iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** V **Month of publication:** May 2022

DOI: https://doi.org/10.22214/ijraset.2022.42854

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue V May 2022- Available at www.ijraset.com

Health Care Chatbot Using NLP and Flask

Kshitij Thakre[1], Dr. P. R. Rothe[2], Sakshi Kukade[3], Pranali Shinde[4], Komal Madame[5]
[1, 2, 3, 4]*Department of Electronics and Telecommunication Engineering,Priyadarshini College of Engineering, Nagpur - 440019*
[5]*Guide*

Abstract: Health care is very important to start a good life. However, if you have a health problem, it is very difficult to talk to your doctor. Chatbots can be used to communicate with text or voice interfaces and receive responses via artificial intelligence. Chatbots are programs designed to automatically interact with incoming messages. Chatbots can be programmed to respond the same each time and respond differently to messages containing specific keywords. In addition, you can use machine learning to adapt your response to your situation. More and more hospitals, nursing homes, and even private centers are now using online human service chatbots on their websites. These bots connect with potential patients accessing the site, find specialists, make appointments, and ensure appropriate treatment. In any case, the use of artificial intelligence in industries where people's lives can be a problem still raises personal concerns. The question arises whether the above tasks should be left to human staff. This healthcare chatbot system helps hospitals provide 24/7 online healthcare support. It answers both deep and general questions. It also helps generate leads and automatically delivers lead information to sales. Asking in turn helps the patient by guiding them on exactly what they are looking for.

I. ACKNOWLEDGEMENT

Priyadarshini College of Engineering is a well established & renowned institute and follows a goal of creating technocrats and brings it into reality, which will perform challenging endeavor in technical field for welfare of human being.
We wish to avail this opportunity to express our sincere thanks to our Guide Dr.P.R.Rothe & Mr.Deoram G. Gahane , who continuously supervised our work with utmost care and zeal. They have always guided us in our endeavor to present our project on "HEALTH CARE CHATBOT USING NLP AND FLASK".
It's a great pleasure to express our deep sense of gratitude and the whole hearted thanks to our principal Dr.S.A.Dhale, Priyadarshini College of Engineering, Nagpur.
We offer special thanks to our Head of the Department Dr.V.K.Takasande, for giving us the opportunity to undertake this project.
We offer hearty gratitude to, Lab assistants of ET Department, PCE, for their valuable support during the execution of the project work. Lastly, we express our deep sense of gratitude to all teaching and non-teaching staff of the Institute and all our friends, family members, who directly and indirectly helped us to complete our project successfully and to bring it into reality.

II. INTRODUCTION

Today, our lives require great health care. Today, people are busy with work receptions, work at work, and additional addictions to the Internet. You are not involved in your health. So you avoid going to the hospital because of a small problem. This can be a significant disadvantage. Therefore, before consulting your doctor, consider developing a health chatbot system with AI that can identify the illness and provide basic information about the illness. Chatbots are software programs used to interact with customers using natural language processing in text or text-to-speech format. Chatbots were originally developed to interact with humans only for entertainment purposes. In modern times, the biggest challenge facing India as a country is to provide quality and affordable medical services to a growing population, while at the same time not being cost-effective. Difficulty in accessing and transporting medical facilities, especially in rural areas, means that patients postpone treatment or seek closer but at the same timeless cost-effective medical facilities to meet their medical needs. It means that. The role of medical chatbots, which is an efficient way to provide patients with the timely quality of care, access, and treatment, is to connect patients to chatbots and administer appropriate antibiotics/drugs and precautions. Useful for.

A. Necessity

The healthcare industry is no stranger to emergencies. And time plays a very crucial role in tackling them! Healthcare chatbots provide helpful information instantly, especially in times where every second is important. For example, if a patient rushes in with an attack, the doctor can get the patient's information like previous records, other diseases, allergies, check-ups, etc., instantly over a bot.

©IJRASET: All Rights are Reserved

B. Objectives

A chatbot is a piece of software that conducts a conversation with users via auditory or textual methods. A medical chatbot facilitates the job of a healthcare provider and helps improve their performance by interacting with users in a human-like way.
There are countless cases where intelligent medical chatbots could help physicians, nurses, therapists, patients, or their families. They can step in and minimize the amount of time they spend on tasks like:

1) Providing health-related information to users
2) Guidance for patient
3) Medication management and dosage
4) Connecting people and organizations with first responders
5) FAQ-type queries (contact details, directions, opening hours and service/treatment details)

It's important to note that despite the fact that chatbots can offer valuable facts and symptoms, they aren't qualified to give an official diagnosis. The main premise behind these talking or texting smart algorithms is to become the first point of contact before any human involvement is needed.

III. LITERATURE SURVEY

In paper[1] it reviews the present evidence for the feasibility and effectiveness of online one-on-one psychological state interventions that use the text-based synchronous chat. Synchronous written conversations (or "chats") are getting increasingly popular as Web-based psychological state interventions. This review provides an evaluation of individual synchronous Web-based chat technologies as a mode of psychological intervention and supports supported the present evidence of the appliance of this technology during this area of psychological state research, we see tentative support for this mode of intervention. Interventions utilizing text-based synchronous communication showed better outcomes compared with Waitlist conditions and overall equivalent outcomes compared with Treatment as was common and were a minimum of nearly as good because of the comparison interventions. However, the difficulty of whether these technologies are cost-effective in clinical practice remains a consideration for future research studies. In paper [2] the chatbot will act as a virtual doctor and makes it possible for the patient to interact with a virtual doctor. linguistic communication processing and pattern matching algorithm for the event of this chatbot. it's developed using the python language. supported the survey given it's found at the no of the proper answer given by the chatbot is 80% and incorrect/ambiguous answer given is 20%. This survey of chatbots and analysis of the results suggested that this software is used for teaching and as a virtual doctor for awareness and first care. [3] This literature review presents the History, Technology, and Existing Work of the Chat-Bots. Chatbots have good conversational abilities they're good at emulating a personality's conversation, and they learn through knowledge Evolution of chatbots• TURING TEST 1950 ELIZA 1960. PARRY (1972) RACTER 1984 ALEXA. GOOGLES MEENA IBM WATSON APPLE SIRI Chats-Bots mentioned during this Literature survey are additionally supported for a medicinal purpose like K-Bot

IV. SYSTEM MODELING

In our proposed system the user can chat with the bot regarding the query through text. The system usesan expert system to answer the queries. Users can also viewthe available doctors for that particular disease. This system can be used by multiple users to get counseling sessions online. The data of the chatbot is stored in the database in the form of a pattern template.

A. Block Diagram

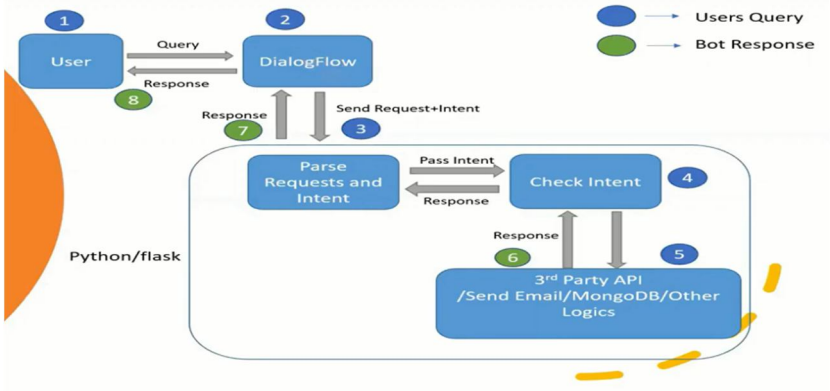


Fig.3.1: Block Diagram

B. Description of Programming Languages

1) *Python:* It is a general-purpose coding language which means that, unlike HTML, CSS, and JavaScript,it can be used for other types of programming and software development besides web development.
2) *Machine Learning:* Machine learning (ML) is the study of computer algorithms that improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.
3) *NLP (Natural Language Processing):* Natural language processing (NLP) refers to the branch of computer science—and more specifically, the branch of artificial intelligence or AI—concerned with giving computers the ability to understand the text and spoken words in much the same way human beings can.
4) *Flask:* Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries description

C. Description of Libraries use

1) *Numpy:* NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed.NumPy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.
2) *Pandas:* Pandas is an open source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays. As one of the most popular data wrangling packages, Pandas works well with many other data science modules inside the Python ecosystem, and is typically included in every Python distribution, from those that come with your operating system to commercial vendor distributions like ActiveState's ActivePython.
3) *Matplotlib:* Matplotlib is a plotting library available for the Python programming language as a component of NumPy, a big data numerical handling resource. Matplotlib uses an object oriented API to embed plots in Python applications.

V. IMPLEMENTATION

In our chatbot, you must first register with our chatbot system. Then this data will be stored in our system. You will then connect to the chatbot. First, you will see the option to start the chatbot. When you click the start button, the chatbot will ask you questions like "What kind of symptoms are you experiencing", then the bot will show you some symptoms on the display screen. You just need to answer "yes" or "no". If you say yes, these types of symptoms happen to me, the Chatbot will match the symptoms from its dataset and accordingly it will prescribe the original medication and doctor's contact details, so you can contact your doctor directly or make an appointment at the doctor. In this chatbot, we can contact the doctor directly by clicking on the website link provided on the chatbot.

A. Software Development and Implementation

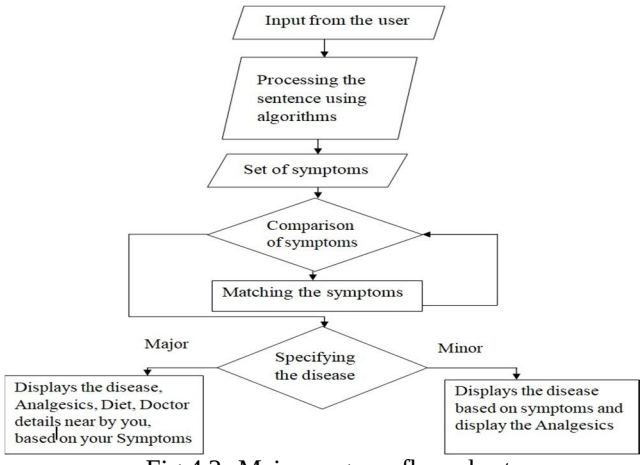


Fig.4.2: Main program flow chart

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue V May 2022- Available at www.ijraset.com

1) Main Program:

QuestionDiagonosisTkinter.

```
# Importing the libraries
from tkinter import *
from tkinter import messagebox
import os
import webbrowser
import numpy as np
import pandas as pd

class HyperlinkManager:
        def __init__(self, text):
        self.text = text
        self.text.tag_config("hyper", foreground="blue", underline=1)
        self.text.tag_bind("hyper", "<Enter>", self._enter)
        self.text.tag_bind("hyper", "<Leave>", self._leave)
        self.text.tag_bind("hyper", "<Button-1>", self._click)

        self.reset()

    def reset(self):
            self.links = {}

def add(self, action):
        # add an action to the manager.  returns tags to use in
        # associated text widget
tag = "hyper-%d" % len(self.links)
        self.links[tag] = action
        return "hyper", tag
def _enter(self, event):
        self.text.config(cursor="hand2")

    def _leave(self, event):
        self.text.config(cursor="")

    def _click(self, event):
        for tag in self.text.tag_names(CURRENT):
            if tag[:6] == "hyper-":
                self.links[tag]()
                return
# Importing the dataset
training_dataset = pd.read_csv('Training.csv')
test_dataset = pd.read_csv('Testing.csv')

# Slicing and Dicing the dataset to separate features from predictions
X = training_dataset.iloc[:, 0:132].values
Y = training_dataset.iloc[:, -1].values
# Dimensionality Reduction for removing redundancies
dimensionality_reduction = training_dataset.groupby(training_dataset['prognosis']).max()
```

©IJRASET: All Rights are Reserved

```
# Encoding String values to integer constants
from sklearn.preprocessing import LabelEncoder
labelencoder = LabelEncoder()
y = labelencoder.fit_transform(Y)

# Splitting the dataset into training set and test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)

# Implementing the Decision Tree Classifier
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier()
classifier.fit(X_train, y_train)

# Saving the information of columns
cols     = training_dataset.columns
cols     = cols[:-1]

# Checking the Important features
importances = classifier.feature_importances_
indices = np.argsort(importances)[::-1]
features = cols

  # Implementing the Visual Tree
from sklearn.tree import _tree

# Method to simulate the working of a Chatbot by extracting and formulating questions
def print_disease(node):
      #print(node)
      node = node[0]
      #print(len(node))
      val  = node.nonzero()
      #print(val)
      disease = labelencoder.inverse_transform(val[0])
      return disease
def recurse(node, depth):
         global val,ans
         global tree_,feature_name,symptoms_present
         indent = "  " * depth
         if tree_.feature[node] != _tree.TREE_UNDEFINED:
            name = feature_name[node]
            threshold = tree_.threshold[node]
            yield name + " ?"
#             ans = input()
            ans = ans.lower()
if ans == 'yes':
                val = 1
            else:
                val = 0
            if  val <= threshold:
```

```
                yield from recurse(tree_.children_left[node], depth + 1)
            else:
                symptoms_present.append(name)
                yield from recurse(tree_.children_right[node], depth + 1)
        else:


strData=""
            present_disease = print_disease(tree_.value[node])
#             print( "You may have " +  present_disease )
#             print()
            strData="You may have :" +  str(present_disease)

            QuestionDigonosis.objRef.txtDigonosis.insert(END,str(strData)+'\n')

            red_cols = dimensionality_reduction.columns
            symptoms_given = red_cols[dimensionality_reduction.loc[present_disease].values[0].nonzero()]
#             print("symptoms present  " + str(list(symptoms_present)))
#             print()


strData="symptoms present:  " + str(list(symptoms_present))
            QuestionDigonosis.objRef.txtDigonosis.insert(END,str(strData)+'\n')
#             print("symptoms given "  +  str(list(symptoms_given)) )
#             print()
            strData="symptoms given: "  +  str(list(symptoms_given))
            QuestionDigonosis.objRef.txtDigonosis.insert(END,str(strData)+'\n')
            confidence_level = (1.0*len(symptoms_present))/len(symptoms_given)
#             print("confidence level is " + str(confidence_level))
#             print()
            strData="confidence level is: " + str(confidence_level)
            QuestionDigonosis.objRef.txtDigonosis.insert(END,str(strData)+'\n')
#             print('The model suggests:')
#             print()

strData='The model suggests:'
            QuestionDigonosis.objRef.txtDigonosis.insert(END,str(strData)+'\n')
            row = doctors[doctors['disease'] == present_disease[0]]
#             print('Consult ', str(row['name'].values))
#             print()
            strData='Consult '+ str(row['name'].values)
            QuestionDigonosis.objRef.txtDigonosis.insert(END,str(strData)+'\n')
#             print('Visit ', str(row['link'].values))
#print(present_disease[0])
            hyperlink = HyperlinkManager(QuestionDigonosis.objRef.txtDigonosis)
            strData='Visit '+ str(row['link'].values[0])
            def click1():
                webbrowser.open_new(str(row['link'].values[0]))
            QuestionDigonosis.objRef.txtDigonosis.insert(INSERT, strData, hyperlink.add(click1))
            #QuestionDigonosis.objRef.txtDigonosis.insert(END,str(strData)+'\n')
```

```
            yield strData
def tree_to_code(tree, feature_names):
        global tree_,feature_name,symptoms_present
        tree_ = tree.tree_
        #print(tree_)
        feature_name = [
            feature_names[i] if i != _tree.TREE_UNDEFINED else "undefined!"
            for i in tree_.feature
        ]
        #print("def tree({}):".format(", ".join(feature_names)))
        symptoms_present = []
#       recurse(0, 1)
def execute_bot():
#    print("Please reply with yes/Yes or no/No for the following symptoms")


  tree_to_code(classifier,cols)

# This section of code to be run after scraping the data
doc_dataset = pd.read_csv('doctors_dataset.csv', names = ['Name', 'Description'])


diseases = dimensionality_reduction.index
diseases = pd.DataFrame(diseases)

doctors = pd.DataFrame()
doctors['name'] = np.nan
doctors['link'] = np.nan
doctors['disease'] = np.nan

doctors['disease'] = diseases['prognosis']


doctors['name'] = doc_dataset['Name']
doctors['link'] = doc_dataset['Description']

record = doctors[doctors['disease'] == 'AIDS']
record['name']
record['link']

# Execute the bot and see it in Action
#execute_bot()
class QuestionDigonosis(Frame):
    objIter=None
    objRef=None
    def __init__(self,master=None):
        master.title("Question")
        # root.iconbitmap("")
        master.state("z")
#       master.minsize(700,350)
```

```
        QuestionDigonosis.objRef=self
        super().__init__(master=master)
        self["bg"]="light blue"
        self.createWidget()
        self.iterObj=None
def createWidget(self):
        self.lblQuestion=Label(self,text="Question",width=12,bg="bisque")
        self.lblQuestion.grid(row=0,column=0,rowspan=4)


self.lblDigonosis = Label(self, text="Digonosis",width=12,bg="bisque")
        self.lblDigonosis.grid(row=4, column=0,sticky="n",pady=5)

        # self.varQuestion=StringVar()
        self.txtQuestion = Text(self, width=100,height=4)
        self.txtQuestion.grid(row=0, column=1,rowspan=4,columnspan=20)

        self.varDiagonosis=StringVar()
        self.txtDigonosis =Text(self, width=100,height=14)
        self.txtDigonosis.grid(row=4, column=1,columnspan=20,rowspan=20,pady=5)

        self.btnNo=Button(self,text="No",width=12,bg="bisque", command=self.btnNo_Click)
        self.btnNo.grid(row=25,column=0)
        self.btnYes = Button(self, text="Yes",width=12,bg="bisque", command=self.btnYes_Click)
        self.btnYes.grid(row=25, column=1,columnspan=20,sticky="e")

        self.btnClear = Button(self, text="Clear",width=12,bg="bisque", command=self.btnClear_Click)
        self.btnClear.grid(row=27, column=0)
self.btnStart = Button(self, text="Start",width=12,bg="bisque", command=self.btnStart_Click)
        self.btnStart.grid(row=27, column=1,columnspan=20,sticky="e")
    def btnNo_Click(self):
        global val,ans
        global val,ans
        ans='no'
        str1=QuestionDigonosis.objIter.__next__()
        self.txtQuestion.delete(0.0,END)
        self.txtQuestion.insert(END,str1+"\n")
        def btnYes_Click(self):
        global val,ans
        ans='yes'
        self.txtDigonosis.delete(0.0,END)
        str1=QuestionDigonosis.objIter.__next__()
#         self.txtDigonosis.insert(END,str1+"\n")
    def btnClear_Click(self):
        self.txtDigonosis.delete(0.0,END)

self.txtQuestion.delete(0.0,END)
    def btnStart_Click(self):
        execute_bot()
        self.txtDigonosis.delete(0.0,END)
```

```
        self.txtQuestion.delete(0.0,END)
        self.txtDigonosis.insert(END,"Please Click on Yes or No for the Above symptoms in Question")
        QuestionDigonosis.objIter=recurse(0, 1)
        str1=QuestionDigonosis.objIter.__next__()
        self.txtQuestion.insert(END,str1+"\n")

class MainForm(Frame):
    main_Root = None
    def destroyPackWidget(self, parent):
        for e in parent.pack_slaves():
            e.destroy()
    def __init__(self, master=None):
        MainForm.main_Root = master
        super().__init__(master=master)
        master.geometry("300x250")
        master.title("Account Login")
        self.createWidget()

def createWidget(self):
        self.lblMsg=Label(self, text="Health Care Chatbot", bg="PeachPuff2", width="300", height="2", font=("Calibri", 13))
        self.lblMsg.pack()
        self.btnLogin=Button(self, text="Login", height="2", width="300", command = self.lblLogin_Click)
        self.btnLogin.pack()
        self.btnRegister=Button(self, text="Register", height="2", width="300", command = self.btnRegister_Click)
        self.btnRegister.pack()
        self.lblTeam=Label(self, text="Made by:", bg="slateblue4", width = "250", height = "1", font=("Calibri", 13))
        self.lblTeam.pack()
        self.lblTeam1=Label(self, text="Kshitij Thakre", bg="RoyalBlue1", width = "250", height = "1", font=("Calibri", 13))
        self.lblTeam1.pack()
        self.lblTeam2=Label(self, text="Sakshi Kukde", bg="RoyalBlue2", width = "250", height = "1", font=("Calibri", 13))
        self.lblTeam2.pack()
        self.lblTeam3=Label(self, text="Pranali Shinde", bg="RoyalBlue3", width = "250", height = "1", font=("Calibri", 13))
        self.lblTeam3.pack()
        self.lblTeam4=Label(self, text="Komal Madame", bg="RoyalBlue4", width = "250", height = "1", font=("Calibri", 13))
        self.lblTeam4.pack()

    def lblLogin_Click(self):
        self.destroyPackWidget(MainForm.main_Root)
        frmLogin=Login(MainForm.main_Root)
        frmLogin.pack()
    def btnRegister_Click(self):
        self.destroyPackWidget(MainForm.main_Root)
        frmSignUp = SignUp(MainForm.main_Root)
        frmSignUp.pack()

class Login(Frame):
    main_Root=None
    def destroyPackWidget(self,parent):
        for e in parent.pack_slaves():
            e.destroy()
```

```
def __init__(self, master=None):
        Login.main_Root=master
        super().__init__(master=master)
        master.title("Login")

master.geometry("300x250")
        self.createWidget()
    def createWidget(self):
        self.lblMsg=Label(self, text="Please enter details below to login",bg="blue")
        self.lblMsg.pack()
        self.username=Label(self, text="Username * ")
        self.username.pack()
        self.username_verify = StringVar()
        self.username_login_entry = Entry(self, textvariable=self.username_verify)
        self.username_login_entry.pack()
        self.password=Label(self, text="Password * ")
        self.password.pack()
        self.password_verify = StringVar()
        self.password_login_entry = Entry(self, textvariable=self.password_verify, show='*')
        self.password_login_entry.pack()
        self.btnLogin=Button(self, text="Login", width=10, height=1, command=self.btnLogin_Click)
        self.btnLogin.pack()
 def btnLogin_Click(self):
        username1 = self.username_login_entry.get()
        password1 = self.password_login_entry.get()

#       messagebox.showinfo("Failure", self.username1+":"+password1)
        list_of_files = os.listdir()
        if username1 in list_of_files:
            file1 = open(username1, "r")
            verify = file1.read().splitlines()
            if password1 in verify:
                messagebox.showinfo("Sucess","Login Sucessful")
                self.destroyPackWidget(Login.main_Root)
                frmQuestion = QuestionDigonosis(Login.main_Root)
                frmQuestion.pack()
            else:

messagebox.showinfo("Failure", "Login Details are wrong try again")
        else:
            messagebox.showinfo("Failure", "User not found try from another user\n or sign up for new user")

class SignUp(Frame):
    main_Root=None
    print("SignUp Class")
    def destroyPackWidget(self,parent):
        for e in parent.pack_slaves():
            e.destroy()
    def __init__(self, master=None):
        SignUp.main_Root=master
```

```
        master.title("Register")
        super().__init__(master=master)
        master.title("Register")
        master.geometry("300x250")
        self.createWidget()
    def createWidget(self):
        self.lblMsg=Label(self, text="Please enter details below", bg="blue")
        self.lblMsg.pack()
        self.username_lable = Label(self, text="Username * ")
        self.username_lable.pack()
        self.username = StringVar()
        self.username_entry = Entry(self, textvariable=self.username)
        self.username_entry.pack()

self.password_lable = Label(self, text="Password * ")
        self.password_lable.pack()
        self.password = StringVar()
        self.password_entry = Entry(self, textvariable=self.password, show='*')
        self.password_entry.pack()
        self.btnRegister=Button(self, text="Register", width=10, height=1, bg="blue", command=self.register_user)
        self.btnRegister.pack()


    def register_user(self):
        file = open(self.username_entry.get(), "w")
        file.write(self.username_entry.get() + "\n")
        file.write(self.password_entry.get())
        file.close()
  self.destroyPackWidget(SignUp.main_Root)
        self.lblSucess=Label(root, text="Registration Success", fg="green", font=("calibri", 11))

        self.lblSucess.pack()

self.btnSucess=Button(root, text="Click Here to proceed", command=self.btnSucess_Click)
        self.btnSucess.pack()
    def btnSucess_Click(self):


        self.destroyPackWidget(SignUp.main_Root)
        frmQuestion = QuestionDigonosis(SignUp.main_Root)

        frmQuestion.pack()

root = Tk()

frmMainForm=MainForm(root)
frmMainForm.pack()
root.mainloop()
```

2) Subprogram

Healthcare_chatbotconsole.py

```
######## A Healthcare Domain Chatbot to simulate the predictions of a General Physician ########
######## A pragmatic Approach for Diagnosis ############

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
training_dataset = pd.read_csv('Training.csv')
test_dataset = pd.read_csv('Testing.csv')

# Slicing and Dicing the dataset to separate features from predictions
X = training_dataset.iloc[:, 0:132].values
#print(X)
y = training_dataset.iloc[:, -1].values
#print(y)

# Dimensionality Reduction for removing redundancies
dimensionality_reduction = training_dataset.groupby(training_dataset['prognosis']).max()
#print(dimensionality_reduction)

# Encoding String values to integer constants
from sklearn.preprocessing import LabelEncoder
labelencoder = LabelEncoder()
y = labelencoder.fit_transform(y)
#print(y)

# Splitting the dataset into training set and test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)

# Implementing the Decision Tree Classifier
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier()
classifier.fit(X_train, y_train)

# Saving the information of columns
cols     = training_dataset.columns
cols     = cols[:-1]
# Checking the Important features
importances = classifier.feature_importances_
indices = np.argsort(importances)[::-1]
features = cols

# Implementing the Visual Tree
from sklearn.tree import _tree
```

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue V May 2022- Available at www.ijraset.com

```
# Method to simulate the working of a Chatbot by extracting and formulating questions
def execute_bot():

    print("Please reply with yes/Yes or no/No for the following symptoms")
    def print_disease(node):
        #print(node)
        node = node[0]
        #print(len(node))
        val  = node.nonzero()
        #print(val)
        disease = labelencoder.inverse_transform(val[0])

            return disease
    def tree_to_code(tree, feature_names):
        tree_ = tree.tree_
        #print(tree_)
        feature_name = [
            feature_names[i] if i != _tree.TREE_UNDEFINED else "undefined!"
            for i in tree_.feature
        ]
        #print("def tree({}):".format(", ".join(feature_names)))
        symptoms_present = []
        def recurse(node, depth):

indent = "  " * depth
            if tree_.feature[node] != _tree.TREE_UNDEFINED:
                name = feature_name[node]
                threshold = tree_.threshold[node]
                print(name + " ?")
                ans = input()
                ans = ans.lower()
               if ans == 'yes':
                    val = 1
                else:
                    val = 0
                if  val <= threshold:
                    recurse(tree_.children_left[node], depth + 1)
                else:
                    symptoms_present.append(name)
                    recurse(tree_.children_right[node], depth + 1)
else:
                present_disease = print_disease(tree_.value[node])
                print( "You may have " +  present_disease )
                print()
                red_cols = dimensionality_reduction.columns
                symptoms_given = red_cols[dimensionality_reduction.loc[present_disease].values[0].nonzero()]
print("symptoms present  " + str(list(symptoms_present)))
                print()
                print("symptoms given "  +  str(list(symptoms_given)) )
```

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

```
                print()
            confidence_level = (1.0*len(symptoms_present))/len(symptoms_given)
            print("confidence level is " + str(confidence_level))
            print()
            print('The model suggests:')
            print()
            row = doctors[doctors['disease'] == present_disease[0]]
            print('Consult ', str(row['name'].values))
            print()
            print('Visit ', str(row['link'].values))
            #print(present_disease[0])
        recurse(0, 1)

    tree_to_code(classifier,cols)
# This section of code to be run after scraping the data

doc_dataset = pd.read_csv('doctors_dataset.csv', names = ['Name', 'Description'])

        diseases = dimensionality_reduction.index
        diseases = pd.DataFrame(diseases)

        doctors = pd.DataFrame()
        doctors['name'] = np.nan
        doctors['link'] = np.nan
        doctors['disease'] = np.nan

        doctors['disease'] = diseases['prognosis']

        doctors['name'] = doc_dataset['Name']
        doctors['link'] = doc_dataset['Description']

        record = doctors[doctors['disease'] == 'AIDS']
        record['name']
        record['link']

        # Execute the bot and see it in Action
        execute_bot()
```

Doctor.cvs Files

Dr. Amarpreet Singh Riar,https://www.practo.com/delhi/doctor/amarpreet-singh-riar-general-physician?specialization=General%20Physician&practice_id=1026302

Dr. (Maj.)Sharad Shrivastava,https://www.practo.com/delhi/doctor/dr-54-general-physician-1?specialization=General%20Physician&practice_id=1071396

Dr. Anirban Biswas,https://www.practo.com/delhi/doctor/anirban-biswas-diabetologist?specialization=General%20Physician&practice_id=789800

Dr. Aman Vij,https://www.practo.com/delhi/doctor/dr-aman-vij-general-physician?specialization=General%20Physician&practice_id=704972

Dr. Mansi Arya,https://www.practo.com/delhi/doctor/dr-mansi-arya-bhardwaj-homeopath?specialization=Homoeopath&practice_id=786334

Dr. Sunil Kumar Dwivedi,https://www.practo.com/delhi/doctor/dr-sunil-kumar-dwivedi-

homeopath?specialization=Homoeopath&practice_id=654411
Dr. Chhavi Bansal,https://www.practo.com/delhi/doctor/dr-chhavi-bansal-homeopath-1?specialization=Homoeopath&practice_id=1124019
Dr. Sneh Khera,https://www.practo.com/delhi/doctor/dr-sneh-khera-homeopath?specialization=Homoeopath&practice_id=709153
Dr. Inderjeet Singh,https://www.practo.com/delhi/doctor/inderjeet-singh-ayurveda-sexologist?specialization=Homoeopath&practice_id=1219975
Dr. Suman Mohan,https://www.practo.com/delhi/doctor/dr-suman-mohan-homoeopath?specialization=Homoeopath&practice_id=1173704
Dr. Manish Munjal,https://www.practo.com/delhi/doctor/dr-manish-munjal-ear-nose-throat-ent-specialist-1?specialization=Ear-Nose-Throat%20(ENT)%20Specialist&practice_id=1045243
Dr. Ajay Jain,https://www.practo.com/delhi/doctor/dr-ajay-jain-ear-nose-throat-ent-specialist-1?specialization=Ear-Nose-Throat%20(ENT)%20Specialist&practice_id=664069
Dr. Anshul Gupta,https://www.practo.com/delhi/doctor/dr-anshul-gupta-ear-nose-throat-ent-specialist-1?specialization=Ear-Nose-Throat%20(ENT)%20Specialist&practice_id=712546
Dr. B B Khatri,https://www.practo.com/delhi/doctor/dr-b-b-khatri-ear-nose-throat-ent-specialist?specialization=Ear-Nose-Throat%20(ENT)%20Specialist&practice_id=702104
Dr. Rajeev Adhana,https://www.practo.com/delhi/clinic/adhana-ent-clinic-dilshad-garden?subscription_id=1296734&reach_subscription_id=45459&specialization=Ear-Nose-Throat%20(ENT)%20Specialist&ad_id=403277995611153&show_all=true
Dr. Vidit Tripathi,https://www.practo.com/delhi/doctor/dr-vidit-tripathi-ear-nose-throat-ent-specialist-3?specialization=Ear-Nose-Throat%20(ENT)%20Specialist&practice_id=736234
Dr. Arun Wadhawan,https://www.practo.com/delhi/doctor/dr-arun-wadhawan-ear-nose-throat-ent-specialist-1?specialization=Ear-Nose-Throat%20(ENT)%20Specialist&practice_id=710928
Dr. Neha Sood,https://www.practo.com/delhi/doctor/dr-neha-sood-ear-nose-throat-ent-specialist-1?specialization=Ear-Nose-Throat%20(ENT)%20Specialist&practice_id=680128
Dr. Vineet Narula,https://www.practo.com/delhi/doctor/vineet-narula-ear-nose-throat-ent-specialist-1?specialization=Ear-Nose-Throat%20(ENT)%20Specialist&practice_id=1107540
Dr. Yogesh Jain,https://www.practo.com/delhi/doctor/dr-yogesh-jain-2-ear-nose-throat-ent-specialist?specialization=Ear-Nose-Throat%20(ENT)%20Specialist&practice_id=1118463
Dr. Rakesh Singh,https://www.practo.com/delhi/doctor/dr-rakesh-singh-ear-nose-throat-ent-specialist?specialization=Ear-Nose-Throat%20(ENT)%20Specialist&practice_id=670997
Dr. Sudha Asokan,https://www.practo.com/delhi/doctor/dr-sudha-asokan-ayurveda-1?specialization=Ayurveda&practice_id=654473
Dr. Mahesh Shah,https://www.practo.com/delhi/doctor/dr-shah-4-sexologist?specialization=Ayurveda&practice_id=688515
Dr. S K Singh,https://www.practo.com/delhi/doctor/dr-s-k-singh-ayurveda?specialization=Ayurveda&practice_id=725617
Dr. Sudhir Bhola,https://www.practo.com/delhi/doctor/sudhir-bhola-alternative-medicine?specialization=Ayurveda&practice_id=1065993
Dr. Jyoti Arora Monga,https://www.practo.com/delhi/doctor/dr-jyoti-arora-ayurveda?specialization=Ayurveda&practice_id=693424
Dr. Vijay Abbot,https://www.practo.com/delhi/doctor/dr-vijay-abbot-sexologist?specialization=Ayurveda&practice_id=688515
Dr. Rakesh Gupta,https://www.practo.com/delhi/doctor/dr-rakesh-gupta-ayurveda?specialization=Ayurveda&practice_id=825935
Dr. Sugeeta Mutreja,https://www.practo.com/delhi/doctor/sugeeta-mutreja-dietitian-nutritionist?specialization=Ayurveda&practice_id=1010860
Dr. Ruchi Gupta,https://www.practo.com/delhi/doctor/dr-ruchi-gupta-1-ayurveda?specialization=Ayurveda&practice_id=825935
Dr. Praveen Rustagi,https://www.practo.com/delhi/doctor/dr-praveen-rustagi-ayurveda?specialization=Ayurveda&practice_id=1121150
Dr. S.K Kashyap,https://www.practo.com/delhi/doctor/s-k-kashyap-dermatologist-cosmetologist?specialization=Dermatologist&practice_id=767677
Dr. Nipun Jain,https://www.practo.com/delhi/doctor/dr-nipun-jain-dermatologist-cosmetologist-2?specialization=Dermatologist&practice_id=850375
Dr. Rohit Batra,https://www.practo.com/delhi/doctor/dr-rohit-batra-dermatologist-cosmetologist-

1?specialization=Dermatologist&practice_id=673325
Dr. Lipy Gupta,https://www.practo.com/delhi/doctor/dr-lipy-gupta-dermatologist-cosmetologist?specialization=Dermatologist&practice_id=689248
Dr. Gaurav Garg,https://www.practo.com/delhi/doctor/dr-gaurav-garg-dermatologist-cosmetologist?specialization=Dermatologist&practice_id=690976
Dr. Parmil Kumar Sharma,https://www.practo.com/delhi/doctor/dr-p-k-sharma-dermatologist-cosmetologist-1?specialization=Dermatologist&practice_id=896686
Dr. Shruti Gupta,https://www.practo.com/delhi/doctor/shruti-gupta-dermatologist?specialization=Dermatologist&practice_id=1009300
Dr. Manisha Chopra,https://www.practo.com/delhi/doctor/dr-manisha-chopra-dermatologist-cosmetologist?specialization=Dermatologist&practice_id=700888
Dr. Ranjan Upadhyay,https://www.practo.com/delhi/doctor/dr-ranjan-upadhyay-dermatologist-cosmetologist-1-6350?specialization=Dermatologist&practice_id=649436
Dr. Gayatri Bala Juneja,https://www.practo.com/delhi/doctor/dr-gayatri-bala-juneja-gynecologist-obstetrician-2?specialization=Gynecologist/Obstetrician&practice_id=1010694

B. Project Outcome Images

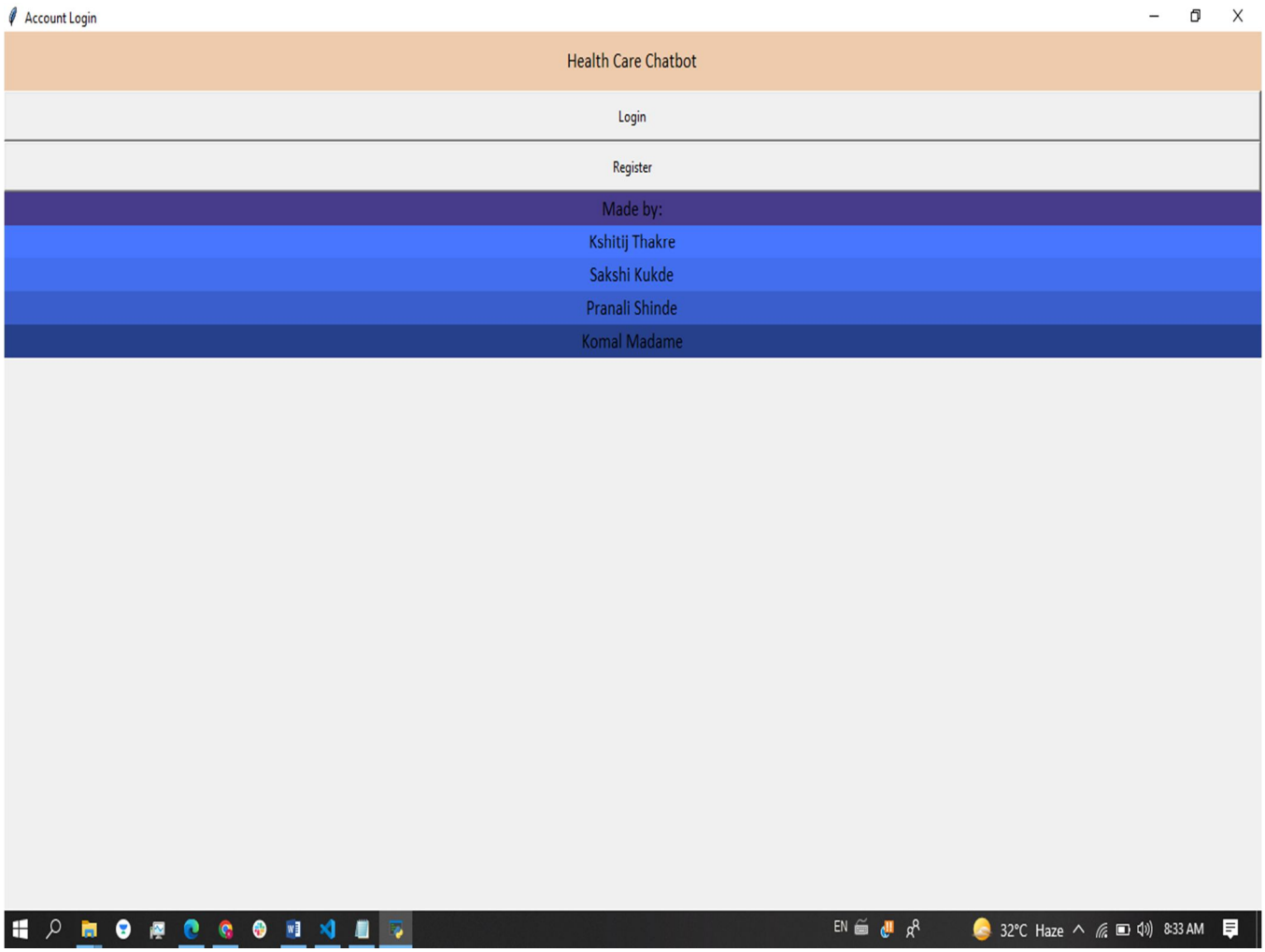


Fig.4.3 Home page

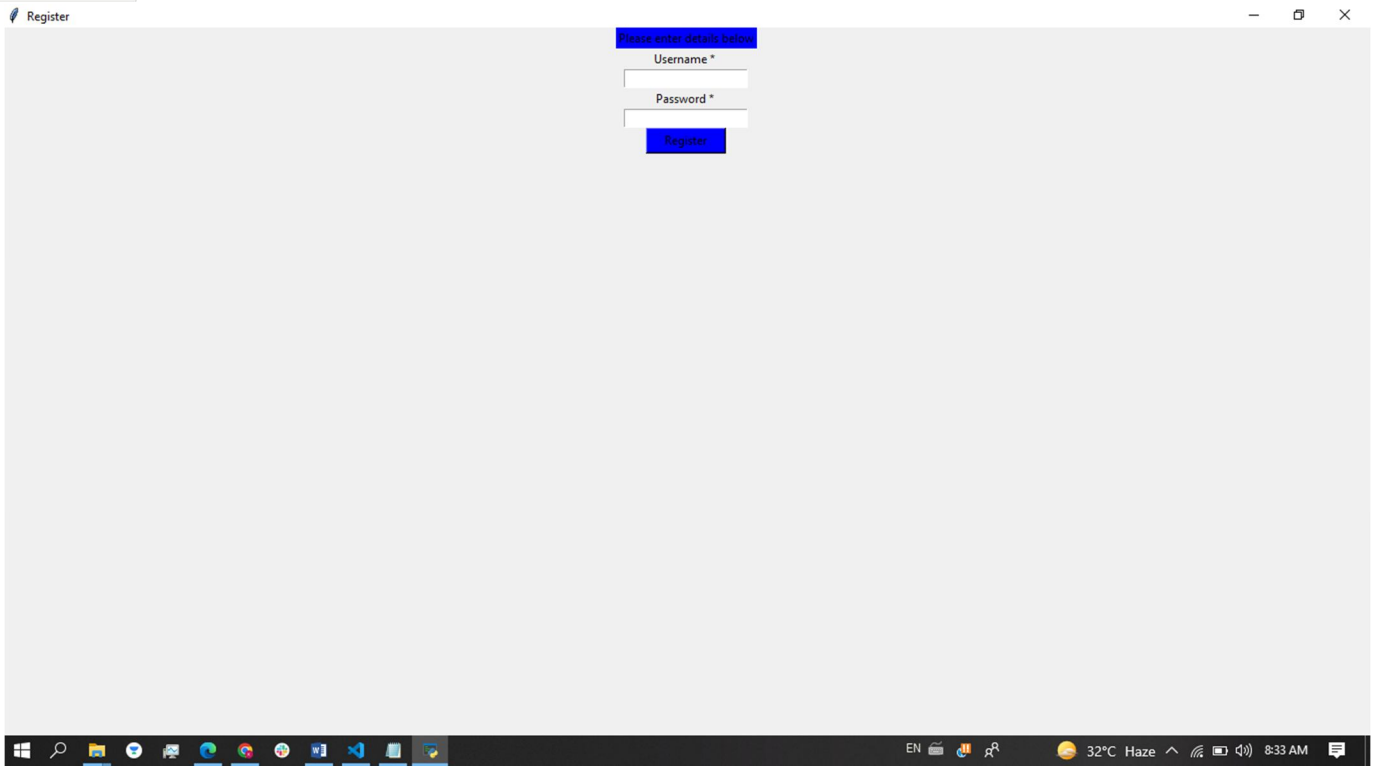


Fig.4.4 Registration Page

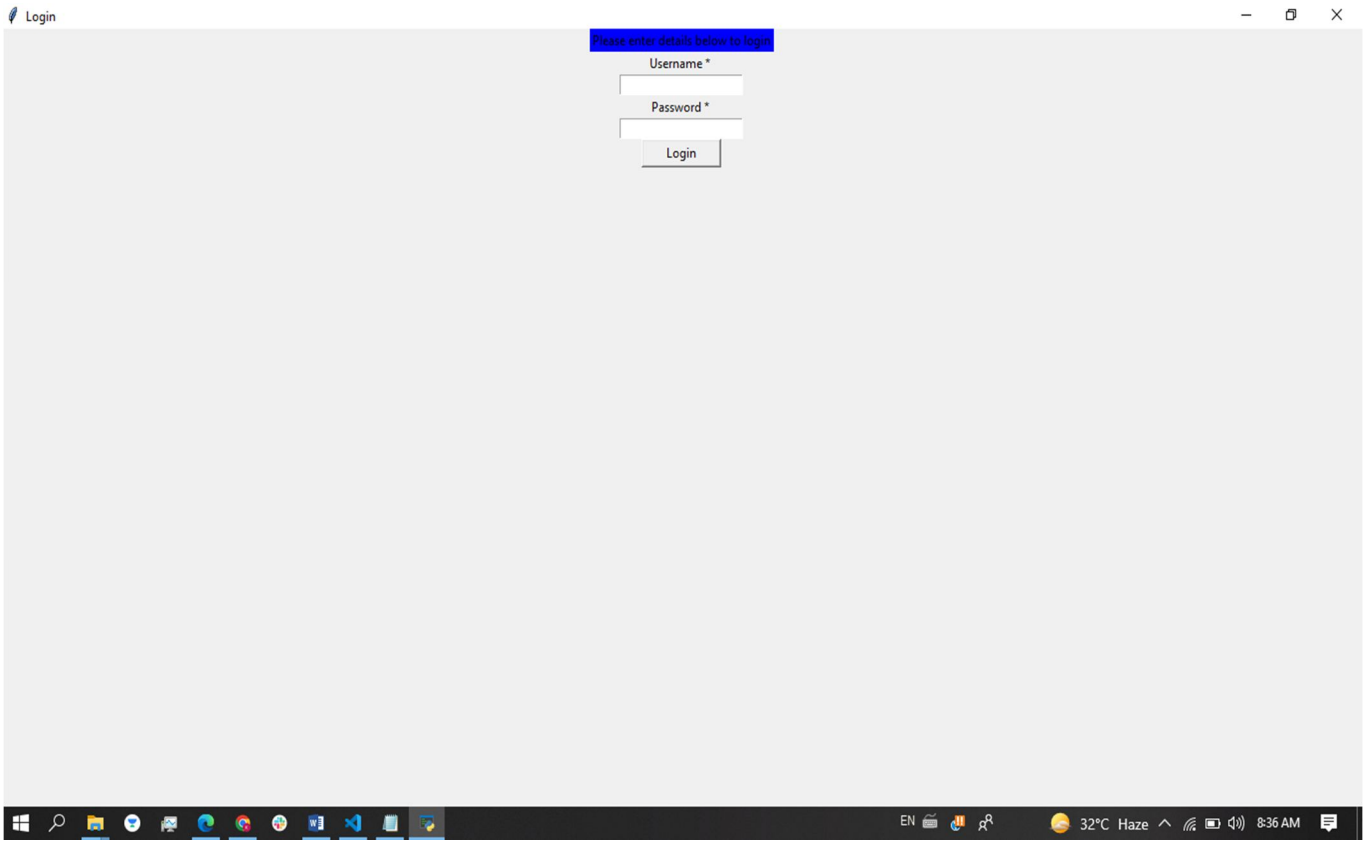


Fig 4.5 Login Page

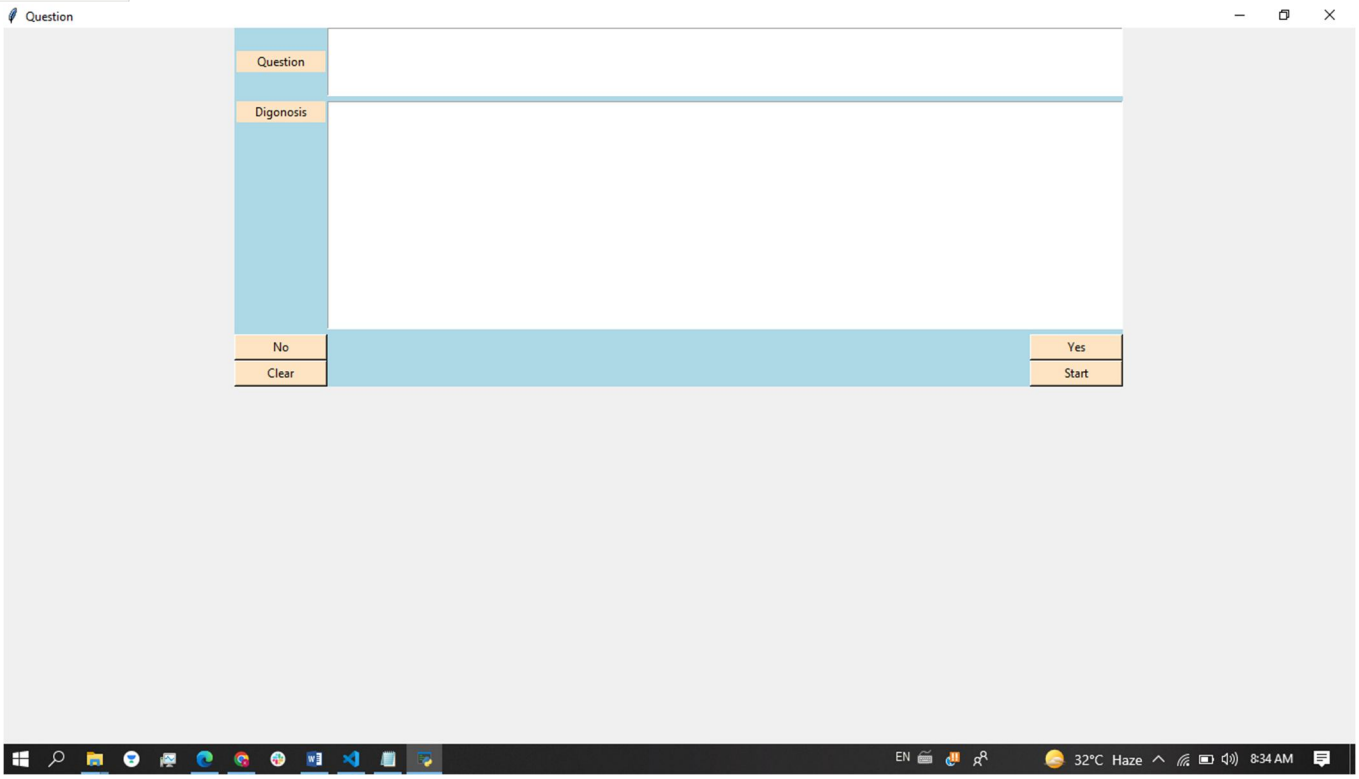


Fig 4.6 Main Dashboard of ChatBot

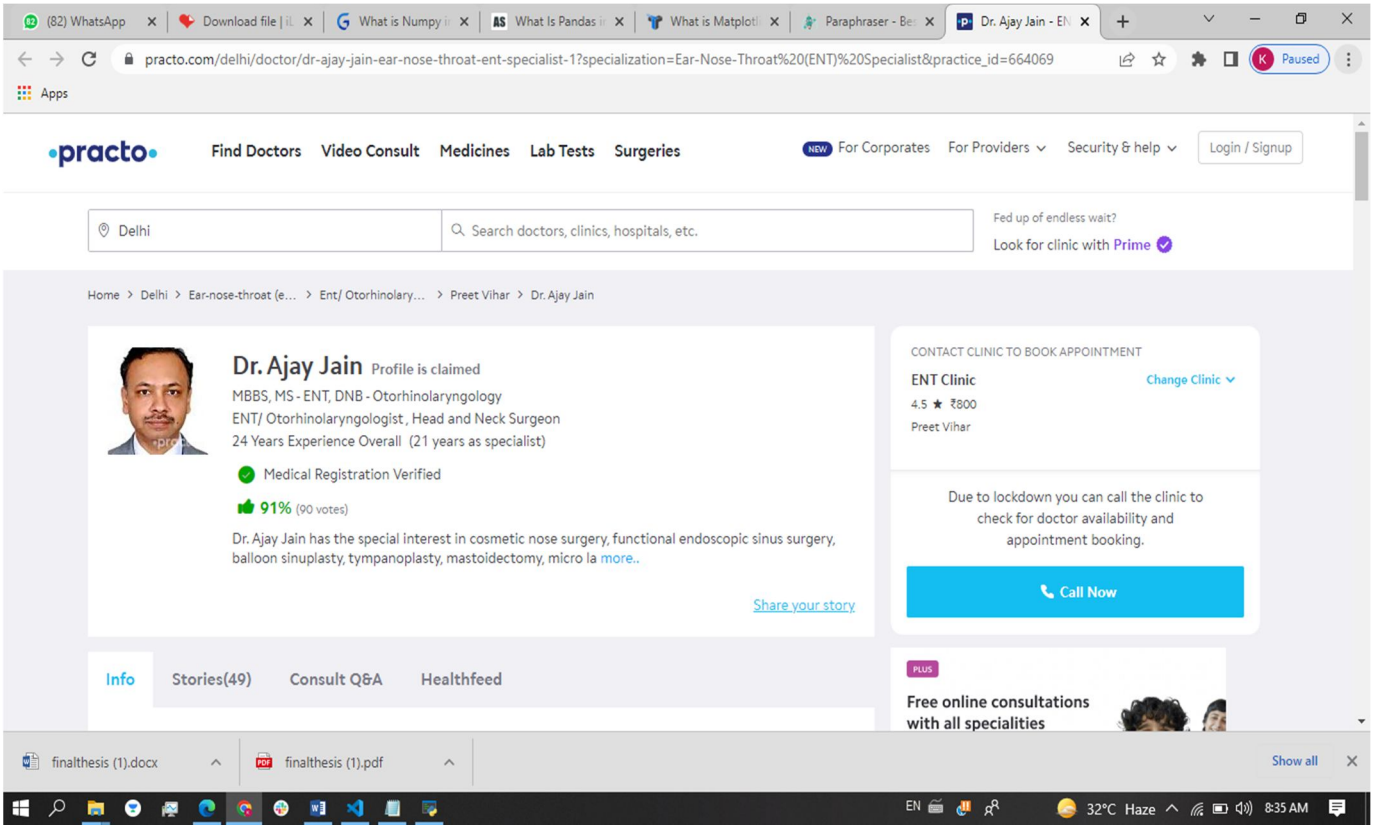


Fig. 4.7 Practo Dashboard

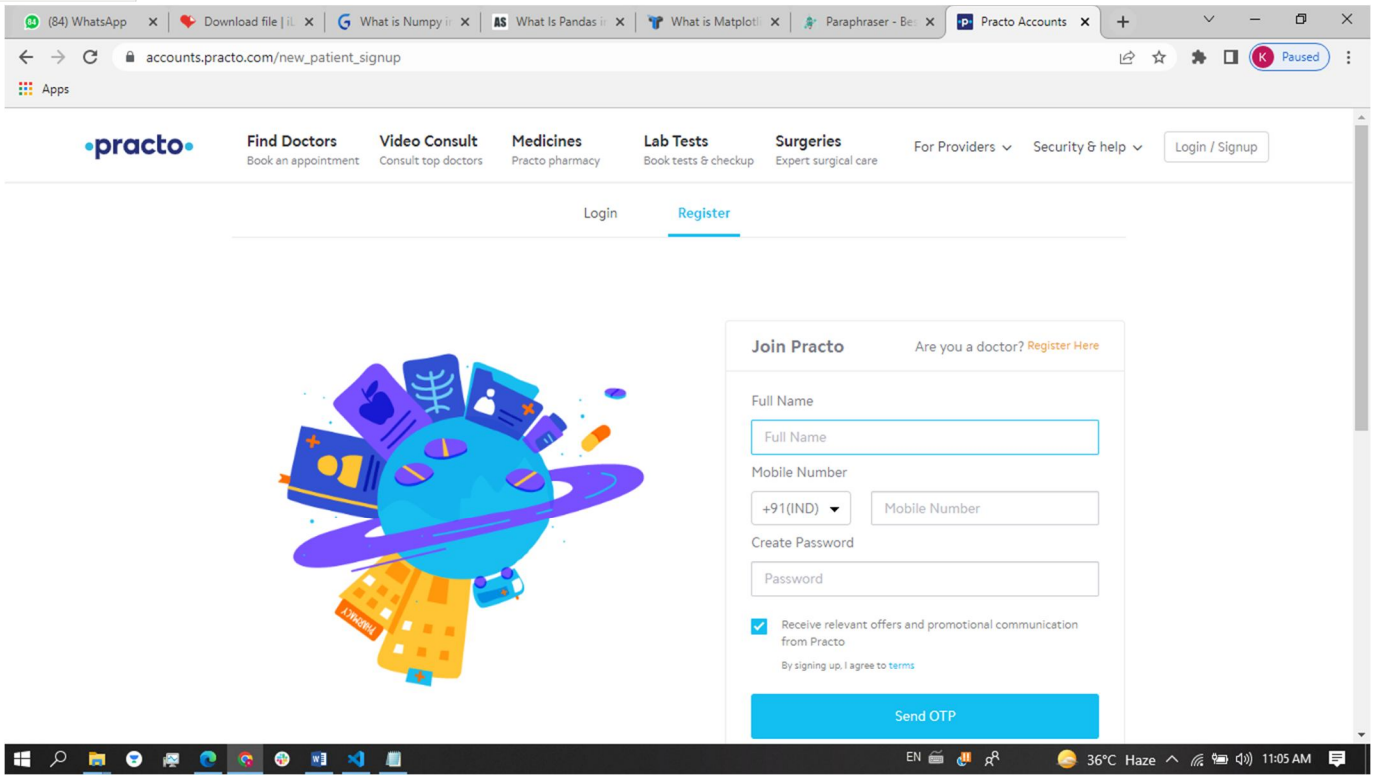


Fig. 4.8 Practo Register Page

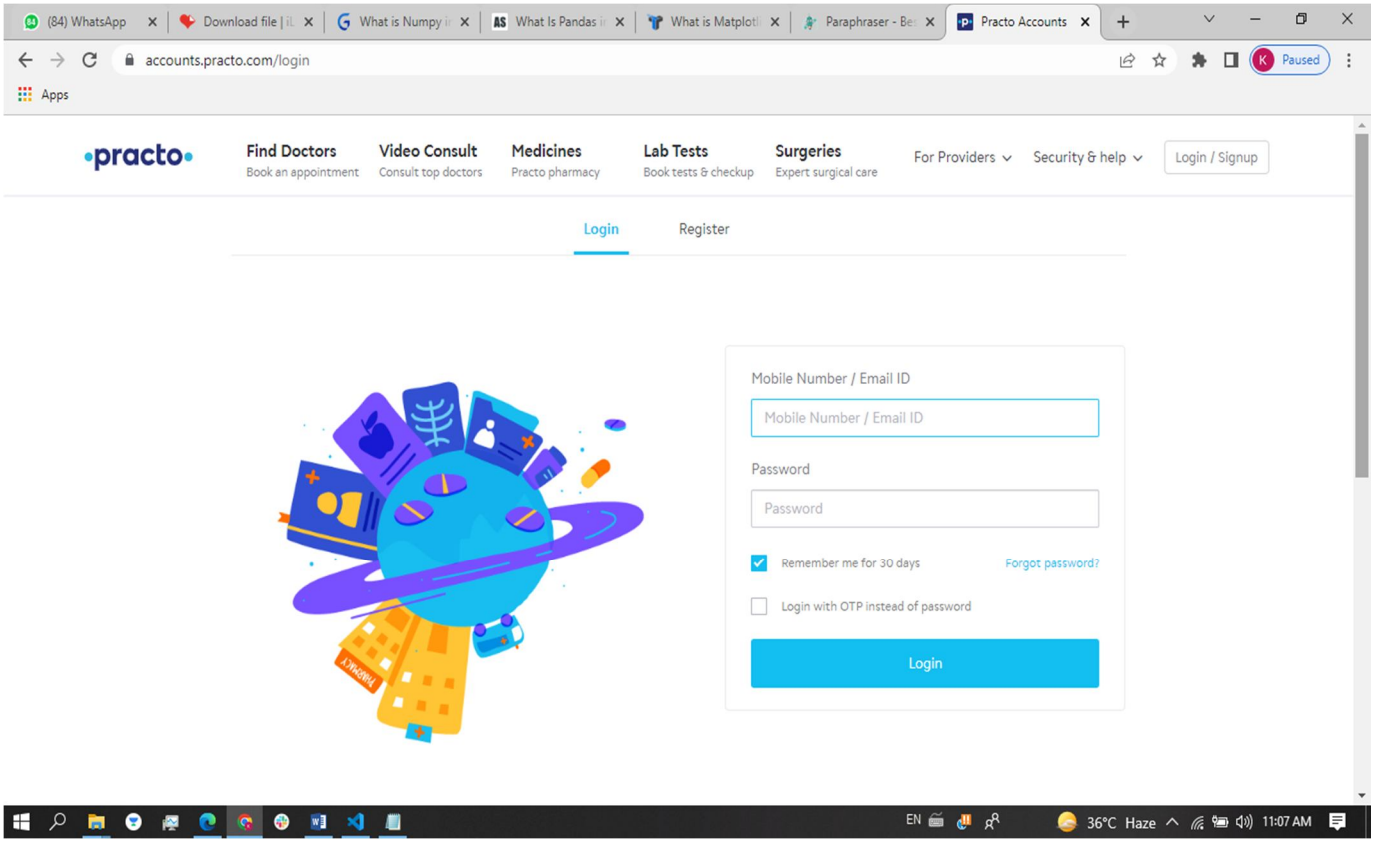


Fig. 4.9 Practo Login Page

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue V May 2022- Available at www.ijraset.com

VI. CONCLUSIONS AND FUTURE SCOPE

A. *Conclusions*

Our medical chatbot provides medical help to the patients for a few general diseases like fever, cold, typhoid, malaria, jaundice, etc. we have a tendency to area unit inventing the system owing to the requirement for the increasing population of our country. Such systems area unit obtainable in foreign however not in our country. As we all know well regarding it that the numbers of doctors area unit less to serve the requirement of the patient. This state of affairs may be higher understood by walking through the city's government hospitals. Thus, the medical chatbot can provide medical help to the patients whereas the doctor isn't obtainable which is able to ultimately improve the potency & performance of the medical business by decreasing the death rate. the applying of chatbots within the medical domain is sort of the simplest way on the far side our imagination. we've got lined most the points that a medical chatbot ought to support to cater to the requirement of the patient. within the past few years, there area unit plenty of models of the medical chatbot are fancied that were quite dear for a traditional person however we've got tried to beat this disadvantage in our 'health care chatbot system

B. *Future Scope*

The future era is the era of messaging apps as a result of people spending a longer time on messaging apps than the other apps. The implementation of personalized drugs would with success save several lives and build medical awareness among the people. No matter how far people are, they will have this medical voice communication. The sole demand is they have an easy desktop or smartphone with an active web association. The economy of the chatbot will be improved by adding a lot of combinations of words and increasing the use of database information so the medical chatbot may handle all types of diseases.

C. *Applications*

1) *Provide Medical Information:* Chatbot algorithms are trained on massive healthcare data including disease symptoms, diagnostics, Markers, and available treatments. Public datasets are used to continuously train chatbots, such as COVIDx for COVID-19 diagnosis, and Wisconsin Breast Cancer Diagnosis (WBCD). Conversational chatbots with different intelligence levels can understand the questions of the user and provide answers based on pre-defined labels in the training data.
   For example: Healthily app provides information about different diseases' symptoms, assessments of overall health, and tracks patient progress. Ada app provided correct disease suggestions earlier than the time of clinical diagnosis 56% of the time (Wikipedia)
2) *Schedule Medical Appointments:* Chatbots are integrated into the medical facility system to extract information about suitable physicians, available slots, and clinics and pharmacy working days. Chatbots ask patients about their current health issue, find matching physician and dentists, provide available time slots, and schedule, reschedule, and delete appointments for patients. Chatbots are also integrated into user's device calendars to send reminders and updates about medical appointments.
3) *Collect Patient Data:* Chatbots can extract patient information using simple questions about name, address, symptoms, current doctor, and insurance details. Chatbots then store this information in the medical facility system to facilitate patient admission, symptom tracking, doctor-patient communication, and medical record keeping.
4) *Handle Insurance Inquiries:* Chatbots can provide insurance services and healthcare resources to patients and plan members. Moreover, integrating RPA or other automation solutions with chatbots allows automating insurance claim processing and healthcare billing. To read more on how RPA can automate insurance processes, feel free to read our article RPA in Insurance Industry: Use Cases & Case Studies [2021]
5) *Provide Mental health Assistance:* Chatbots that provide mental health assistance are trained to to deliver cognitive behavioral therapy (CBT) for patients with depression, post traumatic stress disorder (PTSD), and anxiety, or train autistic patients to improve their social skills and job interview skills. Users can interact with chatbots via text, microphones and cameras.
   For example, Woebot, which we listed among successful chatbots, provides CBT, mindfulness, and Dialectical Behavior Therapy (CBT).
6) *Request prescripTion Refills:* Chatbots collect patient information, name, birthday, contact information, current doctor, last visit to the clinic, and prescription information. The chatbot submits a request to the patient's doctor for a final decision and contacts the patient when a refill is available. This allows doctors to process prescription refills in batch or automate them in cases where doctor intervention is not necessary.

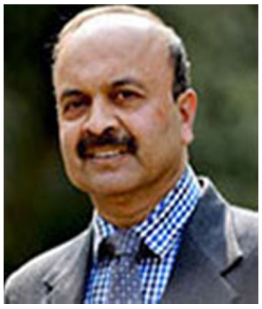
©IJRASET: All Rights are Reserved

REFERENCES

[1] K. Oh, D. Lee, B. Ko and H. Choi, "A Chatbot for Psychiatric Counseling in Mental Healthcare Service Based on Emotional Dialogue Analysis and Sentence Generation," 2017 18th IEEE International Conference on Mobile Data Management (MDM), Daejeon, 2017, pp. 371-375. DOI: 10.1109/MDM.2017.6
[2] Du Preez, S.J. & Lall, Manoj & Sinha, S. (2009). An intelligent web-based voice chat bot. 386 - 391.10.1109/EURCON.2009.5167660
[3] Bayu Setiaji, Ferry Wahyu Wibowo, "Chatbot Using a Knowledge in Database: Human-to-Machine Conversation Modeling", Intelligent Systems Modelling
[4] Dahiya, Menal. (2017). A Tool of Conversation: Chatbot. INTERNATIONAL JOURNAL OF COMPUTER SCIENCES AND ENGINEERING. 5. 158-161.2017
[5] Mrs. Rashmi Dharwadkar1, Dr.Mrs. Neeta A. Deshpande, A Medical ChatBot, International Journal of Computer Trends and Technology (IJCTT) – Volume 60 Issue 1-June

APPENDICES

1) *Certificate of Participation*
2) *Project Team at a Glance*

SR. NO.	NAME	CONTACT NO. , EMAIL ID & ADDRESS	PHOTO
1.	DR.P.R.ROTHE (PROJECT GUIDE)	9822579845 p_rrothe@rediffmail.com 219,Empress Mill Society Shree Nagar Nagpur 440015	
2.		7757928168 kshitijdthakre@gmail.com 48,GandhiIeyout, North Ambazari Road , Nagpur	
3.	ADITYA ROBIN HIREKHAN	8329091140 sakshikukade13@gmail.com Plot No 46 ,Ishawar Nagar, Nagpur, 440009	

4.	GAURAV CHANGDEO GIRADKAR	8080407599 komalmadame9@gmail.com Santoshi Ley Out ,Dattawadi Nagpur,440023	
5.	JYOTI PRABHATKUMAR RANJAN	9049969545 Spranali1008@gmail.com Gandhi Putla CA road Itwari Nagpur	

3) Mapping Of Project Outcomes With Program Outcomes

Project Relevance with Project Outcomes

Priyadarshini College of Engineering, Nagpur

Department of Electronics & Telecommunication Engineering Final YearProject

Year of Study: 2021 – 2022 (8th Sem) (Section- A)

Title of Project: "HEALTH CARE CHATBOT USING NLP AND FLASK "

Project Outcomes

By the end of the course, the students will be able to

PO1	Answering the Frequently asked question.
PO2	Will be able to study the symptoms of the user.
PO3	Use to track the health condition of the user.
PO4	Users will be able to schedule appointments.
PO5	User will be able to Login into the chatbot system with his unique username and password.

Program outcomes

After Completion of course, graduate will able to

1) *Engineering Knowledge:* Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2) *Problem Analysis:* Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics natural sciences, and engineering sciences.
3) *Design/Development of Solutions:* Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4) *Conduct Investigations of Complex Problems :* Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of theinformation to provide valid conclusions
5) *Modern tool Usage:* Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools

including prediction and modeling to complex engineering activities with an understanding of the limitations.

6) *The Engineer and Society:* Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7) *Environment and Sustainability:* Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainabledevelopment.

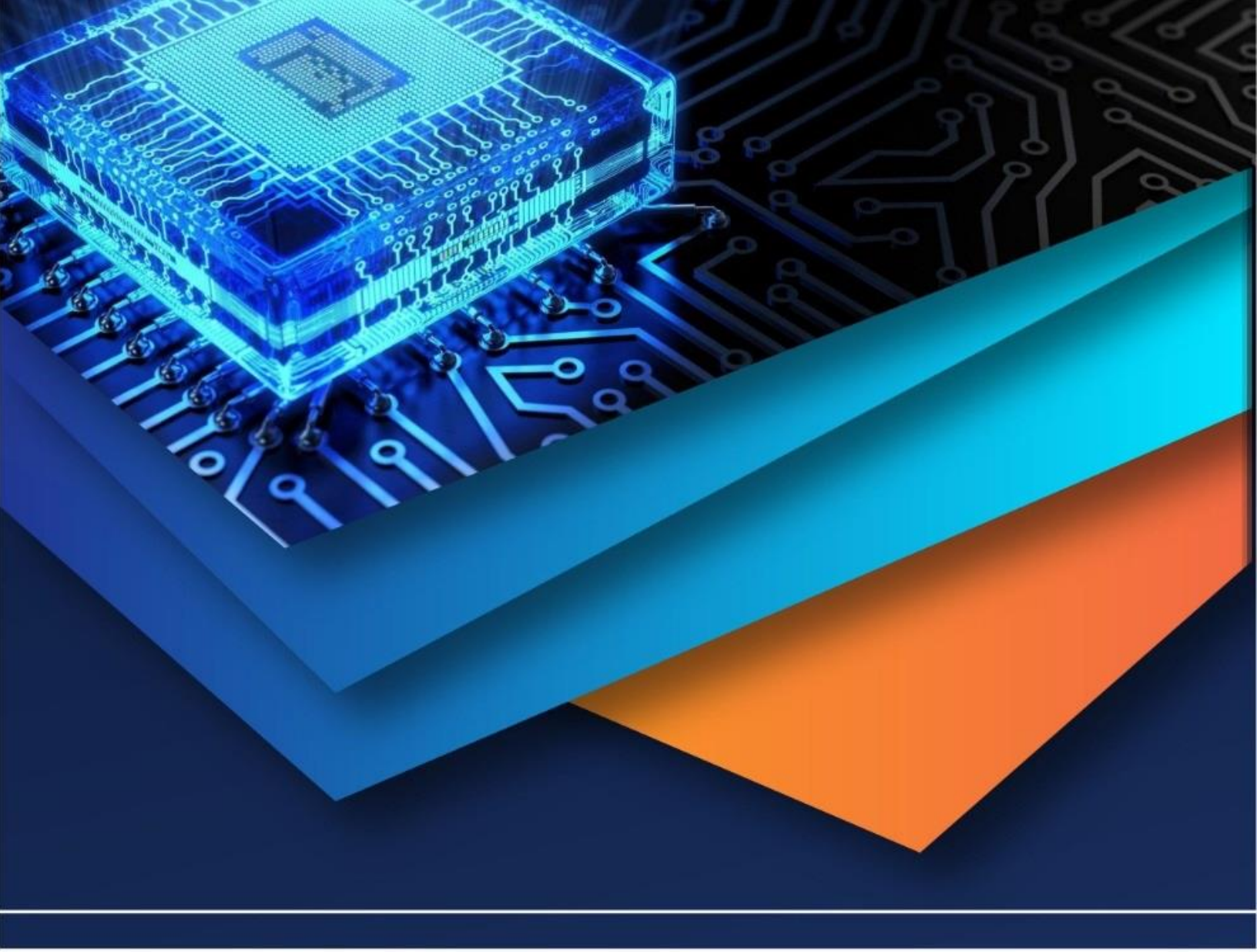
Subject	Course Outcomes	Program Outcomes											
		1	2	3	4	5	6	7	8	9	10	11	12
Final Year Project	Answering the Frequently asked question.	3	2	1		1		3				3	3
	Will be able to study the symptoms of the user.	2		3		3	2					3	3
	Use to track the health condition of the user.	1		3		3	3					2	3
	Users will be able to schedule appointments.	3	2		2	3						3	3
	User will be able to Login into the chatbot system with his unique username and password	1	3		1	2						2	3

8) *Ethics:* Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9) *Individual and Team Work:* Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10) *Communication:* Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instruction.
11) *Project Management and Finance:* Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12) *Life-long Learning:* Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Mapping of Project outcomes with program outcomes
Mapping of Project Outcomes with Program Specific Outcomes

Subject	Course Outcomes	PSOs	
		1	2
Final Year Project	Answering the Frequently asked question.	2	3
	Will be able to study the symptoms of the user.	2	3
	Use to track the health condition of the user.	1	3
	Users will be able to schedule appointments.	1	2
	User will be able to Login into the chatbot system with his unique username and password	2	2

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)