AI Based Healthcare Solution

A

Project Report

submitted

in partial fulfillment

for the award of the Degree of

Bachelor of Technology

In

Department of Computer Science and Engineering

Submitted To:

Prof. Kapil Dev Sharma

(HoD, CSE)

Submitted By:

Mr. Sourabh Tiwari
Mr. Deepak Soni
Mr. Ribabul Ali
Mr. Amrtya Srivastava
Md. Shakeel Ahamad

Department Of Computer Science and Engineering
Rajasthan Institute of Engineering & Technology
Rajasthan Technical University
October 2019-2020

Candidate’s Declaration

I hereby certify that the work embodied in this dissertation entitled **“AI Based Healthcare System”** by **Sourabh Tiwari, Deepak Soni, Amrtya Strivasta, Shakeel Ahamad, Ribabul Ali** is towards partial fulfillment of the requirements for the award of the degree of B.Tech. in Computer Science Engineering submitted to **Department of Computer Science and Engineering, Rajasthan Institute of Engineering and Technology**, **Jaipur** is an authentic record of my own work carried out under the supervision of **Mr Saurabh Sharma and Mr Rahul Sharma** The matter presented in this dissertation has not been submitted by me in any other University/Institute for the award of any other degree or diploma.

(Sourabh Tiwari, Deepak Soni, Amrtya Srivastava, Shakeel Ahamad, Ribabul Ali)

CERTIFICATE

This is to certify that **Mr. Sourabh Tiwari, Mr. Deepak Soni, Mr. Amrtya Srivastava, Mr. Ribabul Ali and Mr. Md Shakeel Ahamad** the students of "Bachelor of Technology" in Department of Computer Science And Engineering, **VII** Semester have submitted their project entitled **"Artificial Intelligence and based healthcare solution"** under my guidance.

Mr. Ravi Gupta

(Project Manager)

ACKNOWLEDGMENT

We wish to express our deep sense of gratitude and indebtedness to Prof. Kapil Sharma, Department of Computer Engineering, RIET Jaipur, for introducing the present topic and for his inspiring guidance, constructive criticism and valuable suggestion throughout this project work..

We also extend our sincere thanks to all our friends who have patiently helped us in accomplishing this undertaking. We also thank Mr.Sourabh Sharma and Mr. Rahul Sharma for their constant support throughout the project work.

Date:

Place:

Deepak Soni (16ERECS012)

Amrtya Srivastava (16ERECS004)

Sourabh Tiwari (16ERECS043)

Md. Shakee Ahmad (16ERECS022)

Ribabul Ali (16ERECS034)

TABLE OF CONTENTS

ABSTRACT

Normally Users are not aware about all the treatment or symptoms regarding the particular disease. For small problem user have to go personally to the hospital for check-up which is more time consuming. Also handling the telephonic calls for the complaints is quite hectic. Such a problem can be solved by using medical ChatBot by giving proper guidance regarding healthy living.

The medical chat-bots functioning depends on Natural language processing that helps users to submit their problem about the health. The User can ask any personal query related to health care through the chat-Bot without physically available to the hospital.Query is sent to ChatBot and gets related answer and display answer on Web Page(Ginger). The System's major concern behind developing this web based platform is analysing customer's sentiments.

To lead a good life healthcare is very much important. But it is very difficult to obtain the consultation with the doctor in case of any health issues. The proposed idea is to create a medical chatbot using Artificial Intelligence that can diagnose the disease and provide basic details about the disease before consulting a doctor .To reduce the healthcare costs and improve accessibility to medical knowledge the medical chatbot is built. Certain chatbots acts as a medical reference books,which helps the patient know more about their disease and helps to improve their health.The user can achieve the real benefit of a chatbot only when it can diagnose all kind of disease and provide necessary information. A text-to-text diagnosis bot engages patients in conversation about their medical issues and provides a personalized diagnosis based on their symptoms. Hence, people will have an idea about their health and have the right protection.

Keywords - Medical Chatbot Natural Language Processing, KNN Algorithm, Logistic Regression Algorithm, Linear Regression Algorithm, Naive Bayes Algorithm ,Word Order Similarity Between Sentences.

CHAPTER 1

INTRODUCTION

The main purpose of the scheme is to build the language gap between the user and health providers by giving immediate replies to the Questions asked by the user. Today's people are more likely addicted to internet but they are not concern about their personal health. They avoid to go in hospital for small problem which may become a major disease in future. Establishing question answer forums is becoming a simple way to answer those queries rather than browsing through the list of potentially relevant document from the web. Many of the existing systems have some limitation such as There is no instant response given to the patients they have to wait for experts acknowledgement for a long time. Some of the processes may charge amount to perform live chat or telephony communication with doctors online.

This system allows computer to communication between human to computer by using natural language processing (NLP). There are three analyses which understand natural language i.e. identification of main linguistic relations is completed to parse subject into object of the sentences. After thatdescription of the texts is done. The semantic interpretation uses knowledge of word meaning.

Chatbot is an Entity which imitate human discussion in its particular accepted set-up together with a text or vocal language with techniques such as Natural Language Processing (NLP). The aim of this system is to replicate a person's discussion. The development of chatbot application canbe done with making a user interface to send input and receive response. It is a system that interact with user by keeping the track of the state of interaction and recollecting the preceding commands to give functionality. The medical chat-bots can be developed by using artificial algorithms that scrutinize user's queries and recognize it and give reply to related query. A big disease can start from small problems such as headache which feels normal but it may beginning of big disease such as brain tumor .most of the disease can be identified by common symptoms so the disease can be predicted if the patient body is analyzed periodically.The system give response by use of an efficient Graphical User Interface such that if actual person is chatting with the user. chatterbot that can be used in various fields like education, healthcare, and route assistance.The central part of the chat-bots includes MySQL.

You can ask medical dosage related queries to this app in voice and system gets output for medicine API and speak out and display all data. Get your age from registration data and provide data related to your data like age, area, gender and so on. Give me age Then predict disease using KNN Algorithm, Logistic Regression Algorithm, Linear Regression Algorithm, Naive Bayes Algorithm.

CHAPTER 2

THE PLATFORM

2.1 Dialogflow

Dialogflow (formerly Api.ai, Speaktoit) is a Google-owned developer of human–computer interaction technologies based on natural language conversations. The company is best known for creating the Assistant (by Speaktoit), a virtual buddy for Android, iOS, and Windows Phone smartphones that performs tasks and answers users' question in a natural language. Speaktoit has also created a natural language processing engine that incorporates conversation context like dialogue history, location and user preferences.

Why build a chatbot?

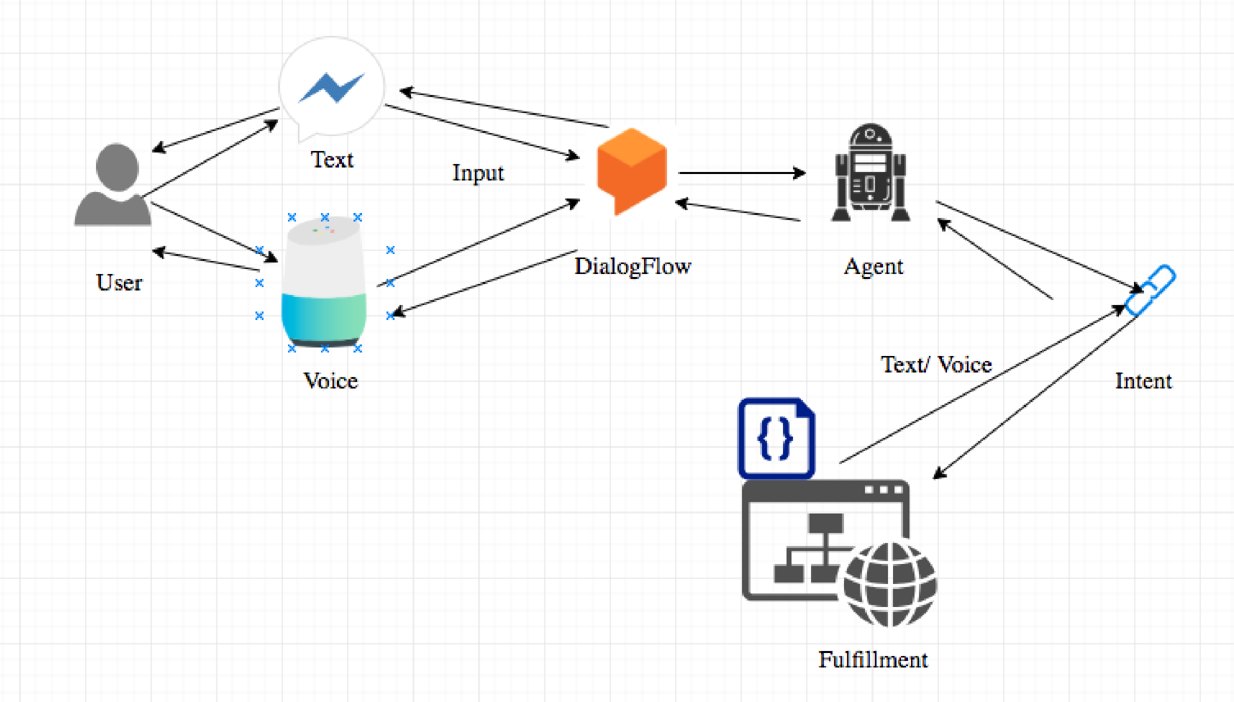
A chatbot is, in essence, a piece of robotic software used to imitate human conversation through text chats and voice commands (a good example being Siri or Amazon Alexa). A.I. based chat bots (learn over a period of time using Machine Learning techniques) — dialog flow is an example of that.

Text / Voice : The user interacts with an app like facebook messenger / google home to start the interaction with the bot.

Agent: A module within dialogflow which incorporates Natural Language Processing to understand what the user meant and to figure out what "action" has to be carried out. The agent transforms the user request into machine readable actionable data.

Intent: Support or the service that the user wants from the agent. Intent is configured by the developers. Intent determines the action by the code.

Fulfillment: This is the code. This part of the conversation lets you pass on the request from your bot to an external source and get response and pass it back to the user. This is achieved via Webhook. Setting up a webhook allows you to pass information from a matched intent into a web service and get a result from it.



2.2 Firebase

Firebase provides a realtime database and backend as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored on Firebase's cloud.The company provides client libraries that enable integration with Android, iOS, JavaScript, Java, Objective-C, Swift and Node.js applications. The database is also accessible through a REST API and bindings for several JavaScript frameworks such as AngularJS, React, Ember.js and Backbone.js.The REST API uses the Server-Sent Events protocol, which is an API for creating HTTP connections for receiving push notifications from a server. Developers using the realtime database can secure their data by using the company's server-side-enforced security rules.

2.3 Postman

Postman is an API(application programming interface) development tool which helps to build, test and modify APIs. Almost any functionality that could be needed by any developer is encapsulated in this tool. It is used by over 5 million developers every month to make their API development easy and simple. It has the ability to make various types of HTTP requests(GET, POST, PUT, PATCH), saving environments for later use, converting the API to code for various languages(like JavaScript, Python).

In this post, I will use the Postman software to send and recieve requests, POST data to the server and I will try to demo some other popular maneuvers. You can treat this article as your first contact with the Postman. So, lets get started ! You can download Postmanfrom here. After downloading and installing the Postman, open the software.

The longest middle input field that looks something like a search bar is where the URL that we want to GET or POST or DELETE,etc. is fed.
Just to the left of it, is a drop down button which has all the various HTTP methods as options. If you want to POST to the URL that you have specified, select POST.
To the right of it is the params button. If you click on it, a new interface will appear. Params are basically the data that we want to send to the server with our request. We will use this params interface to **POST** to put app a new User.
To the left of this button is the **Send** button which is used in sending the request to the server or the app in this case.

2.5 Django

Django is a free, open source web framework written in the Python programming language and used by millions of programmers every year. Its popularity is due to its friendliness to both beginners and advanced programmers: Django is robust enough to be used by the largest websites in the world–Instagram, Pinterest, Bitbucket, Disqus–but also flexible enough to be a good choice for early-stage startups and prototyping personal projects.A web framework is a collection of modular tools that abstracts away much of the difficulty–and repetition–inherent in web development. For example, most websites need the same basic functionality: the ability to connect to a database, set URL routes, display content on a page, handle security properly, and so on. Rather than recreate all of this from scratch, programmers over the years have created web frameworks in all the major programming languages: Django and Flask in Python, Rails in Ruby, and Express in JavaScript among many, many others.Django inherited Python's "batteries-included" approach and includes out-of-the box support for common tasks in web development:

- user authentication
- templates, routes, and views
- admin interface
- robust security
- support for multiple database backends

CHAPTER 3

PROGRAMING

3.1 Code In Dialogflow

```
'use strict';
const axios = require('axios');
const fetch = require('node-fetch');
const functions = require('firebase-functions');
const admin=require('firebase-admin');
const {WebhookClient} = require('dialogflow-fulfillment');
const {Card, Suggestion} = require('dialogflow-fulfillment');
admin.initializeApp(functions.config().firebase);
var db=admin.firestore();
global.name_main='';
global.email_main='';
global.phone_number_main='';
global.Symp_Item=[];
global.arr = [];
process.env.DEBUG = 'dialogflow:debug'; // enables lib debugging statements
exports.dialogflowFirebaseFulfillment = functions.https.onRequest((request, response) => {
  const agent = new WebhookClient({ request, response });
  console.log('Dialogflow Request headers: ' + JSON.stringify(request.headers));
  console.log('Dialogflow Request body: ' + JSON.stringify(request.body));
   function welcome(agent) {
    agent.add(`Welcome to my agent!`);
  }
  function fallback(agent) {
    agent.add(`I didn't understand`);
   agent.add(`I'm sorry, can you try again?`);
  }
  function handleReadFromDB(agent){
          return admin.database().ref('data').once('value').then((snapshot)=> {
const value_symp = snapshot.child('symptoms').val();
   const value_name = snapshot.child('Name').val();
     if(value_symp!==null){
     agent.add(`You Have Entered your Symptoms as ${value_symp}`);
     }
     if(value_name!==null){
     agent.add(`Your Name is ${value_name}`);
     }
   });
  }
  function handlemedical(agent){
  global.name_main=agent.parameters.name;
   global.email_main=agent.parameters.email;
   global.phone_number_main=agent.parameters.number;
  }
  function handlebackpain(agent){
   global.Symp_Item.push('back pain');
 }
  function handlemildfever(agent){
    global.Symp_Item.push('mild fever');
  }
 function handlePermission(agent){
  //agent.add(`${global.name_main} you have following Symptoms :
${global.Symp_Item}`);
  const url = "http://134.119.194.12:5000/disease/";
  const data =
    {
    "symptoms" : global.Symp_Item,
    "name" : global.name_main,
    "email" : global.email_main,
    "phone" : global.phone_number_main
  }
  return axios.post(url, data)
  .then(jsonBody => {
     //var body = JSON.parse(jsonBody);
     agent.add("Probable Disease : - "+ jsonBody.data.prediction[0]);
     agent.add("Confidence Level : - "+ jsonBody.data.prediction[1]);
     console.log(jsonBody.data.prediction);
     return Promise.resolve(agent);
   })
  .catch(err => {
          console.error('Problem making network call', err);
          agent.add('Unable to get result');
          return Promise.resolve(agent);
      });
 }
  let intentMap = new Map();
  intentMap.set('RhymingWord', rymingwordhandler);
  intentMap.set('Default Welcome Intent', welcome);
…
…
…
…
 intentMap.set('Default Fallback Intent', fallback);
intentMap.set('Medical',handlemedical);
 agent.handleRequest(intentMap);});
```

3.2 Code In Python

```
import numpy as np
import pandas as pd
```

```
          def disease_prediction(psymptoms):
              disease_set = {}
              file_path="C:\\Users\\Amrtya
          srivastav\\PycharmProjects\\MedicalApi\\DiseaseApi\\"
              def DecisionTree(psymptoms):
                  from sklearn import tree
                  clf3 = tree.DecisionTreeClassifier()  # empty model of the decision tree
                  clf3 = clf3.fit(X, y)
                  # calculating accuracy-----------------------------------------------------------------
-
                  from sklearn.metrics import accuracy_score
                  y_pred = clf3.predict(X_test)
                  #     print('Accuracy:-',accuracy_score(y_test, y_pred))
                  #     print(accuracy_score(y_test, y_pred,normalize=False
                 # ----------------------------------------------------
                  for k in range(0, len(l1)):
                      # print (k,)
     for z in psymptoms:
                          if (z == l1[k]):
                              l2[k] = 1
                  inputtest = [l2]
                  predict = clf3.predict(inputtest)
                  predicted = predict[0]
                  h = 'no'
                  for a in range(0, len(disease)):
                      if (predicted == a):
                          h = 'yes'
                          break
                  if (h == 'yes'):
                      return disease[a], accuracy_score(y_test, y_pred)
                  else:
                      return "Not Found"
              def LogisticReg(psymptoms):
                  from sklearn.linear_model import LogisticRegression
                  clf4 = LogisticRegression()
                  clf4 = clf4.fit(X, np.ravel(y))
                  # calculating accuracy-----------------------------------------------------------------
-
```

```
            from sklearn.metrics import accuracy_score
                y_pred = clf4.predict(X_test)
                #   print('Accuracy:-',accuracy_score(y_test, y_pred))
                #   print(accuracy_score(y_test, y_pred,normalize=False))
                # -----------------------------------------------------
                for k in range(0, len(l1)):
                    for z in psymptoms:
                        if (z == l1[k]):
                            l2[k] = 1
                inputtest = [l2]
                predict = clf4.predict(inputtest)
                predicted = predict[0]
                h = 'no'
                for a in range(0, len(disease)):
                    if (predicted == a):
                        h = 'yes'
                        break
                if (h == 'yes'):
                    return disease[a], accuracy_score(y_test, y_pred)
else:
                    return "Not Found"
              def knn(psymptoms):
                from sklearn.neighbors import KNeighborsClassifier
                knnobj = KNeighborsClassifier()
                knnobj = knnobj.fit(X, np.ravel(y))
                # calculating accuracy--------------------------------------------------------------------
    -
                from sklearn.metrics import accuracy_score
                y_pred = knnobj.predict(X_test)
                #   print('Accuracy:-',accuracy_score(y_test, y_pred))
                #   print(accuracy_score(y_test, y_pred,normalize=False))
                # -----------------------------------------------------
                for k in range(0, len(l1)):
                    for z in psymptoms:
                        if (z == l1[k]):
                            l2[k] = 1
                inputtest = [l2]
                predict = knnobj.predict(inputtest)
```

```
        predicted = predict[0]
h = 'no'
        for a in range(0, len(disease)):
            if (predicted == a):
                h = 'yes'
                break

        if (h == 'yes'):
            return disease[a], accuracy_score(y_test, y_pred)
        else:
            return "Not Found"
    def NaiveBayes(psymptoms):
        from sklearn.naive_bayes import GaussianNB
        gnb = GaussianNB()
        gnb = gnb.fit(X, np.ravel(y))
        # calculating accuracy-----------------------------------------------------------------
-
        from sklearn.metrics import accuracy_score
        y_pred = gnb.predict(X_test)
        #     print('Accuracy:-',accuracy_score(y_test, y_pred))
        #     print(accuracy_score(y_test, y_pred,normalize=False))
        # -----------------------------------------------------
        for k in range(0, len(l1)):
            for z in psymptoms:
                if (z == l1[k]):
                    l2[k] = 1
        inputtest = [l2]
        predict = gnb.predict(inputtest)
        predicted = predict[0]
        h = 'no'
        for a in range(0, len(disease)):
            if (predicted == a):
                h = 'yes'
                break
        if (h == 'yes'):
            return disease[a], accuracy_score(y_test, y_pred)
else:
            return "Not Found"
```

```
l1 = ['back pain', 'constipation', 'abdominal pain', 'diarrhoea', 'mild fever', 'yellow urine',
    'yellowing of eyes', 'acute liver failure', 'fluid overload', 'swelling of stomach',
    'swelled lymph nodes', 'malaise', 'blurred and distorted vision', 'phlegm', 'throat irritation',
    'redness of eyes', 'sinus pressure', 'runny nose', 'congestion', 'chest pain', 'weakness in limbs',
    'fast heart rate', 'pain during bowel movements', 'pain in anal region', 'bloody stool',
    'irritation in anus', 'neck pain', 'dizziness', 'cramps', 'bruising', 'obesity', 'swollen legs',
    'swollen blood vessels', 'puffy face and eyes', 'enlarged thyroid', 'brittle nails',
    'swollen extremeties', 'excessive hunger', 'extra marital contacts', 'drying and tingling lips',
    'slurred speech', 'knee pain', 'hip joint pain', 'muscle weakness', 'stiff neck', 'swelling joints',
    'movement stiffness', 'spinning movements', 'loss of balance', 'unsteadiness',
    'weakness of one body side', 'loss of smell', 'bladder discomfort', 'foul smell of urine',
  'continuous feel of urine', 'passage of gases', 'internal itching', 'toxic look (typhos)',
    'depression', 'irritability', 'muscle pain', 'altered sensorium', 'red spots over body', 'belly pain',
    'abnormal menstruation', 'dischromic  patches', 'watering from eyes', 'increased appetite', 'polyuria',
    'family history', 'mucoid sputum',
    'rusty sputum', 'lack of concentration', 'visual disturbances', 'receiving blood transfusion',
    'receiving unsterile injections', 'coma', 'stomach bleeding', 'distention of abdomen',
    'history of alcohol consumption', 'fluid overload', 'blood in sputum', 'prominent veins on calf',
    'palpitations', 'painful walking', 'pus filled pimples', 'blackheads', 'scurring', 'skin peeling',
    'silver like dusting', 'small dents in nails', 'inflammatory nails', 'blister', 'red sore around nose',
    'yellow crust ooze']
```

```
        disease = ['Fungal infection', 'Allergy', 'GERD', 'Chronic cholestasis', 'Drug
Reaction',
              'Peptic ulcer diseae', 'AIDS', 'Diabetes', 'Gastroenteritis', 'Bronchial
Asthma', 'Hypertension',
      ' Migraine', 'Cervical spondylosis',
              'Paralysis (brain hemorrhage)', 'Jaundice', 'Malaria', 'Chicken pox',
'Dengue', 'Typhoid', 'hepatitis A',
              'Hepatitis B', 'Hepatitis C', 'Hepatitis D', 'Hepatitis E', 'Alcoholic
hepatitis',
              'Common Cold', 'Pneumonia', 'Dimorphic hemmorhoids(piles)',
              'Heartattack', 'Varicoseveins', 'Hypothyroidism', 'Hyperthyroidism',
'Hypoglycemia', 'Osteoarthristis',
              'Arthritis', '(vertigo) Paroymsal  Positional Vertigo', 'Acne', 'Urinary tract
infection', 'Psoriasis',
              'Impetigo']

        # Prepare empty List as per the range of symptoms
        l2 = []
        for x in range(0, len(l1)):
          l2.append(0)

        # Reading Training data from CSV

        df1 = pd.read_csv(file_path+"Training.csv")
        column = []
      for i in df1.columns:
          column.append(i.replace("_", " "))
        df = pd.read_csv(file_path+'Training.csv', header=None, skiprows=1,
names=column)

        df.replace({'prognosis': {'Fungal infection': 0, 'Allergy': 1, 'GERD': 2, 'Chronic
cholestasis': 3, 'Drug Reaction': 4,
                      'Peptic ulcer diseae': 5, 'AIDS': 6, 'Diabetes ': 7,
'Gastroenteritis': 8,
                      'Bronchial Asthma': 9, 'Hypertension ': 10,
                      'Migraine': 11, 'Cervical spondylosis': 12,
                      'Paralysis (brain hemorrhage)': 13, 'Jaundice': 14, 'Malaria': 15,
'Chicken pox': 16,
                      'Dengue': 17, 'Typhoid': 18, 'hepatitis A': 19,
                    'Hepatitis B': 20, 'Hepatitis C': 21, 'Hepatitis D': 22, 'Hepatitis
E': 23,
                    'Alcoholic hepatitis': 24, 'Tuberculosis': 25,
                    'Common Cold': 26, 'Pneumonia': 27, 'Dimorphic
hemmorhoids(piles)': 28, 'Heart attack': 29,
                    'Varicose veins': 30, 'Hypothyroidism': 31,
                    'Hyperthyroidism': 32, 'Hypoglycemia': 33, 'Osteoarthristis': 34,
'Arthritis': 35,
      '(vertigo) Paroymsal  Positional Vertigo': 36, 'Acne': 37, 'Urinary tract infection':
38,
                    'Psoriasis': 39,
                    'Impetigo': 40}}, inplace=True)

        # print(df.head())
        # Create Feature and Label Matrix -

        X = df[l1]
        y = df[["prognosis"]]
        np.ravel(y)
        tr1 = pd.read_csv(file_path+"Testing.csv")
        column_test = []
        for i in tr1.columns:
          column_test.append(i.replace("_", " "))
        tr = pd.read_csv(file_path+'Testing.csv', header=None, skiprows=1,
names=column_test)
        tr.replace({'prognosis': {'Fungal infection': 0, 'Allergy': 1, 'GERD': 2, 'Chronic
cholestasis': 3, 'Drug Reaction': 4,
      'Peptic ulcer diseae': 5, 'AIDS': 6, 'Diabetes ': 7, 'Gastroenteritis': 8,
                    'Bronchial Asthma': 9, 'Hypertension ': 10,
                    'Migraine': 11, 'Cervical spondylosis': 12,
                    'Paralysis (brain hemorrhage)': 13, 'Jaundice': 14, 'Malaria': 15,
'Chicken pox': 16,
                    'Dengue': 17, 'Typhoid': 18, 'hepatitis A': 19,
                    'Hepatitis B': 20, 'Hepatitis C': 21, 'Hepatitis D': 22, 'Hepatitis
E': 23,
                    'Alcoholic hepatitis': 24, 'Tuberculosis': 25,
                    'Common Cold': 26, 'Pneumonia': 27, 'Dimorphic
hemmorhoids(piles)': 28, 'Heart attack': 29,
                    'Varicose veins': 30, 'Hypothyroidism': 31,
                              'Hyperthyroidism': 32, 'Hypoglycemia': 33, 'Osteoarthristis': 34,
  'Arthritis': 35
                              '(vertigo) Paroymsal  Positional Vertigo': 36, 'Acne': 37,
  'Urinary
        tract infection': 38,
                              'Psoriasis': 39,
                              'Impetigo': 40}}, inplace=True)
        X_test = tr[l1]
         y_test = tr[["prognosis"]]
          np.ravel(y_test)
          d, c = NaiveBayes(psymptoms)
          dl, cl = LogisticReg(psymptoms)
dd, cd = DecisionTree(psymptoms)
dk, ck = knn(psymptoms)
# print(d, c, '\n', '\n', dd, cd, '\n', dk, ck)
return d,dl,dd,dk
```

CHAPTER 4

APPLICATION OF GINGER

When talking about the healthcare industry, the possibilities for chatbots to grow are relatively high. Although the adoption rate is not widespread, however, the number of healthcare chatbots is increasing. As health services become patient-centric, offering personalized and satisfactory experience is of utmost priority for healthcare providers.

With a view to delivering the best services to patients, healthcare companies are embracing technology trends to stay at par with the latest trends and market needs. As customer services take a whole new approach, providers are keen to invest in automated tools to offer timely care services. In the digital age, patients expect faster responses from healthcare service providers. Delay in the same annoys customers, and the outcome is negative online reviews and damage to brand reputation. In a competitive market like healthcare, providers are in no mood to take a chance and so keeping patients happy is their ultimate goal.

Replacing human customer care executive is chatbots that are faster and intelligent to provide appropriate help to both healthcare professionals and patients. Does the use of healthcare technology improve patient experience? With automated assistance stepping in, the work of medical professionals has reduced manifold. They can now better concentrate on valuable things while chatbots assist patients with their medical queries, medication guidance, symptom checks, nutrition, and other matters. It is because of chatbots that patient visit to doctors for minor issues has gone down. For their health questions, patients are turning to chatbots first as they are quick. Soon, there will be a time when chatbots will be the first point of contact for medical help. Patients will go to doctors only when chatbots will fail to handle their issue.

Chapter 5

Implementing Ginger as a Web Service

To convert Ginger to a completely web-based service, we needed to implement it as a Website. So, in order to do this, first we made a logo of our product Ginger. The logo design at this stage kept simple, but with some meaning with the design and the colour templates.

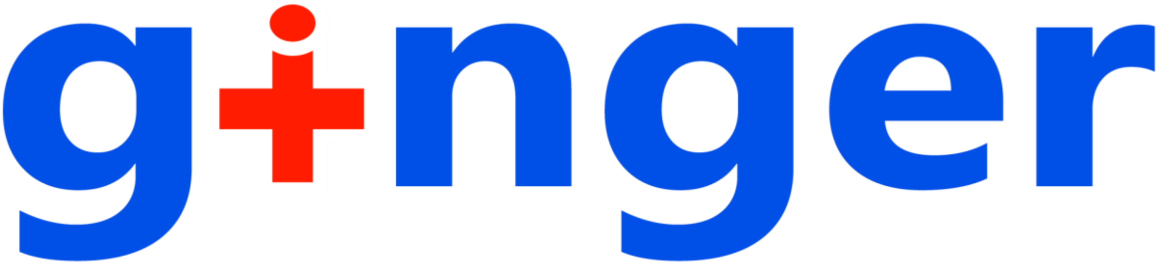
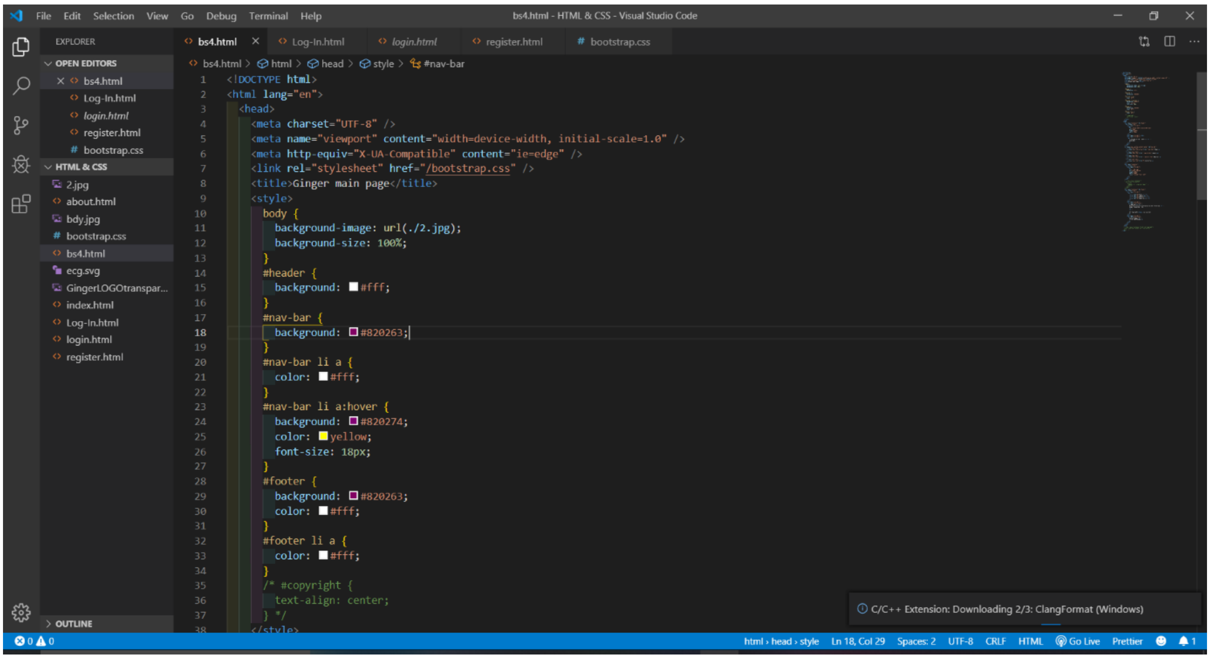


Fig 5.1: Logo of Ginger

After the design of the logo, now we needed to work on our main purpose that is creating a website. So, a website was created with HTML and CSS.

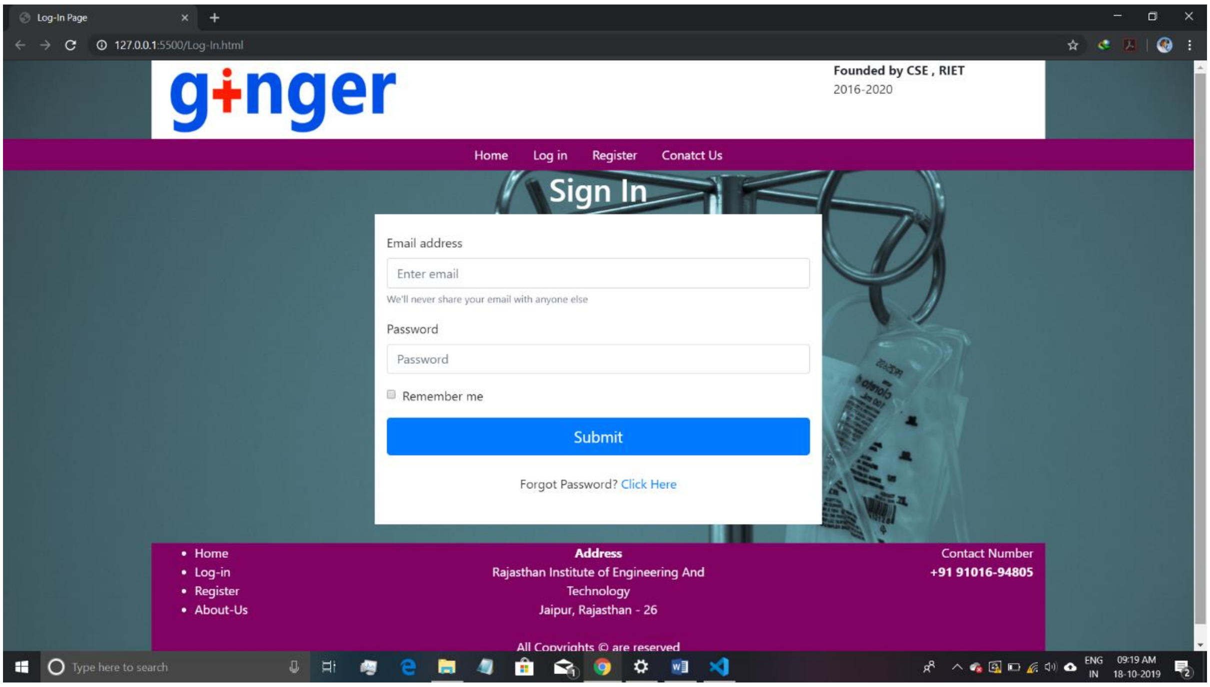
Some Snippets of the website are, as follows:



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <link rel="stylesheet" href="/bootstrap.css" />
    <title>Ginger main page</title>
    <style>
      body {
        background-image: url(./2.jpg);
        background-size: 100%;
      }
      #header {
        background: #fff;
      }
      #nav-bar {
        background: #820263;
      }
      #nav-bar li a {
        color: #fff;
      }
      #nav-bar li a:hover {
        background: #820274;
        color: yellow;
        font-size: 18px;
      }
      #footer {
        background: #820263;
        color: #fff;
      }
      #footer li a {
        color: #fff;
      }
      /* #copyright {
        text-align: center;
      } */
    </style>
```

C/C++ Extension: Downloading 2/3: ClangFormat (Windows)

Fig 5.2: Codes while creating the website



Founded by CSE , RIET
2016-2020

Sign In

Email address

Enter email

We'll never share your email with anyone else

Password

Password

□ Remember me

Submit

Forgot Password? Click Here

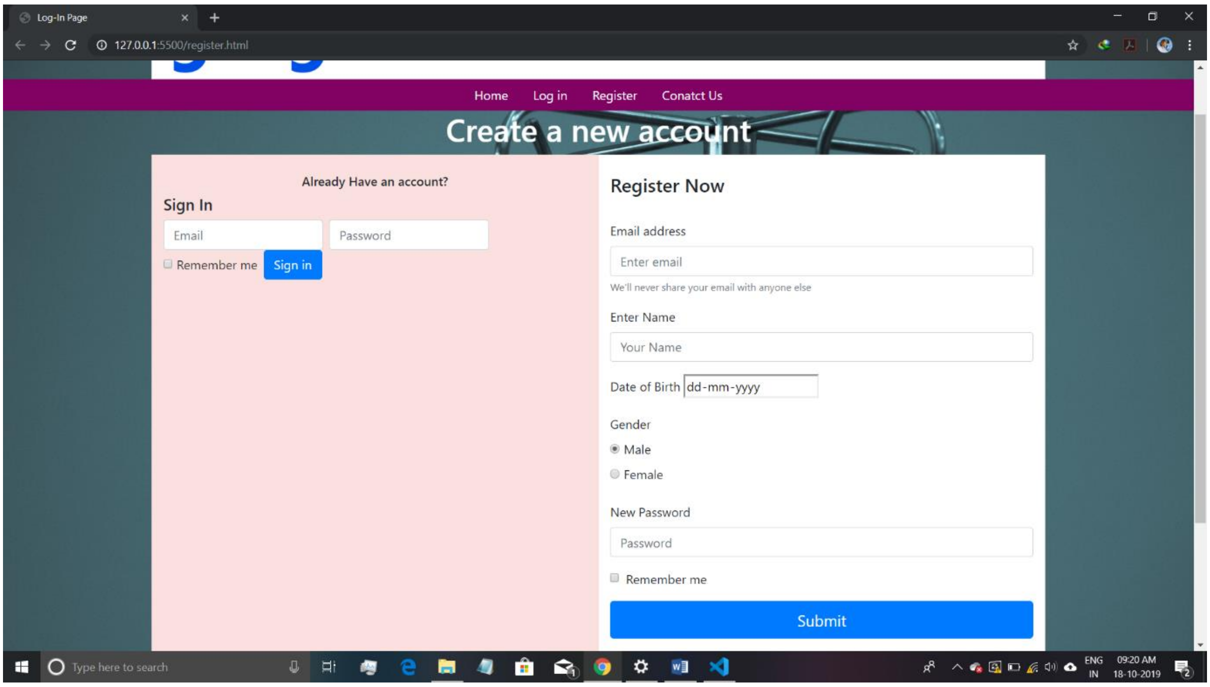
- Home
- Log-in
- Register
- About-Us

Address
Rajasthan Institute of Engineering And Technology
Jaipur, Rajasthan - 26

Contact Number
+91 91016-94805

All Copyrights © are reserved

Fig 5.3: Index page of our website with Log-In



Create a new account

Already Have an account?

Sign In

Email | Password

□ Remember me | Sign in

Register Now

Email address

Enter email

We'll never share your email with anyone else

Enter Name

Your Name

Date of Birth dd-mm-yyyy

Gender

● Male

○ Female

New Password

Password

□ Remember me

Submit

Fig 5.4: New user registration page

CONCLUSION

By reviewing the literature we come to know that this system giving the accurate result. As we are using large dataset which will ensures the better performance compared as earlier. Thus we build up a system which is useful for medical institute or hospitals to help the users to freely ask medical dosage related queries. System gets output for medicine API and display all medicine names. We are using NLP because we want to a computer to communicate with users in their terms. So by using KNN Algorithm, Logistic Regression Algorithm, Linear Regression Algorithm, Naive Bayes Algorithm algorithm and disease symptoms system can predict disease. User can get related answer displayed r on Webpage. and refer this answer for analysis.

REFERENCES

1. https://docs.python.org/3/library/.
2. https://www.springpeople.com/blog/future-scope-of-artificial-intelligence.
3. https://medium.mybridge.co/34-amazing-python-open-source-libraries-for-the-past-year-v-2019-93d6ee11aceb?gi=71baa3d09338.
4. https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/
5. https://realpython.com/jupyter-notebook-introduction/
6. https://becominghuman.ai/introduction-to-artificial-intelligence-5fba0148ec99