
27 2021-06-07 14:17:48.952007: I tensorflow/compiler/jit/xla_gpu_device.cc:99] Not creating XLA devices, tf_xla_enable_xla_devices not set
28 * Serving Flask app "chatbot" (lazy loading)
29 * Environment: production
30 WARNING: This is a development server. Do not use it in a production deployment.
31 Use a production WSGI server instead.
32 * Debug mode: off
33 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

**Result: *PASS***

**Test case 3:** To test the successful launch of the GUI.

**Expected output:** The successful launch of the GUI after the completion of “chatbo.py file”.

**Output observed:**

**Result:** *PASS*

### **9.2.2 Integration Testing**

Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together (“bigbang”). Normally the former is considered a better practice since it allows interface issues to be located more quickly and fixed.

Integration testing works to expose defects in the interfaces and interaction between integrated components (modules). Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system.

Integration testing (integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements.

Integration testing is the second level of the software testing process comes after unit testing. In this testing, units or individual components of the software are tested in a group. The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units.

**Test case 4:** To test the initial response of the bot.

**Expected output:** The bot should give correct output to the query fired by the user.

**Output observed:**

**Result:** *PASS*

**Test case 5:** To test the Front End and back-End connection on firing query.

**Expected Output:** The bot should Successfully connect and give response to the query.

**Output Observed:**

**Result:** *PASS*

**Test case 6:** To test the response of the bot on Welcome and ending query.

**Expected output:** The bot should give correct output to the Welcome and ending query fired by the user.

**Output observed:**

**Result:** *PASS*

### 9.2.3 System testing

System testing, or end-to-end testing, tests a completely integrated system to verify that it meets its requirements. For example, a system test might involve testing a logon interface, then creating and editing an entry, plus sending or printing results, followed by summary processing or deletion (or archiving) of entries, then log off.

In addition, the software testing should ensure that the program, as well as working as expected, does not also destroy or partially corrupt its operating environment or cause other processes within that environment to become inoperative (this includes not corrupting shared memory, not consuming or locking up excessive resources and leaving any parallel processes unharmed by its presence).

System Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements. It has both functional and non-functional testing. System Testing is a black-box testing.

System Testing is usually carried out by a team that is independent of the development team in order to measure the quality of the system unbiased. It includes both functional and Non-Functional testing.

**Test case 7:** To test the bugs i.e., response to an empty query fired by the user.

**Expected output:** The bot should give correct output to the empty query fired by the user.

**Output observed:**

**Result:** *PASS*

**Test case 8:** To test the response to illogical query

**Expected output:** The bot should give correct output to the illogical query fired by the user.

**Output observed:**

**Result:** *PASS*

68

**Test case 9:** To test the bot response on the symptoms and identifying the disease.

**Expected output:** The bot should give correct output to the symptoms provided by the user and identify the correct disease.

**Output observed:**

**Result:** *PASS*

#### **9.2.4 Functional Testing**

Functional testing is to testing the functionality of the software application under test. Basically, it is to check the basic functionality mentioned in the functional specification document. Also check whether software application is meeting the user expectations. We can also say that checking the behavior of the software application against test specification.

This type of testing is mandatory and irrespective of what type of application this should be exercised. In Functional testing, we need check each component are functioning as expected or not, so it is also called as Component Testing.

Functional testing is a software testing process used within software development in which software is tested to ensure that it conforms with all requirements. Functional testing is a way of checking software to ensure that it has all the required functionality that's specified within its functional requirements.

Functional testing is a quality assurance process and a type of black-box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered.



**Test case 10:** To test the "Send" button.

**Expected output:** The query is fired only after pressing "Send" button.

**Result:** *PASS*

**Test case 11:** To test the "Voice" button.

**Expected output:** The "Voice" button should enable after pressing "Voice" icon.

**Output observed:**

**Result:** *PASS*

**Test case 12:** To test the message box.

**Expected output:** The query is typed using keyboard in the message box.

**Output observed:**

**Result:** *PASS*

# **CHAPTER 10**

## **RESULTS**

## 10.1 Screen shots

Question

slurred\_speech ?

Question

Diagnosis

Please Click on Yes or No for the Above symptoms in Question

Talk

No

Clear

Yes

Start

11 RMDSSOE , Department of computer Engineering

## 11.1 Outputs: -

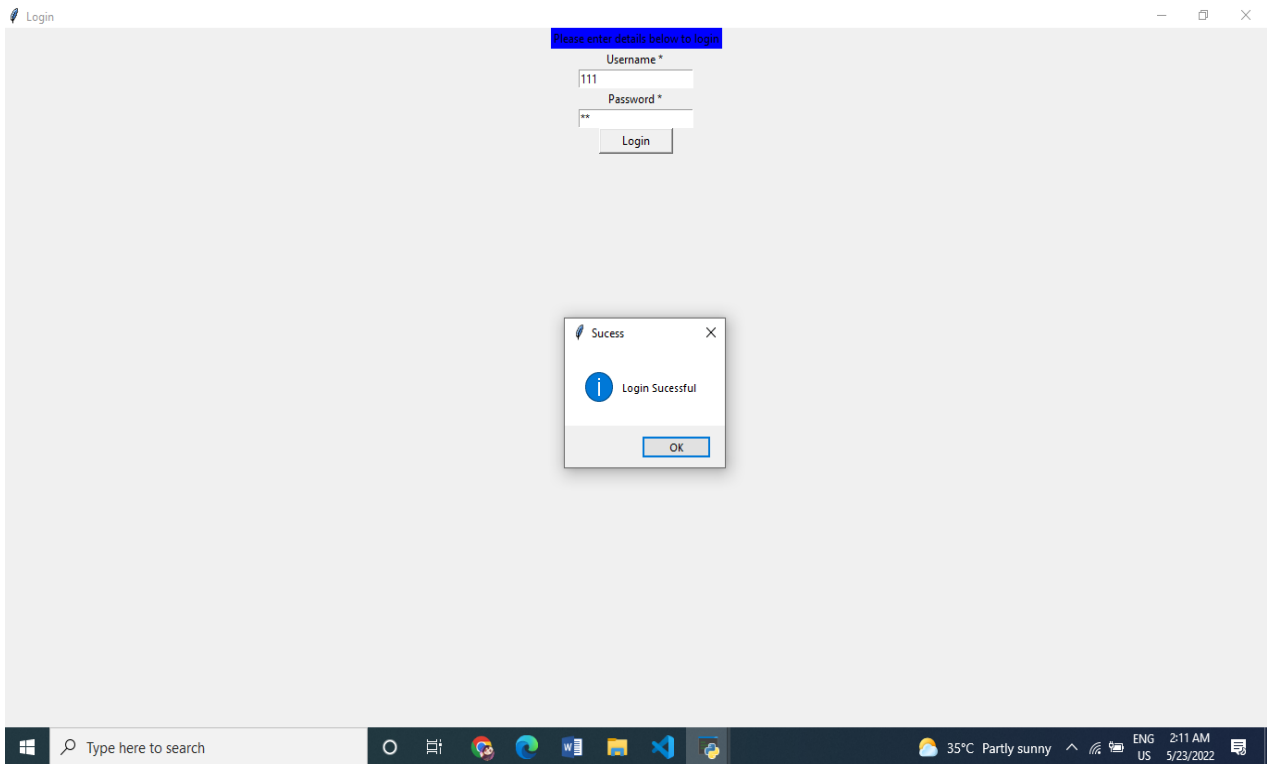


Figure 10.2: Conversation with the Healthbot

**CHAPTER 11**

**DEPLOYMENT AND  
MAINTENANCE**

## 11.1 Installation and un-installation

There some basic installations required on the developer side. The user doesn't require anything to install.

### 11.1.1 Installation of Python IDE

**Step 1:** Click the link, it will take you to the Python official download website

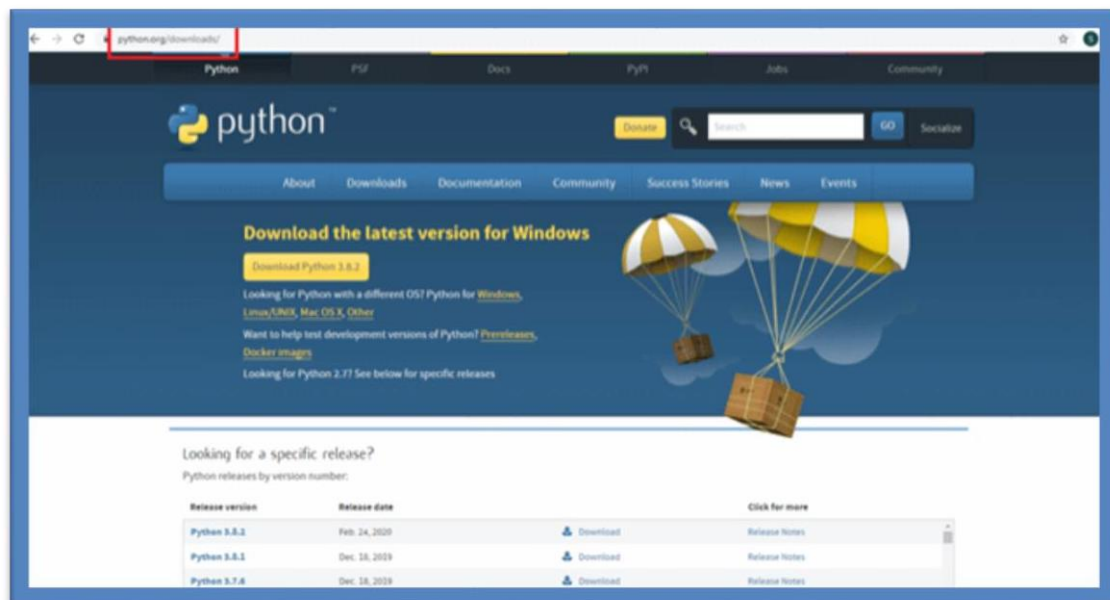


Figure 11.1: Python IDE installation setup

**Step 2:** Click the download button and you will see Python 3.8.2.

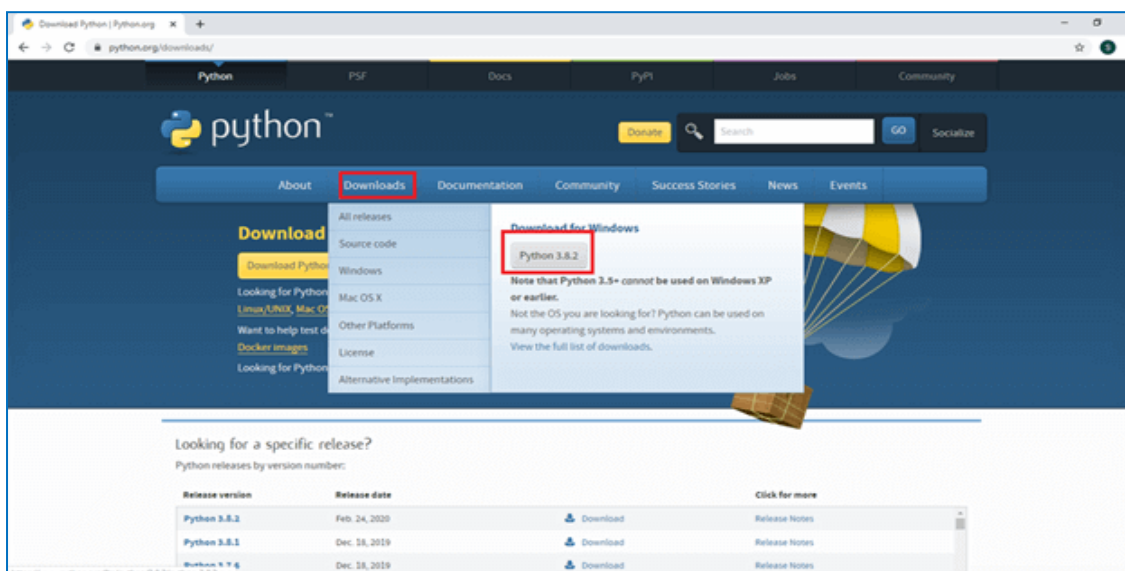


Figure 11.2: Python IDE installation setup

**Step 3:** Click Python 3.8.2 and Python will start to download.

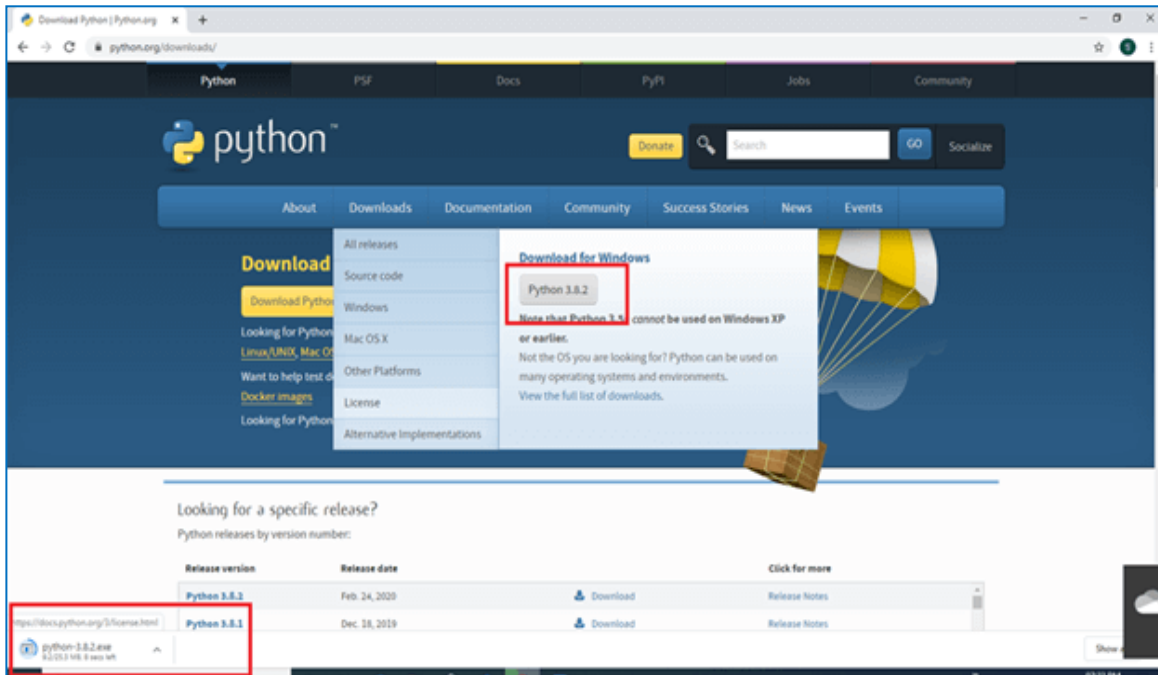


Figure 11.3: Python IDE installation setup

**Step 4:** Next, right click the mouse button you will see open button click to open.

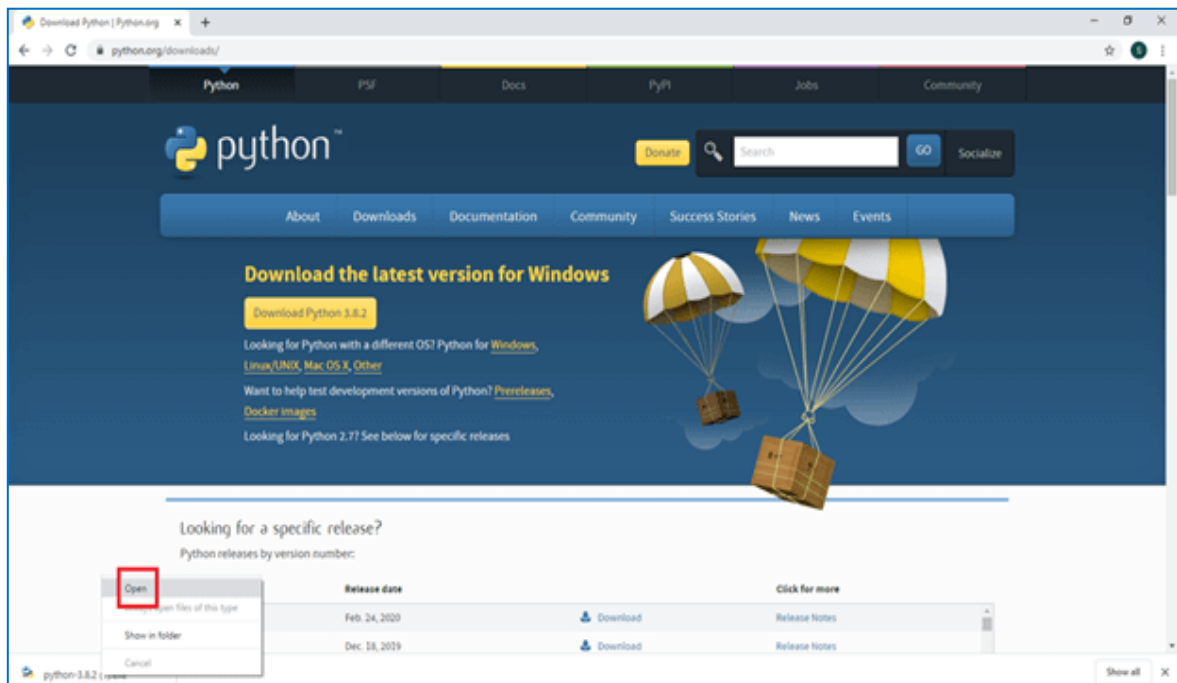




Figure 11.4: Python IDE installation setup  
**Step 5:** Enable to add Python 3.8 to path and click install now.



Figure 11.5: Python IDE installation setup

**Step 6:** Wait a few minutes and display setup was successful. Next you will click the close button.

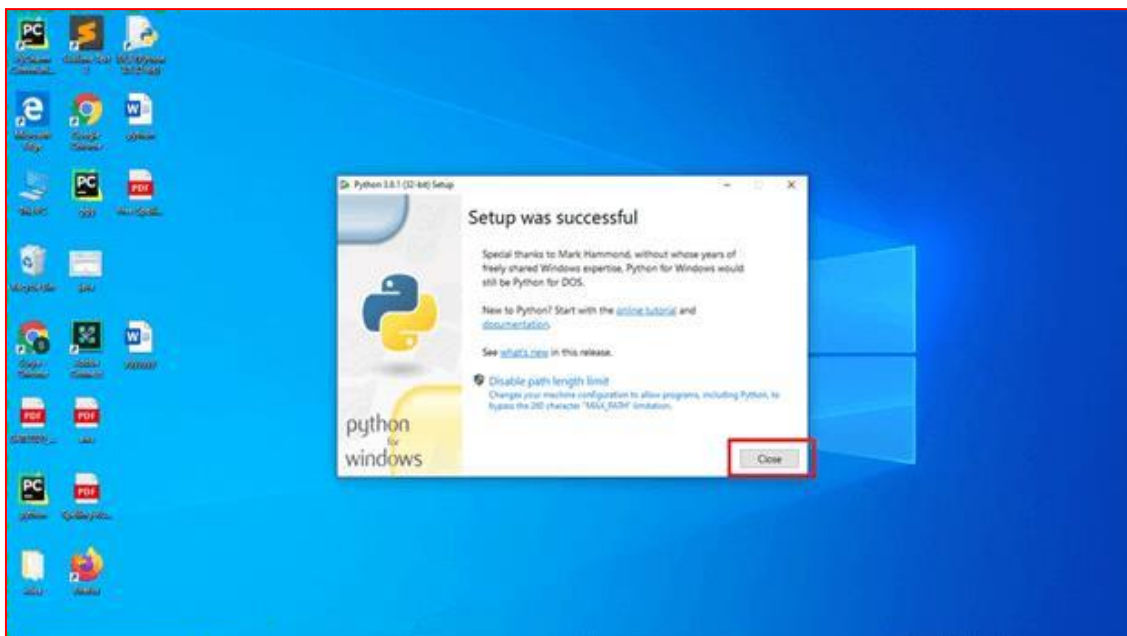


Figure 11.6: Python IDE installation setup  
RMDSSOE , Department of computer Engineering

**Step 7:** Next you will search for IDLE Python, then click enter.

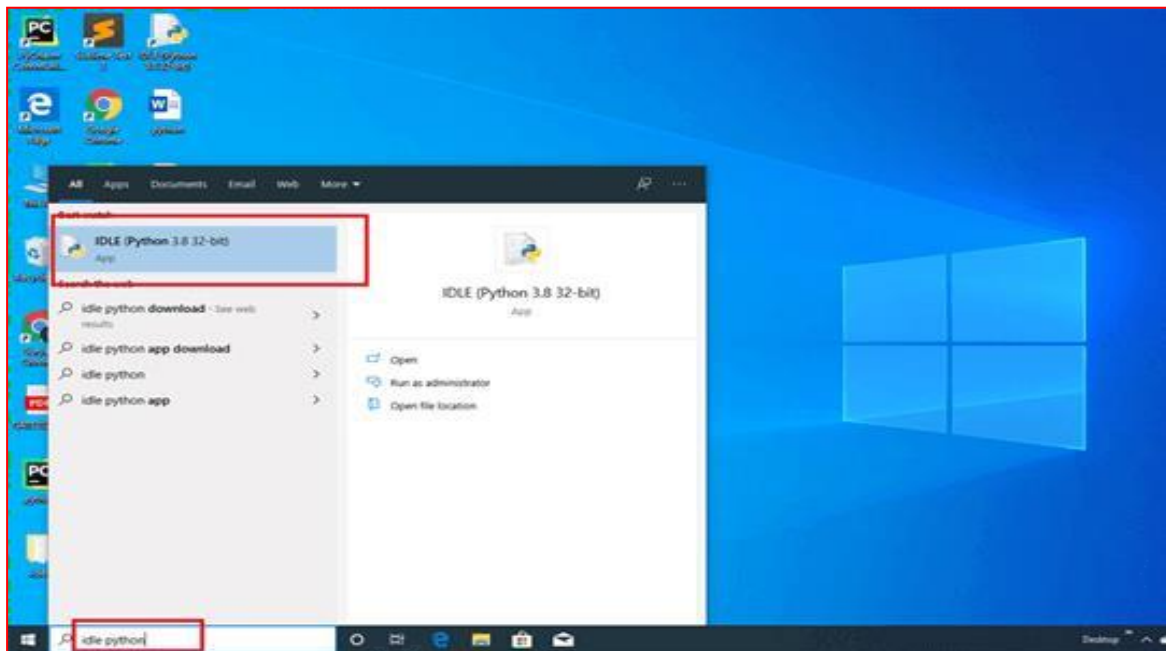


Figure 11.7: Python IDE installation setup

**Step 8:** Next, click the file button. You will see a new file again. Click the new file button.

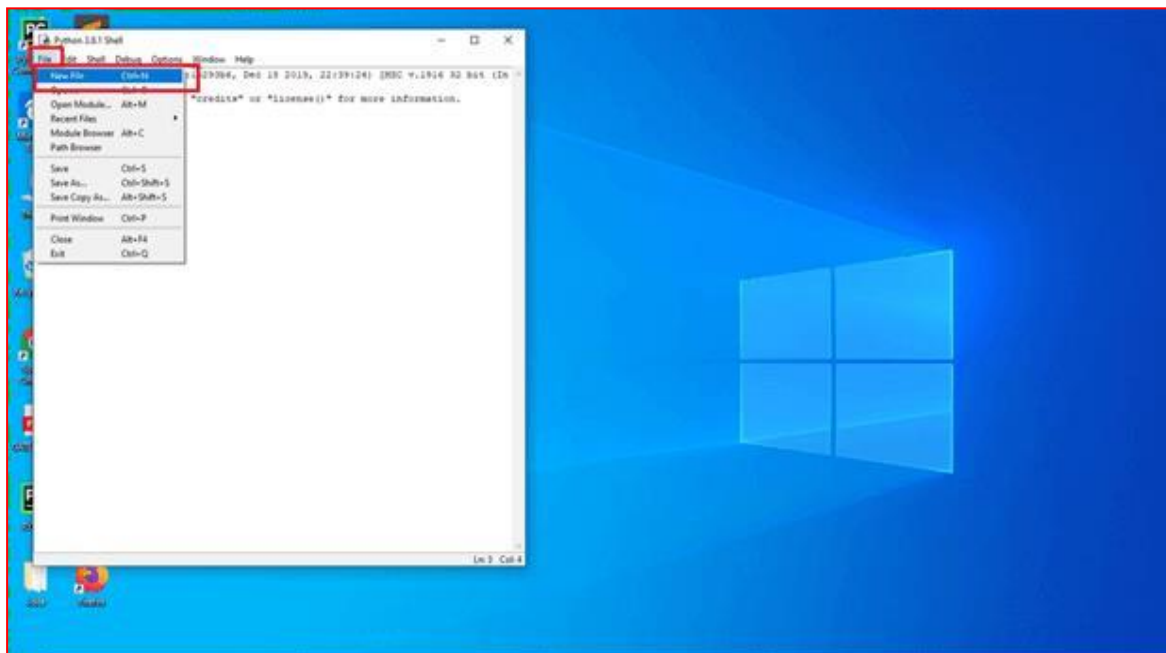
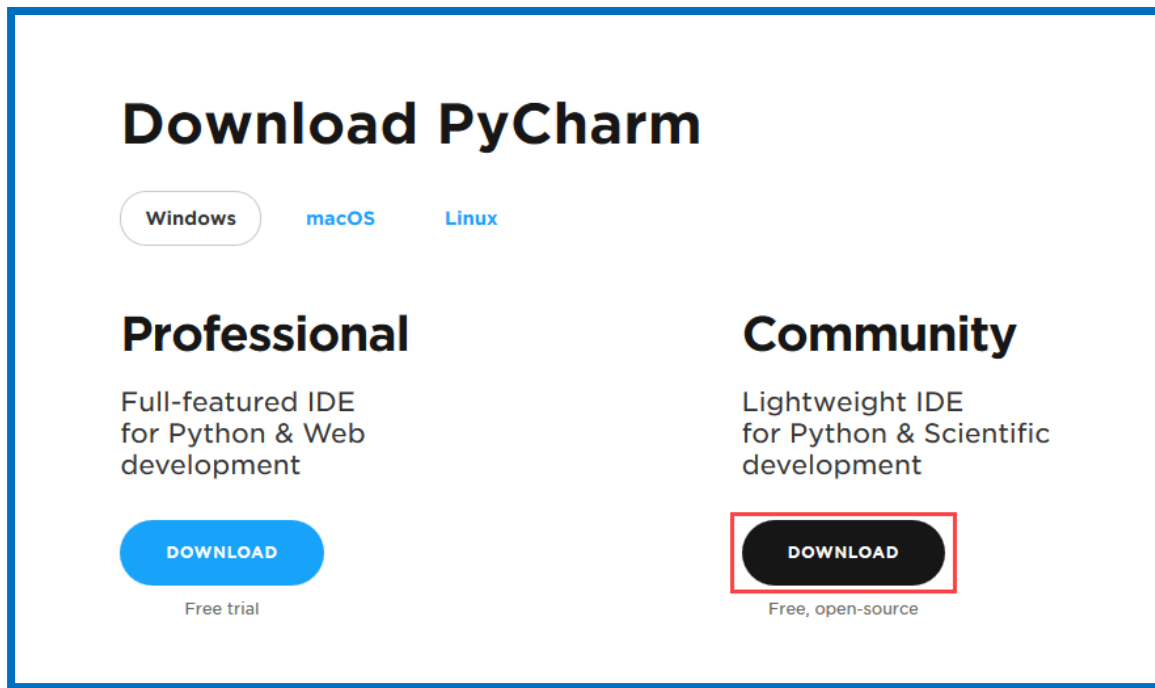


Figure 11.8: Python IDE installation Complete

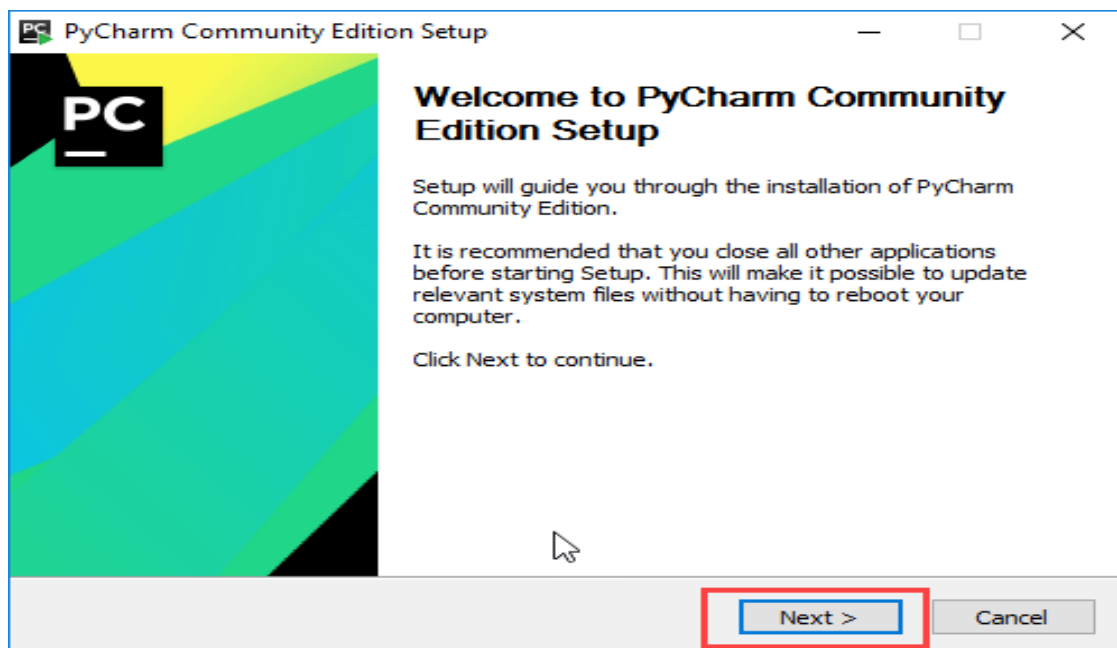
PyCharm:

**Step 1:** To download PyCharm just go the website and click the "DOWNLOAD" link under the Community Section.

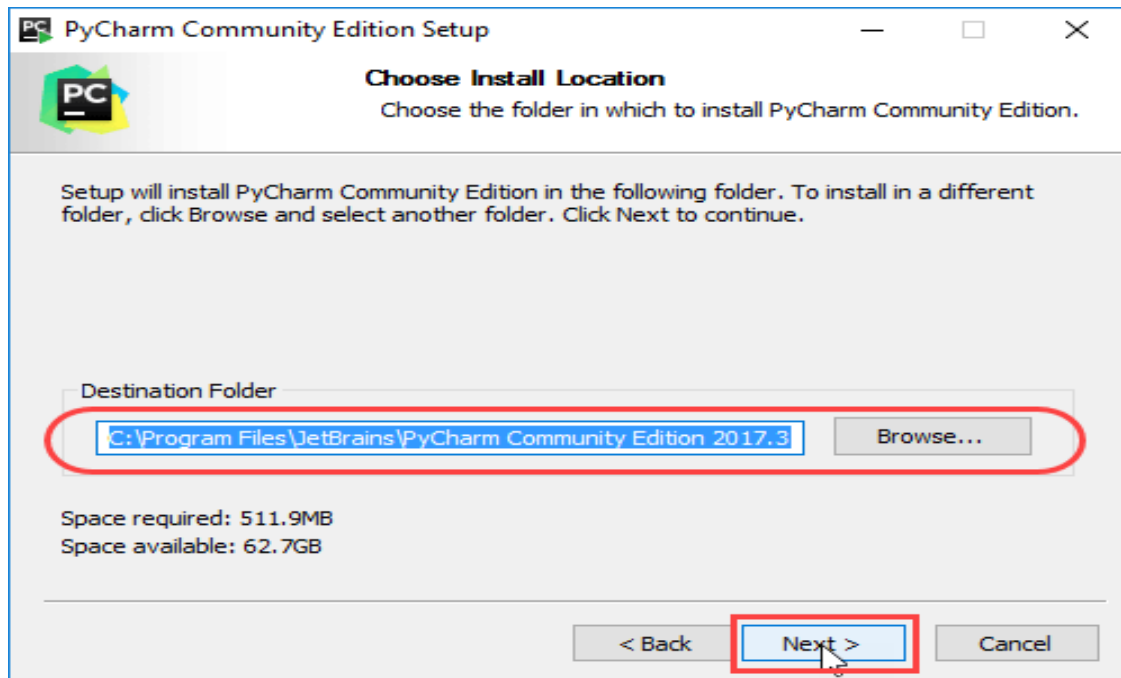
Figure 11.9: PyCharm installation setup



**Step 2:** Once the download is complete, run the exe for install PyCharm. The setup wizard should have started. Click "Next".



**Figure 11.10: PyCharm installation setup Step 3:** On the next screen, Change the installation path if required. Click “Next”.



**Figure 11.11: PyCharm installation setup**

**Step 4:** On the next screen, you can create a desktop shortcut if you want and click on “Next”.

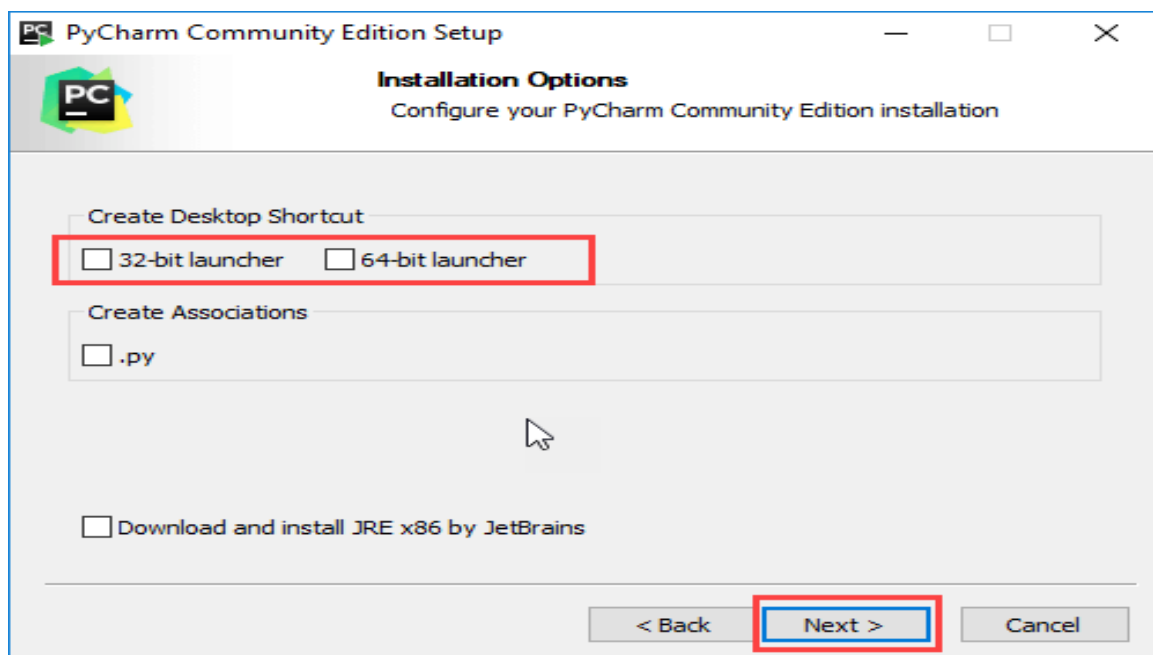


Figure 11.12: PyCharm installation setup **Step 5:** Choose the start menu folder. Keep selected JetBrains and click on “Install”.

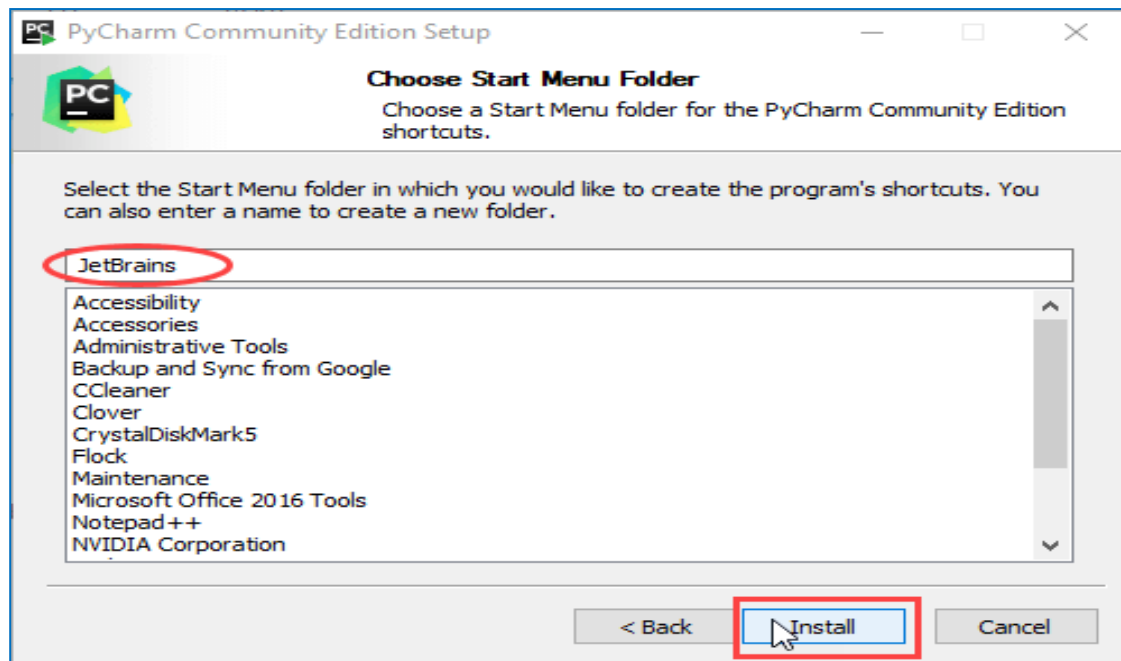


Figure 11.13: PyCharm installation setup

**Step 6:** Wait for the installation to finish.

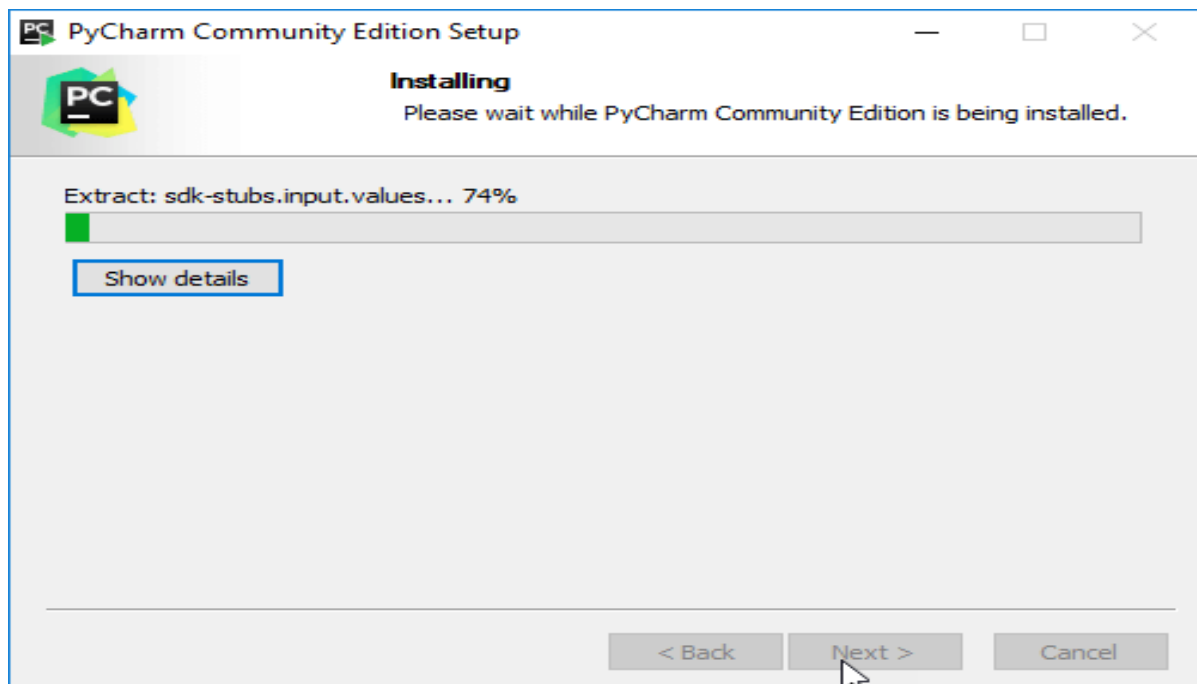


Figure 11.14: PyCharm installation Complete

**Step 7)** Once installation finished, you should receive a message screen that PyCharm is installed. If you want to go ahead and run it, click the “Run PyCharm Community Edition” box first and click “Finish”.

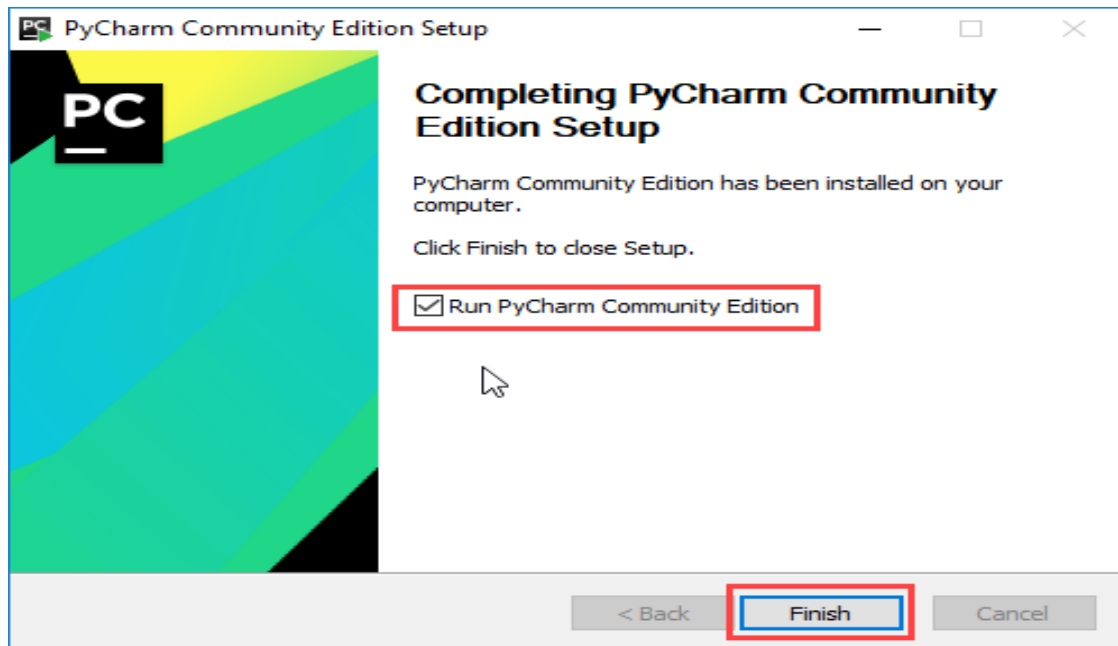


Figure 11.15: Run PyCharm Community Edition

**Step 8)** After you click on "Finish," the Following screen will appear.

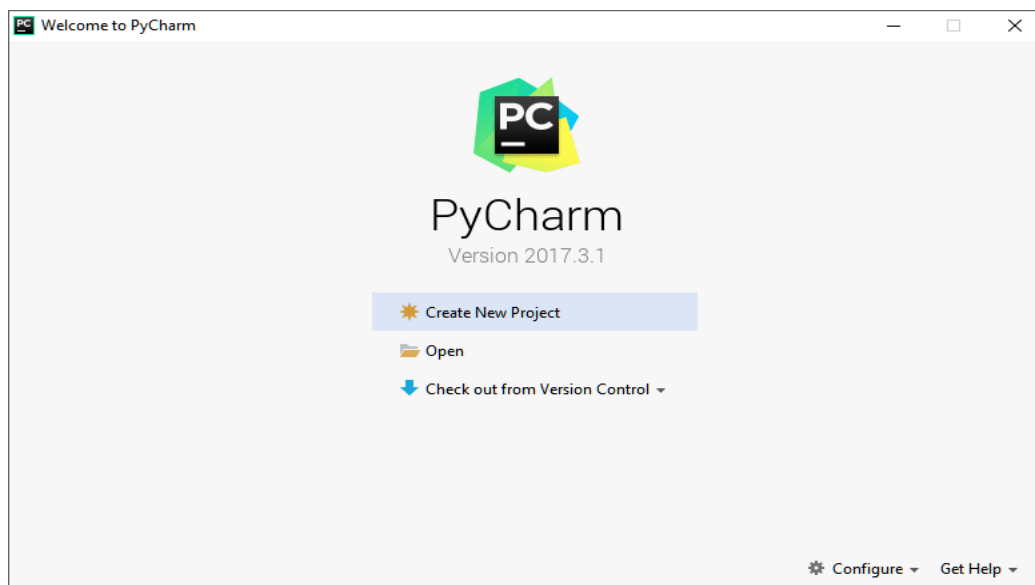
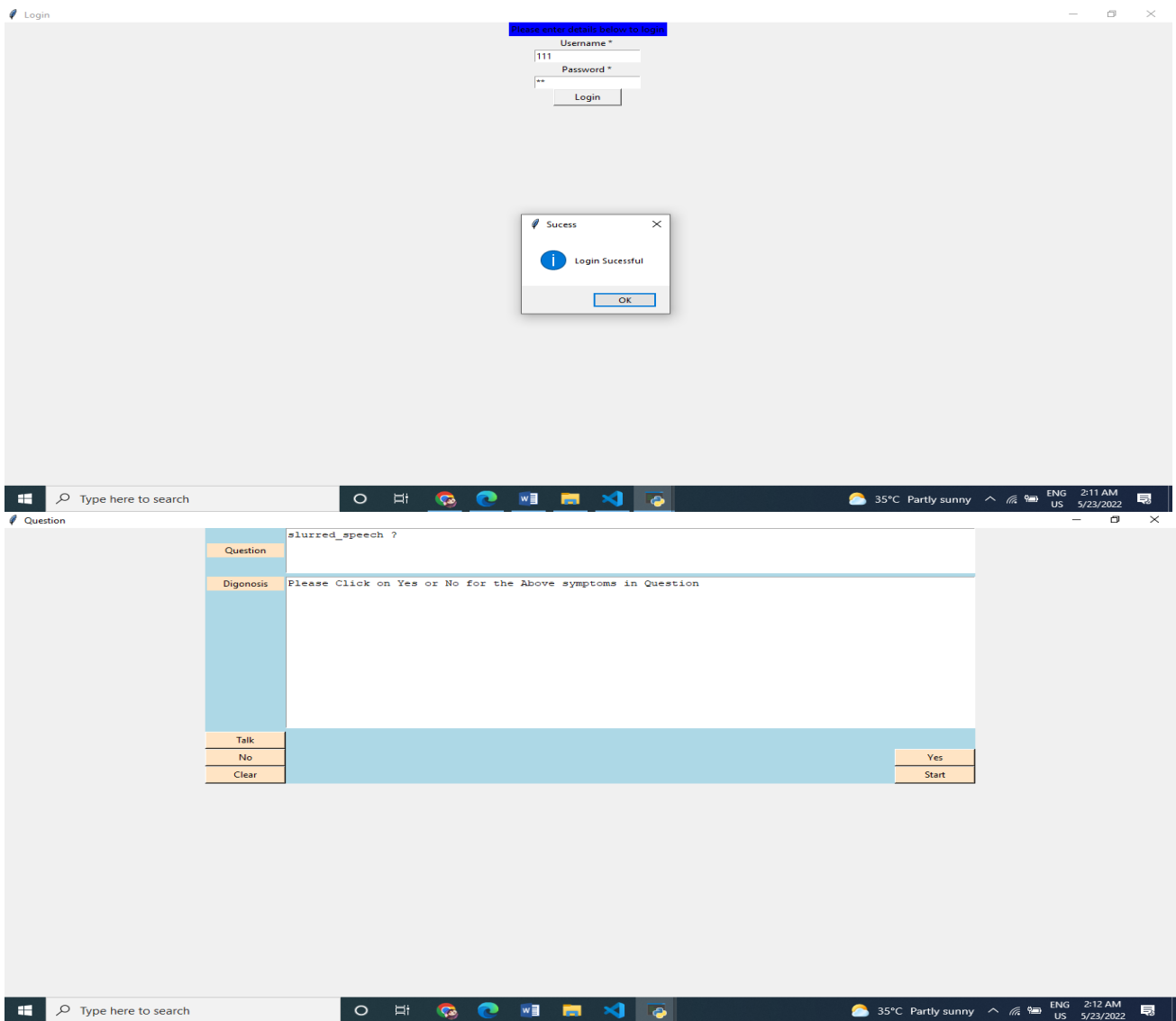


Figure 11.16: Create New PyCharm Project



## **12.1 Conclusion**

In this project, we presented an Artificially Intelligent Chatbot using applications of Deep Learning named as Curebot to fight COVID-19 situation including various viral diseases faced by human being in day-to-day life. Keeping in mind, the situation of the rural population and imbalance between the demand and healthcare services currently provided, so by developing this CureBot we will bridge a gap by creating a conversational application with Natural Language Processing. Our survey gives solution on how human can tackle various diseases implementing Deep learning in CureBot. We have shown how the Data is inputted to Deep Neural Network and how task is constructed as learning problem. We have covered some solutions to the user's query which will be beneficial for proper understanding of the patients. We also have implemented speech to text conversation for better use of Chatbot.

## **12.2 Future Scope**

Despite the initial chatbot hype dwindling down, medical chatbots still have the potential to improve the healthcare industry. The three main areas where they can be particularly useful include diagnostics, patient engagement outside medical facilities, and mental health.

- Customer Service/Administration with an AI Chatbot
- Patient Engagement
- Mental Health
- Research / Treatment
- Claims and Billing

## **12.3 Summary**

- Thus, this project accomplishes high accuracy on speech to text conversation.
- This will cover some solutions to the user's query which will be beneficial for proper understanding of the patients.
- This will provide efficient solutions for user queries.
- This Bot offers enhanced technologies benefits, reduce user hassle costs, and increased user welfare.



## **CHAPTER 13**

## **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] Katlego Mabunda and Abejide Ade-Ibijola, "PathBot: An Intelligent Chatbot for Guiding Visitors and Locating Venues", 6th International Conference on Soft Computing & Machine Intelligence (ISCMCI), Johannesburg Parktonian All-Suite, South Africa, (2019) November 19-20.

[5] Ali Bou Nassif, Ismail Shahin, Imtinan Attili and Mohammad Azze, "Speech Recognition Using Deep Neural Networks: A Systematic Review" *IEEE-Access*, Vol. 7, (2019), pp(99).1-1.

[6] Connor Shorten, Taghi M. Khoshgoftaar and Borko Furht, "Deep Learning applications for COVID-19", *SpringerOpen J Big Data* 8, Article no8, (Jan 2021).

[7] Eleni Adamopoulou and Lefteris Moussiades, "Chatbots: History, Technology, and applications", *ELSEVIER Machine Learning with Applications*, Vol. 2, (Dec 2020).

[8] <https://www.investopedia.com/terms/d/deeplearning.asp#:~:text=Deep%20learning%20is%20a%20subset,learning%20or%20deep%20neural%20network>.

[9] <https://towardsdatascience.com/how-to-build-your-own-chatbot-using-deep-learning-bb41f970e281>.

[10] <https://towardsdatascience.com/deep-learning-for-nlp-creating-a-chatbot-with-keras-da5ca051e051>.

[11] Deep Learning Definition – Investopedia.

[12] <https://towardsdatascience.com/how-to-create-a-chatbot-with-python-deep-learning-in-less-than-an-hour-56a063bdfc44>.

# **INFORMATION OF PROJECT GROUP MEMBERS**



1. **Name:** Sakshi Sajjanshetty
2. **Date of Birth:** 24/06/2000
3. **Gender:** Female
4. **Permanent Address:** The lake district,kondhwa,pune
5. **E-Mail:** [sakshishetty2406@gmail.com](mailto:sakshishetty2406@gmail.com)
6. **Mobile/Contact No.:** 9834673934
7. **Paper**  
**Published:**[https://w  
ww.ijiir.org/vol1iss1  
.html](https://www.ijiir.org/vol1iss1.html)
8. **Placement Details :NA**



1. **Name:** Dhanashree Bhimrao Dhanajkar
2. **Date of Birth:** 13/06/1997
3. **Gender:** Female
4. **Permanent Address:** 09,Dhanajkar Niwas,Near Buddha vihar  
Pawdewadi naka..Nanded..431601..
5. **E-Mail:** dhanashreedhanajkar@gmail.com
6. **Mobile/Contact No.:** 9112488441
7. **Paper Published:**  
[https://www.ijiir.org](https://www.ijiir.org/vol1iss1.html)  
[/vol1iss1.html](https://www.ijiir.org/vol1iss1.html)
8. **Placement Details :**NA



1. **Name:** Abhishek Ramlal gaikwad
2. **Date of Birth:** 14-04-2000
3. **Gender:** M ale
4. **Permanent Address:** Subhash Colony peth Beed beed
5. **E-Mail:** Mr.abhishekgaikwad@gmail.com
6. **Mobile/Contact No:** 8605192561
7. **Paper Published:**  
[https://www.ijjir.org](https://www.ijjir.org/vol1iss1.html)  
[/vol1iss1.html](https://www.ijjir.org/vol1iss1.html)
8. **Placement Details :**NA



1. **Name:** Dheeraj Sukhdev bhor
2. **Date of Birth:** 14-11-1999
3. **Gender:** male
4. **Permanent Address:** Sai-Park, Dighi, Pune-15
5. **E-Mail:** Dheerajbhor77@gmail.com
6. **Mobile/Contact No.:** 7219463623
7. **Paper Published:**  
<https://www.ijir.org/vol1iss1.html>
8. **Placement Details:** NA

