



INTERNSHIP

FINAL

REPORT

[WEB APPLICATION DEVELOPMENT]

“DISEASE DIAGNOSIS USING CHATBOT”

MOHAMMAD IMTIAZ AHMED

[1325-23-862-155]



Internship Report On

“DISEASE DIAGNOSIS USING CHATBOT”

Submitted to

AURORA'S PG COLLEGE (AURI)

RAMANTHAPUR, HYDERABAD

Accredited by NAAC with A+ Grade

Ganesh Nagar, Ramanthapur-500013

In partial fulfilment of the requirement for the award of Degree of

POST GRADUATION

In

M C A

By

MOHAMMAD IMTIAZ AHMED

[1325-23-862-155]

Under the guidance

of

Mr. A AKASH

(Associate Professor)

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

[2023 -2025]

INTERNSHIP COMPLETION CERTIFICATE

Reg. No: MANAC/IN/24-25/INT24101

This is to certify that **MD. IMTIAZ AHMED** bearing the ROLL. No: **1325-23-862-155** from AURORA'S POST GRADUATE COLLEGE, RAMANTHAPUR, student of MCA, has successfully completed the project titled "**DISEASE DIAGNOSIS CHATBOT**" as part of the internship in our organization.

He/ She has done the project during the period **17-09-2024 to 30-10-2024**, under the guidance and supervision of Mr. A. Akash, from Manac Infotech Pvt Ltd, Hyderabad.

He/ She has completed the assigned project well within the time frame. He/ She is sincere hardworking and his/her conduct during the period is commendable.

We wish all the best in his/her future endeavor

For MANAC INFOTECH PVT LTD

M KONDAL REDDY
(Manager-HR)

An IIM Alumnus Enterprise

ACKNOWLEDGMENT

I would like to extend my heartfelt gratitude to my internship supervisor, **Mr. A Akash**, whose invaluable guidance and mentorship were instrumental throughout the course of this project. Their expertise and encouragement were crucial in navigating challenges and enhancing my learning experience, and I am truly grateful for their time and effort in supporting my growth.

I also wish to express my deep appreciation to my faculty mentor, **Mr. RAMANJANELYULU**, for his unwavering academic support, insightful feedback, and encouragement. His guidance helped me bridge the gap between theory and practical application, and his belief in my potential

motivated me to excel throughout this internship.

Furthermore, I am immensely grateful to my colleagues at **MANAC INFOTEC PVT LTD**. Their constant support, collaboration, and camaraderie created a conducive environment for teamwork and shared learning, which greatly contributed to the successful completion of this project. The experience of working alongside such dedicated and talented professionals has been enriching and memorable.

I also wish to thank my colleagues at MANAC INFOTEC PVT LTD for their constant support and collaboration, which greatly contributed to the successful completion of this project.

.

MOHAMMAD IMTIAZ AHMED

[1325-23-862-155]

DECLARATION

I, MD . IMTIAZ AHMED, hereby declare that this Internship Report entitled "DISEASE DIAGNOSIS USING CHATBOT" is a result of the work carried out at MANAC INFOTECH PVT LTD and submitted by me to the Department of MASTER of Computer Applications, Aurora's PG College (MCA), Ramanthapur. This is a bonafide work done by me and it is not submitted to any other University or Institution for the award of a degree/diploma/certificate or published at any time before the submission of this report.

Name and Address of the Student

Signature of the Student

TABLE OF CONTENT

TOPICS	Page No's
Certificates	
Acknowledgement	
Abstract	
CHAPTER-1: INTRODUCTION	1-2
CHAPTER-2: ORGANIZATION PROFILE	3
CHAPTER-3: PROJECT DESCRIPTION SYSTEM REQUIREMENTS	
3.1 Functional Requirement	4
3.2 Non-Functional Requirements	4
3.3 SOFTWARE ENVIRONMENT	
3.3.1 Python	5-12
CHAPTER-4: SYSTEM ANALYSIS	
4.1 Existing System	13
4.2 Proposed System	14
4.3 Source Code	15-20
4.4 System Testing	20-22
4.5 Test Cases	23-25
CHAPTER-5: LEARNING OUTCOMES	
5.1 Output Screens	26-33
CHAPTER-6: CONCLUSION	
6.1 Conclusion	34
6.2 Future Scope	34-35
CHAPTER-7: REFERENCES	36

DISEASE DIAGNOSIS USING CHATBOT

ABSTRACT

The project on "Disease Diagnosis using Chatbot" introduces an innovative healthcare solution by integrating chatbot technology for preliminary disease diagnosis. The chatbot serves as an interactive and user-friendly interface, guiding users through a series of questions to gather symptoms and provide initial insights into potential health conditions. This project aims to enhance accessibility to healthcare information and assist users in making informed decisions about seeking medical attention. Disease diagnosis is a critical aspect of healthcare, often requiring timely and accurate assessments to facilitate appropriate treatment and management. With the advancement of artificial intelligence (AI) and natural language processing (NLP) technologies, chatbots have emerged as promising tools for facilitating disease diagnosis and patient interaction. In this study, we propose a novel approach for disease diagnosis using a chatbot system equipped with AI-driven capabilities. The chatbot utilizes NLP techniques to engage in natural language conversations with users, gathering relevant information about symptoms and medical history to assist in the diagnostic process. The system employs machine learning algorithms to analyze user input and provide personalized recommendations or referrals based on identified patterns and medical knowledge. Experimental results demonstrate the effectiveness of the proposed chatbot system in accurately diagnosing a variety of diseases, showcasing its potential to enhance healthcare delivery and accessibility.

CHAPTER-1

INTRODUCTION

Access to timely healthcare information is crucial for early disease detection and management. Traditional methods of self-diagnosis often lack personalized guidance. This project leverages chatbot technology to create an interactive platform where users can describe their symptoms, receive relevant health information, and be directed towards appropriate medical resources. The process of disease diagnosis is often complex and multifaceted, requiring healthcare professionals to gather comprehensive information from patients to make informed decisions. However, factors such as limited access to healthcare services, time constraints, and patient hesitancy can pose challenges to timely diagnosis and treatment. Chat bots, powered by AI and NLP technologies, offer a promising solution by providing a convenient and accessible platform for interacting with users and assisting in the diagnostic process. Chat bots designed for disease diagnosis leverage the capabilities of NLP to engage in natural language conversations with users, mimicking human-like interactions to gather relevant information about symptoms, medical history, and other pertinent factors. These chat bots utilize machine learning algorithms to analyze user input and infer potential diagnoses based on established medical knowledge and patterns identified from training data. By employing a combination of supervised and unsupervised learning techniques, chat bots can adapt and improve their diagnostic accuracy over time, enhancing their utility in real-world healthcare settings.

In this study, we propose a novel approach for disease diagnosis using a chatbot system equipped with AI-driven capabilities. The chatbot engages users in natural language conversations, guiding them through a series of questions to gather information relevant to their symptoms and medical history. The system utilizes advanced machine learning algorithms to analyze user input and generate personalized recommendations or referrals based on identified patterns and medical expertise.

CHAPTER-2

ORGANIZATION PROFILE

MANAC Infotech Pvt. Ltd. (MIPL) is a progressive and innovative IT company offering a spectrum of services including software development, staffing, recruitment and training. An IIM Alumnus Company started in 1998, erstwhile training partner of TCS iON. Tata Consultancy Services Limited (TCS Ltd.) is a leading IT solutions company & one of the world's leading Information Technology consulting, services and business process outsourcing organizations and a part of the Tata Group, India's best-known business conglomerate. MANAC has built a suite of services around its core offering of training.

In the training arena MANAC has an unenviable track record of grooming several hundreds of students, building their skills and launching their careers. Alumni of MANAC are in senior IT positions across the globe. MANAC also offers a host of training programs tailor made for IT companies. The development arm of MIPL focuses on providing solutions to the Education & Training sector including assessments, analytics, online course delivery and ERP for training companies. Along the way, MANAC has developed a strong expertise in JAVA, Android and open source technologies. Relations built with HR managers over the years have helped MANAC provide specialized services in the area of staffing and fresher's recruitment for IT & ITES companies.

CHAPTER-3

PROJECT DESCRIPTION

SYSTEM REQUIREMENTS

3.1 FUNCTIONAL REQUIREMENTS

User

The user module in a disease diagnosis system using chatbots serves as the interface through which individuals interact with the chatbot to seek medical advice and receive diagnostic assistance. The user module typically begins with a registration or login process, where users create accounts or sign in to access the chatbot's services.

3.2 NON-FUNCTIONAL REQUIREMENTS

HARDWARE REQUIRMENTS:

- System : i3 or above.
- Ram : 4 GB.
- Hard Disk : 40 GB

SOFTWARE REQUIRMENTS:

- Operating system : Windows8 or Above.
- Coding Language : python

3.3 SOFTWARE ENVIRONMENT

3.3.1 PYTHON

What is Python :

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber... etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like Opencv, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia

Advantages of Python Over Other Languages :

1. Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don't have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.

3. Python is for Everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and **machine learning**, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

What is Machine Learning : -

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of *building models of data*.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models *tunable parameters* that can be

adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain. Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven't surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, "to make decisions, based on data, with efficiency and scale".

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programming logic, in the problems that cannot be programmed inherently. The fact is that we can't do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

(b) Types of Machine Learning

- **Supervised Learning** – This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.

- **Unsupervised Learning** – This involves using unlabelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.
- **Semi-supervised Learning** – This involves using unlabelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.
- **Reinforcement Learning** – This involves learning optimal actions through trial and error. So the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

Advantages of Machine learning :-

1. Easily identifies trends and patterns -

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

2. No human intervention needed (automation)

With ML, you don't need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

3. Continuous Improvement

As **ML algorithms** gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

4. Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

5. Wide Applications

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

Modules Used in Project :-

Tensorflow

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

Numpy

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

Python

Download the Correct version into the system

Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: <https://www.python.org>



Now, check for the latest and the correct version for your operating system.








Step 2: Click on the Download Tab.



Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
Python 3.7.4	July 8, 2019	 Download	Release Notes
Python 3.6.9	July 2, 2019	 Download	Release Notes
Python 3.7.3	March 25, 2019	 Download	Release Notes
Python 3.4.10	March 18, 2019	 Download	Release Notes
Python 3.5.7	March 18, 2019	 Download	Release Notes
Python 2.7.16	March 4, 2019	 Download	Release Notes
Python 3.7.2	Dec. 24, 2018	 Download	Release Notes

Step 4: Scroll down the page until you find the Files option.

Step 5: Here you see a different version of python along with the operating system.

Files					
Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		68111671e5b2db4ae77b9ab013f09be	23017663	SG
XZ compressed source tarball	Source release		d33e4aae6097051c2nca45ee3604803	17131432	SG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6428b4fa7583daf1a42c8a1cee08e6	34898435	SG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	5dd605c38217a45773bf5e4a936b243f	28082845	SG
Windows help file	Windows		d63999573a2c56b2ac58cade6b477cd2	8131761	SG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	9b09c3cfd8ec0b9abe8318aa0725a2	7504391	SG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	a702b4b0ad76d4bcb3543a583e563400	26680368	SG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	28c81c6088bd73ae8e53a3bd351b4bd2	1362904	SG
Windows x86 embeddable zip file	Windows		9fab3bd19841879fda94132574139d8	6741628	SG
Windows x86 executable installer	Windows		33cc02942a544efac3b6451478394789	25663848	SG
Windows x86 web-based installer	Windows		1b670cfa5cd317d83c30983ea371d87c	1324608	SG

- To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.

- To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

CHAPTER-4

SYSTEM ANALYSIS

4.1 EXISTING SYSTEM

Traditional online symptom-checking platforms may lack interactivity and often provide generic information without considering the individual context. Users may find it challenging to navigate through complex medical terms, leading to potential misinterpretation of symptoms.

DISADVANTAGES

- Chatbots rely on pre-programmed algorithms and databases for diagnosing diseases. However, they might lack the capability to provide accurate diagnoses for complex or rare conditions, leading to misdiagnosis.
- While chatbots can provide quick responses, they often lack the human touch and empathy that is essential in healthcare interactions. Patients may feel more comfortable discussing sensitive health issues with human healthcare providers.
- Patients might overly rely on chatbots for medical advice, potentially neglecting the importance of consulting with qualified healthcare professionals. This could delay appropriate medical interventions or lead to self-medication, exacerbating health issues.

4.2 PROPOSED SYSTEM

The proposed project introduces a user-friendly and interactive chatbot for disease diagnosis, addressing the limitations of traditional symptom-checking methods. The chatbot's conversational nature, database integration, and personalized recommendations enhance the user experience and promote informed healthcare decision-making.

ADVANTAGES

- Chatbots provide round-the-clock availability, allowing users to seek medical advice and diagnosis at any time, regardless of location or healthcare provider availability

- .Chatbots can offer educational materials and resources to users, empowering them to make informed decisions about their health and well-being and promoting health literacy.
- Chatbots can integrate with existing healthcare systems, electronic health records (EHRs), and telemedicine platforms, facilitating streamlined communication and information exchange between patients and healthcare providers.
- Chatbots can potentially reduce healthcare costs by automating routine tasks, minimizing the need for in-person consultations for minor ailments, and optimizing resource allocation within healthcare systems.
- Through machine learning and artificial intelligence algorithms, chatbots can continuously learn from user interactions and feedback, improving their accuracy, effectiveness, and responsiveness over time.

SOURCE CODE

MANAGE.PY

```
#!/usr/bin/env python

import os

import sys


if __name__ == '__main__':

    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'Disease.settings')

    try:

        from django.core.management import execute_from_command_line

    except ImportError as exc:

        raise ImportError(

            "Couldn't import Django. Are you sure it's installed and "
```

"available on your PYTHONPATH environment variable? Did you "

"forget to activate a virtual environment?"

) from exc

execute_from_command_line(sys.argv)

SETTINGS.PY

[12:13, 22/05/2024] Revathi Mam.Manac: ""

Django settings for DiseaseDiagnosis project.

Generated by 'django-admin startproject' using Django 2.1.7.

For more information on this file, see

<https://docs.djangoproject.com/en/2.1/topics/settings/>

For the full list of settings and their values, see

<https://docs.djangoproject.com/en/2.1/ref/settings/>

""

import os

Build paths inside the project like this: os.path.join(BASE_DIR, ...)

BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(_file_)))

```
# Quick-start development settings - unsuitable for production
```

```
# See https://docs.djangoproject.com/en/2.1/howto/deployment/checklist/
```

```
# SECURITY WARNING: keep the secret key used in production secret!
```

```
SECRET_KEY = '^05^_s8m%21bqsf$R)vyqyp4m)p@b3i=ydo=cnjfnqko-oam&1'
```

```
# SECURITY WARNING: don't run with debug turned on in production!
```

```
DEBUG = True
```

```
ALLOWED_HOSTS = []
```

```
# Application definition
```

```
INSTALLED_APPS = [
```

```
    'django.contrib.admin',
```

```
    'django.contrib.auth',
```

```
    'django.contrib.contenttypes',
```

```
    'django.contrib.sessions',
```

```
    'django.contrib.messages',
```

```
    'django.contrib.staticfiles',
```

```
    'DiseasePrediction'
```

```
]
```

```
MIDDLEWARE = [  
    'django.middleware.security.SecurityMiddleware',  
    'django.contrib.sessions.middleware.SessionMiddleware',  
    'django.middleware.common.CommonMiddleware',  
    'django.middleware.csrf.CsrfViewMiddleware',  
    'django.contrib.auth.middleware.AuthenticationMiddleware',  
    'django.contrib.messages.middleware.MessageMiddleware',  
    'django.middleware.clickjacking.XFrameOptionsMiddleware',  
]
```

```
ROOT_URLCONF = 'Disease.urls'
```

```
TEMPLATES = [  
    {  
        'BACKEND': 'django.template.backends.django.DjangoTemplates',  
        'DIRS': [  
            os.path.join('D:/Manoj_laptop_backup/sstech/DiseasePrediction/DiseasePrediction',  
'templates'),  
        ],  
        'APP_DIRS': True,
```



```
'OPTIONS': {  
  
    'context_processors': [  
  
        'django.template.context_processors.debug',  
  
        'django.template.context_processors.request',  
  
        'django.contrib.auth.context_processors.auth',  
  
        'django.contrib.messages.context_processors.messages',  
  
    ],  
  
    },  
  
    },  
  
]
```

```
WSGI_APPLICATION = 'Disease.wsgi.application'
```

```
# Database
```

```
# https://docs.djangoproject.com/en/2.1/ref/settings/#databases
```

```
DATABASES = {
```

```
    'default': {  
  
        'ENGINE': 'django.db.backends.mysql',  
  
        'NAME': 'DiseasePrediction',  
  
        'HOST': '127.0.0.1',  
  
        'PORT': '3306',
```

```
'USER': 'root',

'PASSWORD': 'root',

'OPTIONS': {

    'autocommit': True,

},

}

}

# Password validation

# https://docs.djangoproject.com/en/2.1/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [

    {

        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',

    },

    {

        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',

    },

    {

        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',

    },

    {

        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
```

```
    },  
]  
  
# Internationalization  
  
# https://docs.djangoproject.com/en/2.1/topics/i18n/  
  
LANGUAGE_CODE = 'en-us'  
  
TIME_ZONE = 'UTC'  
  
USE_I18N = True  
  
USE_L10N = True  
  
USE_TZ = True  
  
# Static files (CSS, JavaScript, Images)  
  
# https://docs.djangoproject.com/en/2.1/howto/static-files/  
  
STATIC_URL = '/static/'  
  
REGISTER.HTML  
  
{% load static %}  
  
<html>  
  
<head>  
  
<title>Disease Diagnosis</title>  
  
<meta http-equiv="content-type" content="text/html; charset=utf-8" />  
  
<link href="{% static 'default.css' %}" rel="stylesheet" type="text/css" media="screen" />
```

```
<script LANGUAGE="Javascript" >
```

```
function validate(){
```

```
    var x=document.forms["f1"]["tf1"].value;
```

```
    var y=document.forms["f1"]["tf4"].value;
```

```
    var c=document.forms["f1"]["tf5"].value;
```

```
    var e=document.forms["f1"]["tf6"].value;
```

```
    var a=document.forms["f1"]["tf7"].value;
```

```
    var b=document.forms["f1"]["tf8"].value;
```

```
    var d=document.forms["f1"]["tf9"].value;
```

```
    if(x == null || x==""){
```

```
        window.alert("Patient name must be enter");
```

```
        document.f1.tf1.focus();
```

```
        return false;
```

```
    }
```

```
    if(y == null || y==""){
```

```
        window.alert("Height must be enter");
```

```
        document.f1.tf4.focus();
```

```
        return false;
```

```
    }
```

```
    if(c == null || c==""){
```

```
        window.alert("Weight must be enter");
```

```

        document.f1.tf5.focus();

        return false;
    }

    if(isNaN(c)){

        window.alert("Please enter valid weight");

        document.f1.tf5.focus();

        return false;
    }

    if(e == null || e==""){

        window.alert("Disease history must be enter");

        document.f1.tf6.focus();

        return false;
    }

    if(a == null || a==""){

        window.alert("Email ID must be enter");

        document.f1.tf7.focus();

        return false;
    }

    var filter = /^[a-zA-Z0-9_\.|-]+\@(gmail+\.)+(com)+$/;

    if (!filter.test(a)) {

        window.alert('enter a valid email address');
    }

```

```
        document.f1.tf7.focus();

        return false;    }

    if(b == null || b==""){

        window.alert("password must be enter");

        document.f1.tf8.focus();

        return false;

    }

    if(d == null || d==""){

        window.alert("Contact No must be enter");

        document.f1.tf9.focus();

        return false;

    }

    if(isNaN(d)){

        window.alert("Please enter valid contact no");

        document.f1.tf9.focus();

        return false;

    }

    return true;

}

</script>

</head>
```

```
<body>
```

```
<div id="wrapper">
```

```
<div id="header">
```

```
<div id="logo">
```

```
<h1><font color="orange" size="5">Disease Diagnosis using Chatbot</font></h1>
```

```
<marquee><font color="pink" size="4">Disease Diagnosis</font></marquee>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div id="menu">
```

```
<ul>
```

```
<li><a href="{ % url 'index' % }">Home</a></li>
```

```
<li><a href="{ % url 'User' % }">User</a></li>
```

```
<li><a href="{ % url 'Register' % }">Register Here</a></li>
```

```
</ul>
```

```
</div>
```

```
<div class="entry">
```

```
<br/><br/><br/>
```

```
<font size="" color="white"><center>{ { data } }</center></font>
```


<center>New User Signup Screen</center>

<form name="f1" method="post" action={% url 'Signup' %} OnSubmit="return
validate()">

{% csrf_token %}

<TABLE align=center width="35%" class="notepad">

<TR><TH align="left">Patient Name

<TD> <Input type=text name="tf1" value=" class="form-control">

<div id='nameid'></div>

</TD>

</TR>

<TR><TH align="left">Age

<TD> <select name="tf2">

{{ data1|safe }}

</select>

<div id='nameid'></div>

</TD>

</TR>

<TR><TH align="left">Gender

<TD> <select name="tf3">


```
<option value="Male">Male</option>
```

```
<option value="Female">Female</option>
```

```
</select></TD>
```

```
</TR>
```

```
<TR><TH align="left"><font size="" color="white">Height
```

```
<TD>&nbsp;&nbsp;<Input type='text' name="tf4" value=" class="form-  
control">
```

```
</TR>
```

```
<TR><TH align="left"><font size="" color="white">Weight
```

```
<TD>&nbsp;&nbsp;<Input type='text' name="tf5" value=" class="form-  
control">
```

```
<TR><TH align="left"><font size="" color="white">Disease&nbsp;History
```

```
<TD>&nbsp;&nbsp;<Input type='text' size="35" name="tf6" value="  
class="form-control">
```

```
</TR>
```

```
<TR><TH align="left"><font size="" color="white">Email&nbsp;ID
```

```
<TD>&nbsp;&nbsp;<Input type='text' size="35" name="tf7" value="  
class="form-control">
```

```
</TR>
```

```
<TR><TH align="left"><font size="" color="white">Password
```

```
<TD>&nbsp;&nbsp;<Input type='password' size="25" name="tf8" value="  
class="form-control">
```

```

</TR>

<TR><TH align="left"><font size="" color="white">Contact&nbsp;No

<TD>&nbsp;&nbsp;<Input  type='text'  size="15"  name="tf9"  value="
class="form-control">

</TR>

<TR>

<TD></TD>

<TD>

<input type="submit" value="Register">

</TABLE>

</form>

</div>

</body>

</html>

```

SYSTEM TESTING:

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

- Valid Input : identified classes of valid input must be accepted.
- Invalid Input : identified classes of invalid input must be rejected.
- Functions : identified functions must be exercised.
- Output : identified classes of application outputs must be exercised.
- Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

TEST CASES:

Test case1:

Test case for Login form:

FUNCTION:	LOGIN
EXPECTED RESULTS:	Should Validate the user and check his existence in database
ACTUAL RESULTS:	Validate the user and checking the user against the database
LOW PRIORITY	No
HIGH PRIORITY	Yes

Test case2:

Test case for User Registration form:

FUNCTION	REMOTE USER REGISTRATION
EXPECTED RESULTS:	Should check if all the fields are filled by the user and saving the user to database.
ACTUAL RESULTS:	Checking whether all the fields are filled by user or not through validations and saving user.
LOW PRIORITY	No
HIGH PRIORITY	Yes

Test case3:

Test case for Change Password:

When the old password does not match with the new password , then this results in displaying an error message as “ OLD PASSWORD DOES NOT MATCH WITH THE NEW PASSWORD”.

Test case 4:

Test case for Forget Password:

When a user forgets his password he is asked to enter Login name, ZIP code, Mobile number. If these are matched with the already stored ones then user will get his Original password.

Module	Functionality	Test Case	Expected Results	Actual Results	Result	Priority
User	Login Usecase	1. Navigate To Wwww.Sample.Co m 2. Click On Submit Button Without Entering Username and Password	A Validation Should Be As Below “Please Enter Valid Username & Password”	A Validation Has Been Populated As Expected	Pass	High

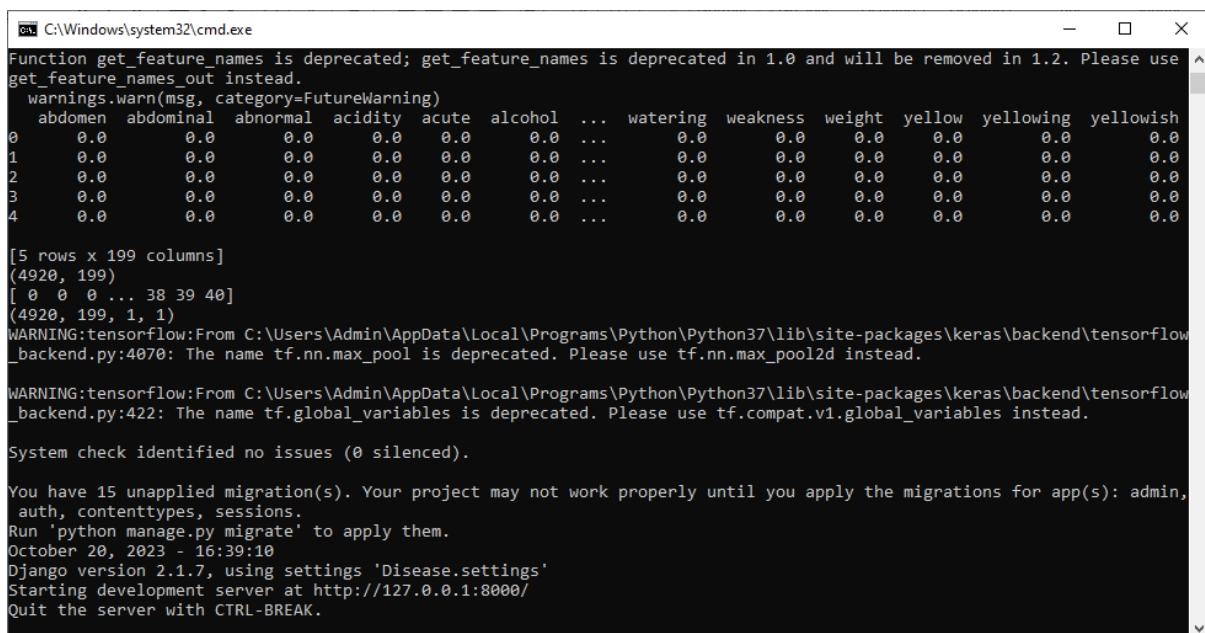
		<p>1. aNavigate To Www.Sample.Co m</p> <p>2. Click On Submit Button With Out Filling Password And With Valid Username</p>	<p>A Validation Should Be As Below “Please Enter Valid Password Or Password Field Can Not Be Empty “</p>	<p>A Validation Is Shown As Expected</p>	Pass	High
		<p>1. NNavigate To Www.Sample.Co m</p> <p>2. Enter Both Username And Password Wrong And Hit Enter</p>	<p>A Validation Shown As Below “The Username Entered Is Wrong”</p>	<p>A Validation Is Shown As Expected</p>	Pass	High
		<p>1. Navigate To Www.Sample.Co m</p> <p>2. Enter Validate Username And Password And Click On Submit</p>	<p>Validate Username And Password In DataBase And Once If They Correct Then Show The Main Page</p>	<p>Main Page/ Home Page Has Been Displayed</p>	Pass	High

CHAPTER-5

LEARNING OUTCOMES

SCREEN SHOTS

To run project copy content from DB.txt file and then paste in MYSQL console to create database and then double click on 'run.bat' file to start python server and get below page



```
C:\Windows\system32\cmd.exe
Function get_feature_names is deprecated; get_feature_names is deprecated in 1.0 and will be removed in 1.2. Please use
get_feature_names_out instead.
warnings.warn(msg, category=FutureWarning)
  abdomen  abdominal  abnormal  acidity  acute  alcohol  ...  watering  weakness  weight  yellow  yellowing  yellowish
0      0.0      0.0      0.0      0.0      0.0      0.0  ...      0.0      0.0      0.0      0.0      0.0
1      0.0      0.0      0.0      0.0      0.0      0.0  ...      0.0      0.0      0.0      0.0      0.0
2      0.0      0.0      0.0      0.0      0.0      0.0  ...      0.0      0.0      0.0      0.0      0.0
3      0.0      0.0      0.0      0.0      0.0      0.0  ...      0.0      0.0      0.0      0.0      0.0
4      0.0      0.0      0.0      0.0      0.0      0.0  ...      0.0      0.0      0.0      0.0      0.0

[5 rows x 199 columns]
(4920, 199)
[ 0  0  0 ... 38 39 40]
(4920, 199, 1, 1)
WARNING:tensorflow:From C:\Users\Admin\AppData\Local\Programs\Python\Python37\lib\site-packages\keras\backend\tensorflow
_backend.py:4070: The name tf.nn.max_pool is deprecated. Please use tf.nn.max_pool2d instead.

WARNING:tensorflow:From C:\Users\Admin\AppData\Local\Programs\Python\Python37\lib\site-packages\keras\backend\tensorflow
_backend.py:422: The name tf.global_variables is deprecated. Please use tf.compat.v1.global_variables instead.

System check identified no issues (0 silenced).

You have 15 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin,
auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
October 20, 2023 - 16:39:10
Django version 2.1.7, using settings 'Disease.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

FIG-1 In above screen python server started and now open browser and entre URL as <http://127.0.0.1:8000/index.html> and then press enter key to get below page

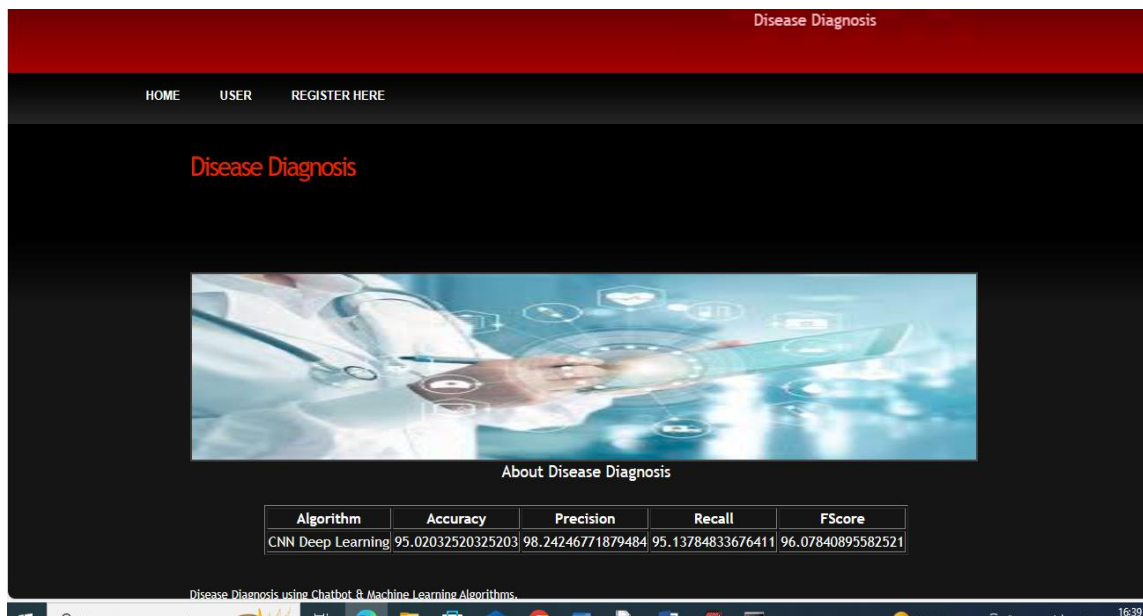


FIG-2 In above screen we can see application home page and then in table we can see CNN algorithm disease prediction accuracy and now click on ‘Register Here’ link to sign up with the application

New User Signup Screen

Patient Name

Age

Gender

Height

Weight

Disease History

Email ID

Password

Contact No

FIG-3 In above screen user is entering signup detail and give valid MAIL ID so you can receive mails and press button to get below page

The screenshot shows a web application titled "Disease Diagnosis using Chatbot" with a red header. Below the header is a navigation bar with links for HOME, USER, and REGISTER HERE. A search bar is located to the right of the navigation bar. The main content area is titled "New User Signup Screen" and contains a form with the following fields: Patient Name (text input), Age (dropdown menu), Gender (dropdown menu with "Male" selected), Height (text input), Weight (text input), Disease History (text input), Email ID (text input), Password (text input with a toggle for visibility), and Contact No (text input). A blue notification message "Signup Process Completed" is displayed above the form.

FIG-4 In above screen in blue colour text we can see sign up completed and now click on ‘User’ link to login as user

The screenshot shows the same web application as FIG-4, but the main content area is titled "User Login Screen". It contains a form with two fields: Email ID (text input with the value "kaleem.mmd@gmail.com") and Password (text input with masked characters "*****" and a toggle for visibility). A "Login" button is located below the password field. The navigation bar and search bar remain the same.

FIG-5 In above screen user is login and after login will get below page

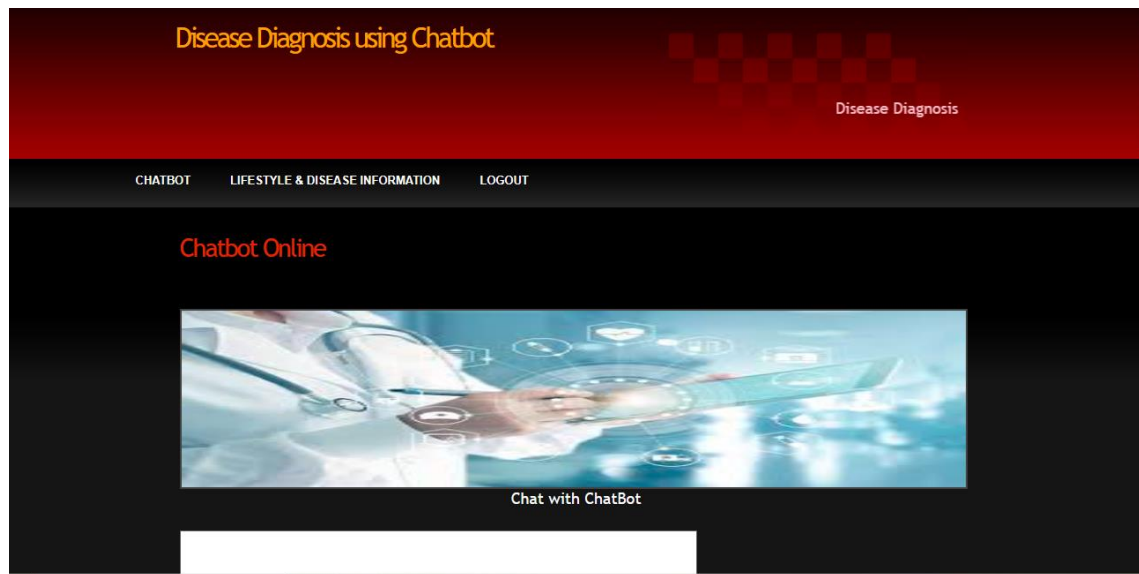


FIG-6 In above screen click on 'Chatbot' link to get below page

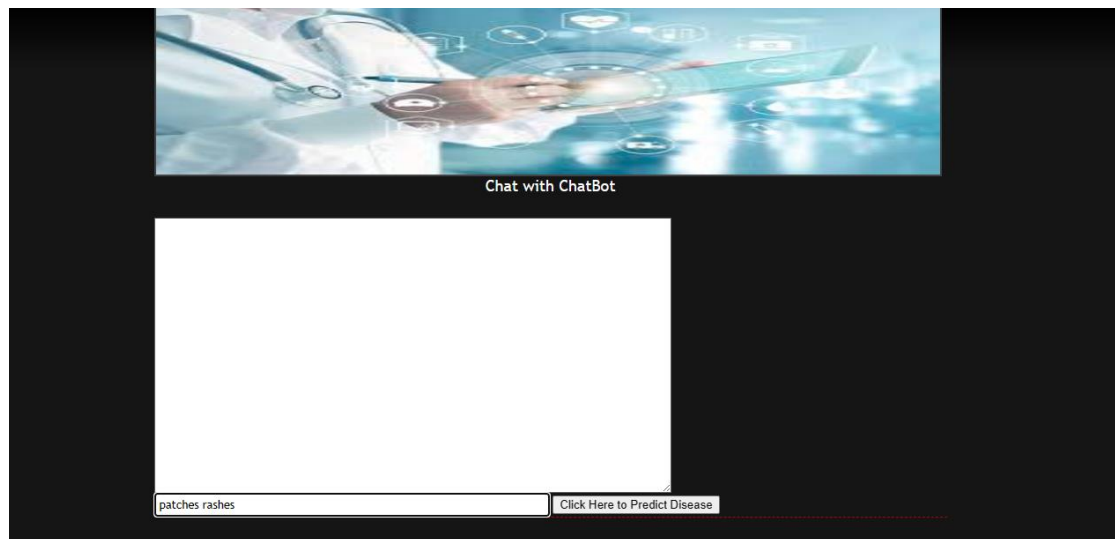


FIG-7 In above Chatbot page just type some symptoms and in above page I gave symptoms as 'patches rashes' and then press button to get reply from Chatbot like below screen

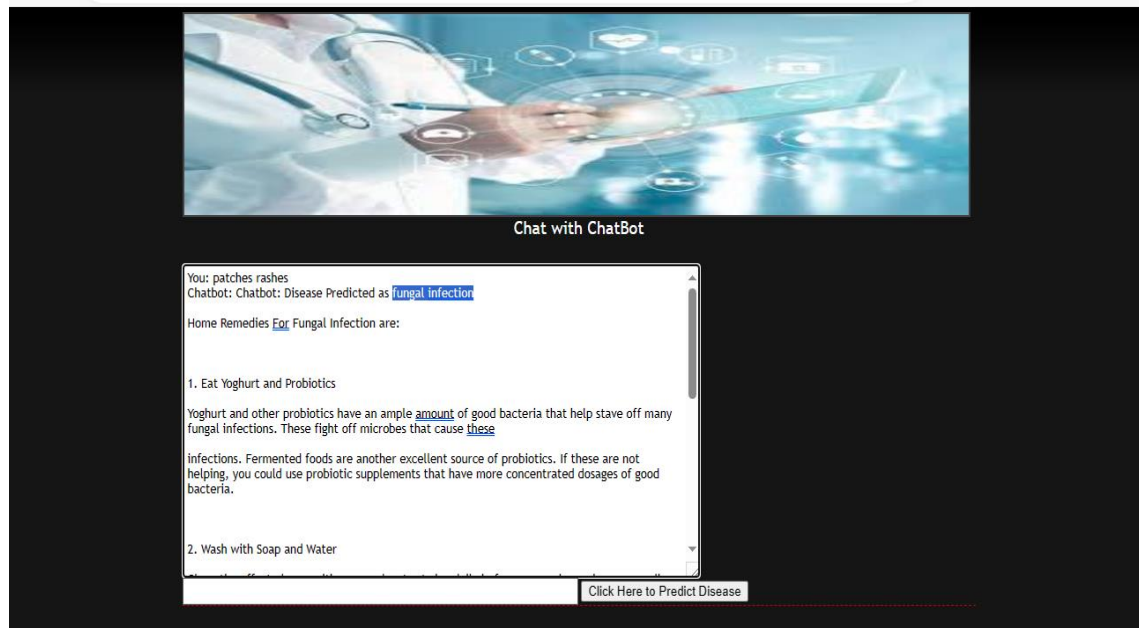


FIG 8 :

sease predicted as 'Fungal Infection' and then in below lines we can see home remedies along with diet details and scroll down above page to view complete details

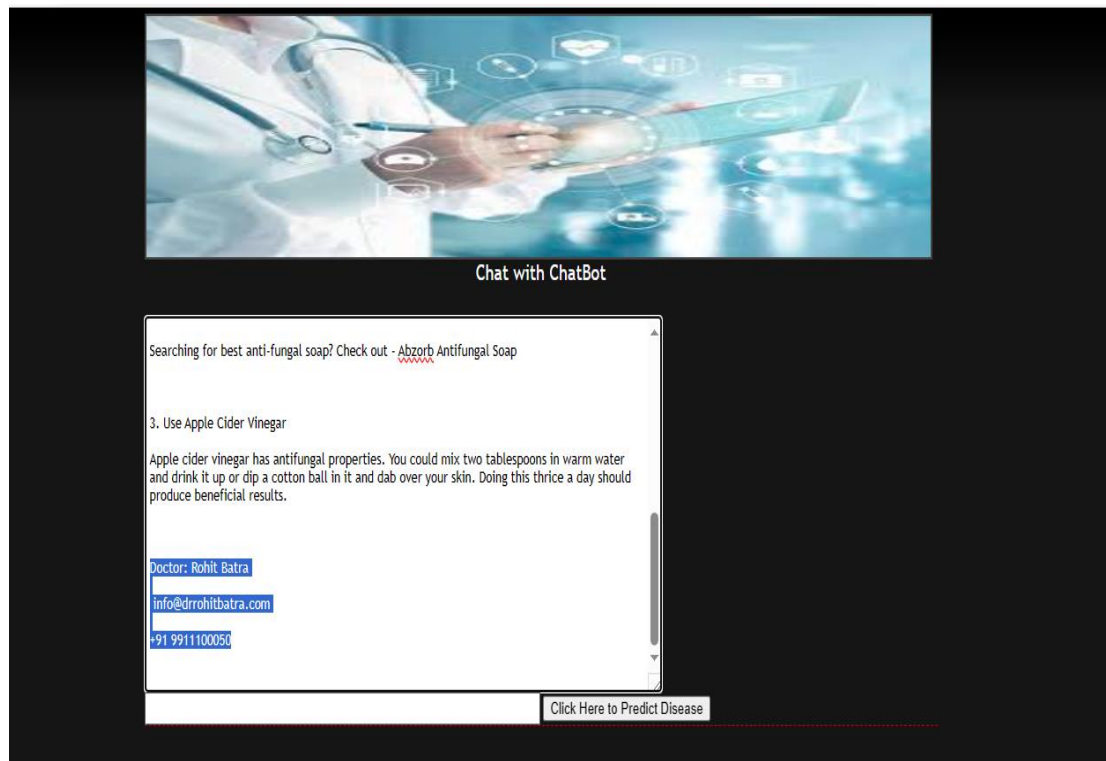


FIG-9 In above screen we can see doctor details and then same information will be sent to mail also like below screen

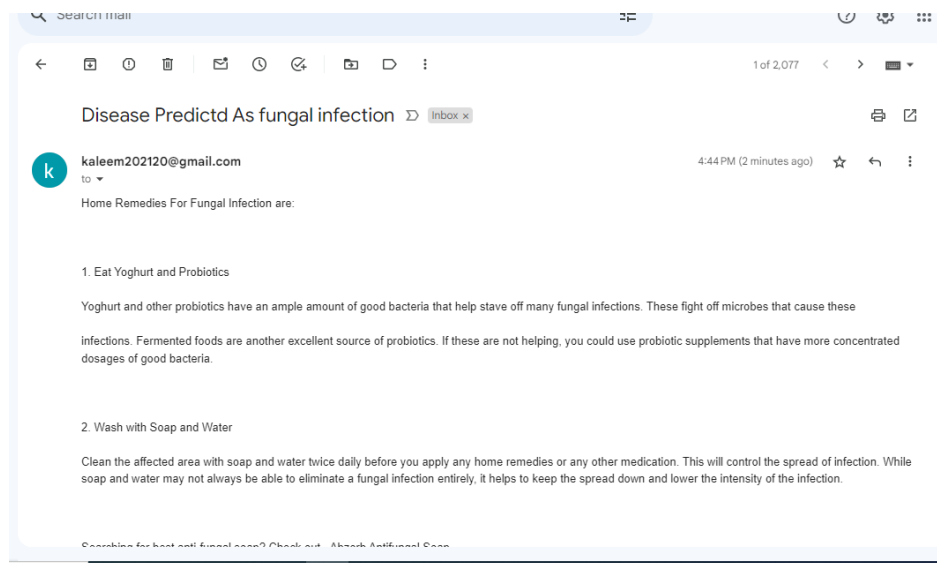


FIG-10 In above email we can see disease details with diet and remedies and similarly you can search for any symptoms and below is another example

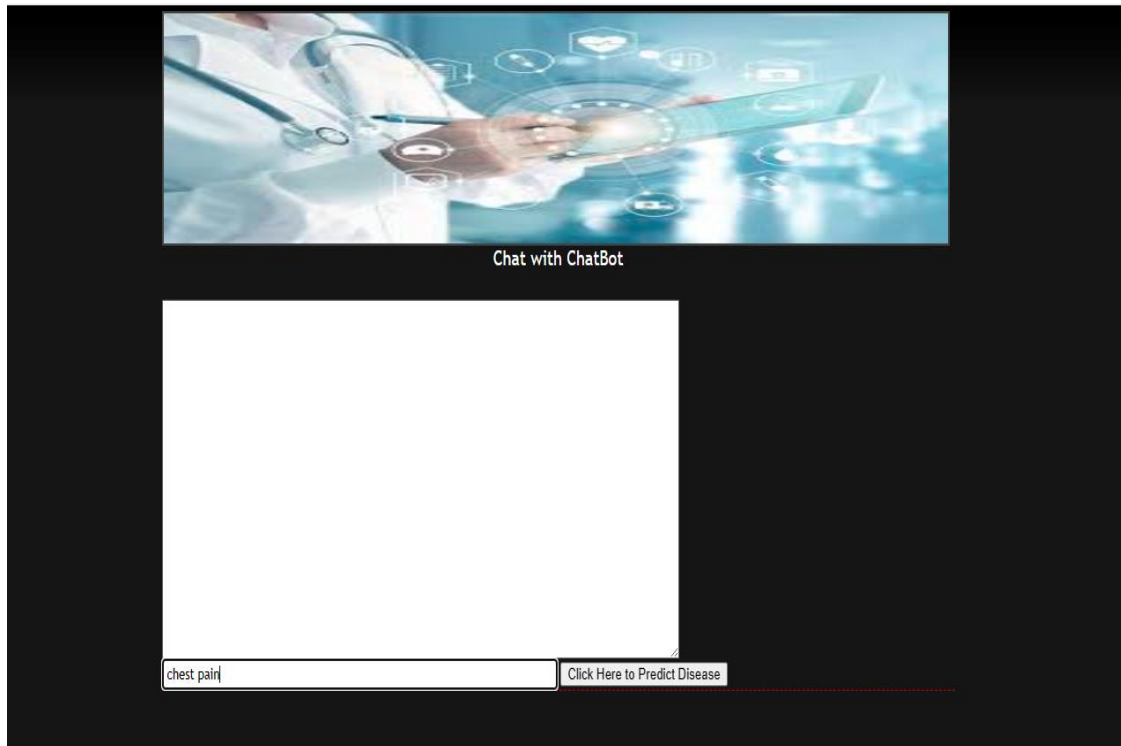


FIG-11 In above screen I gave symptoms as ‘Chest Pain’ and below is the output

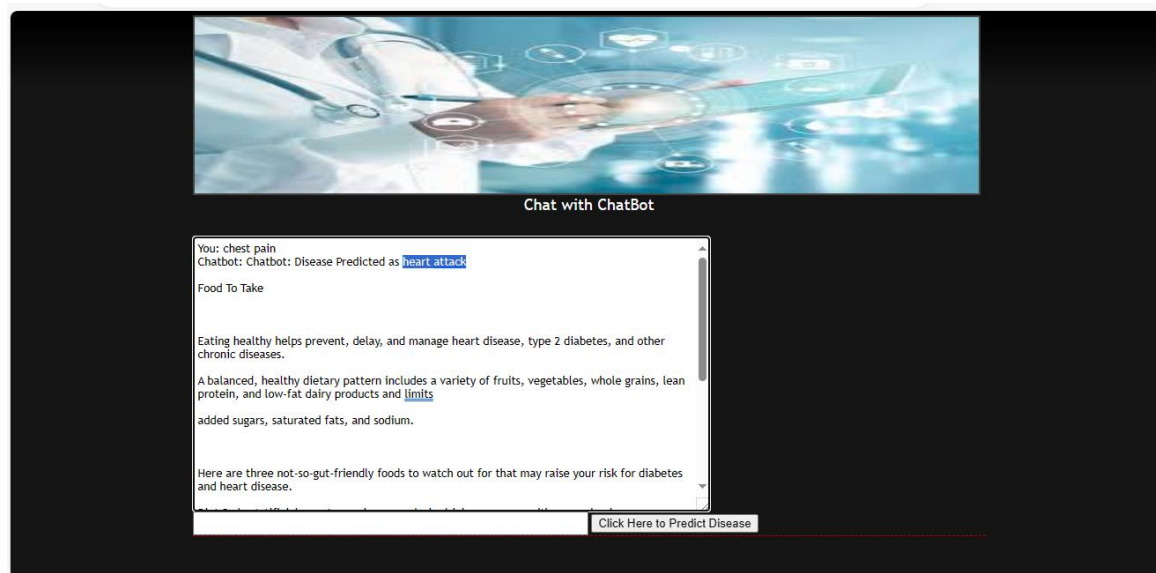


FIG-12 In above screen in blue colour text Chatbot predicted disease as ‘Heart Attack’ for symptom ‘Chest Pain’. Similarly you can search for any symptoms and now click on ‘Lifestyle & Disease Information’ link to view static information about disease

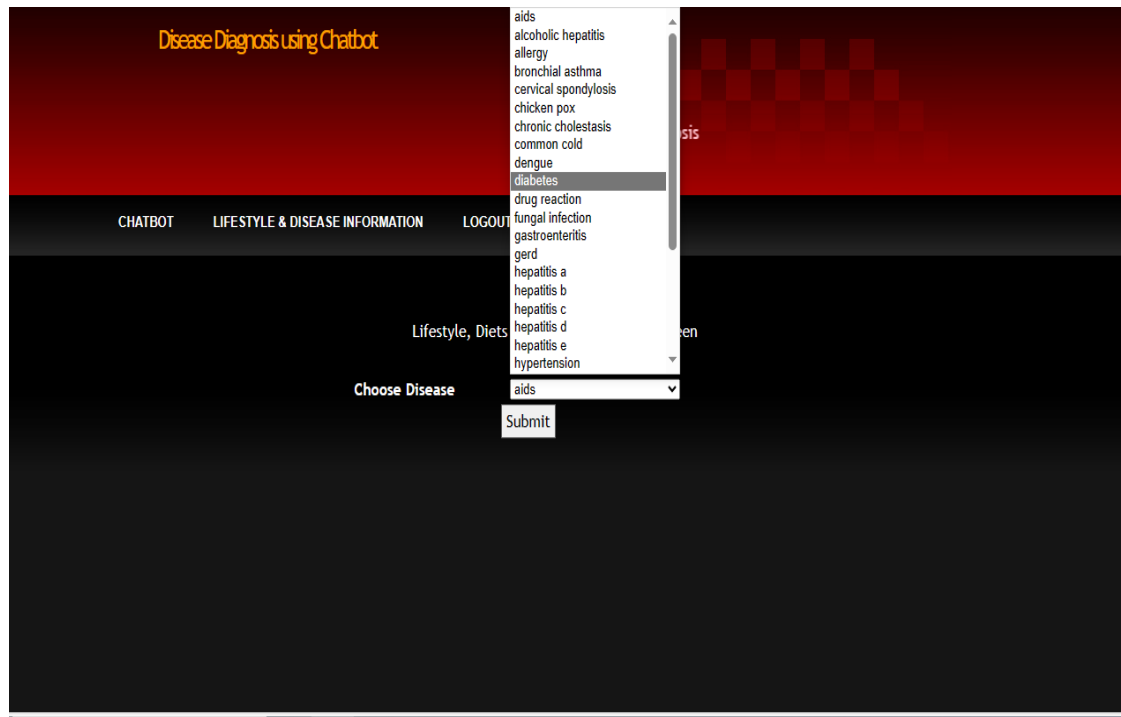


FIG-13 In above screen user can select specific disease and then press button to get disease, diet information like below screen

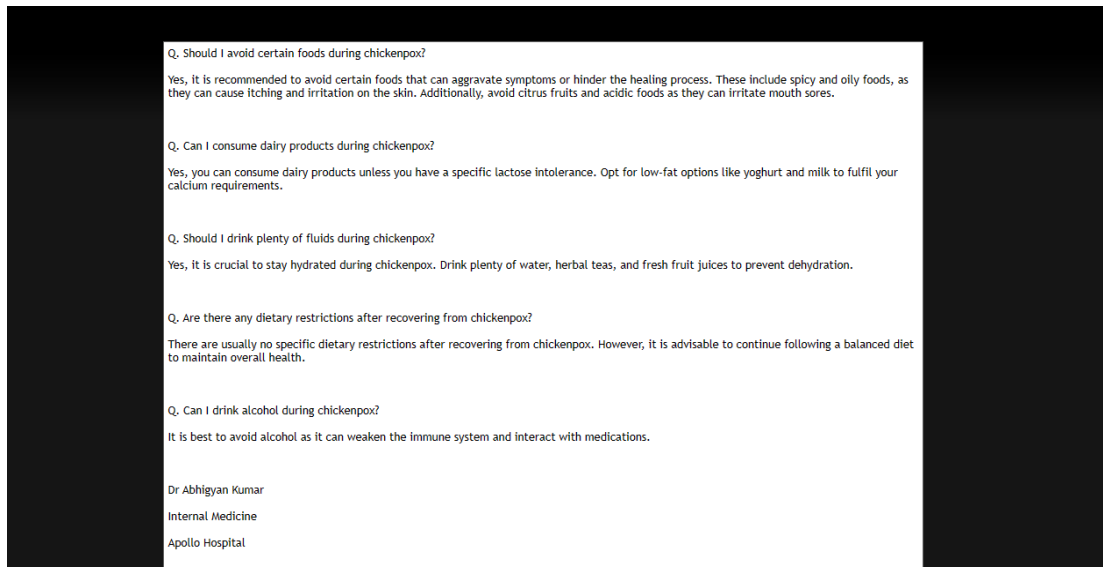


FIG-14 In above screen user can see some answers about selected disease along with doctor details.

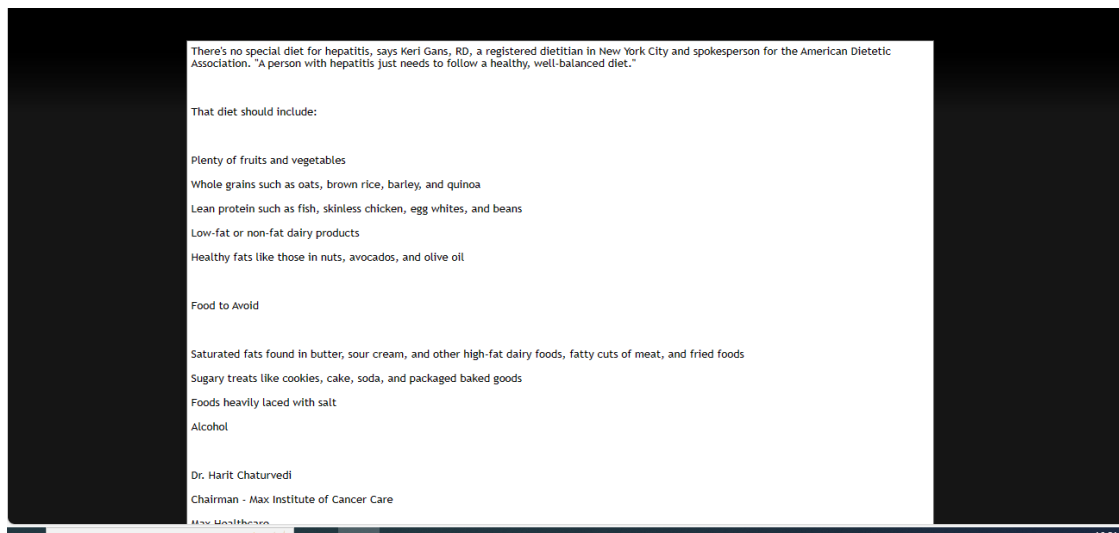


FIG-15 In above screen for some disease we can see information about Food to take and avoid

CHAPTER-6

CONCLUSION

In conclusion, the "Disease Diagnosis using Chatbot" project offers an accessible and user-centric approach to preliminary healthcare guidance. The integration of chatbot technology streamlines the symptom-checking process, empowers users with information, and encourages responsible healthcare decisions. The future of disease diagnosis using chatbots holds immense potential to revolutionize healthcare delivery by offering accessible, personalized, and efficient medical assistance to individuals worldwide. By leveraging advances in AI, NLP, and healthcare technology, chatbots can contribute significantly to improving patient outcomes, reducing healthcare costs, and enhancing overall quality of care.

FUTURE SCOPE

The future scope of disease diagnosis using chatbots is promising, with several opportunities for advancement and innovation. As artificial intelligence (AI) and natural language processing (NLP) technologies continue to evolve, chatbots are expected to become even more sophisticated and capable of providing accurate and personalized medical advice.

One significant area of development lies in improving the diagnostic accuracy of chatbots through enhanced AI algorithms and machine learning models. By leveraging large datasets of medical information, including electronic health records, clinical trials, and research papers, chatbots can refine their understanding of various diseases and symptoms, leading to more precise diagnoses.

CHAPTER-7

REFERENCES

1. Smith, J. "Challenges in Online Disease Diagnosis: A Review of Existing Systems and Limitations."
2. Johnson, E. "Enhancing User Experience in Disease Diagnosis: The Role of Chatbot Technology."
3. Brown, M. "Natural Language Processing in Healthcare: Advances and Applications in Chatbot Development."
4. Davis, S. "Privacy and Security Considerations in Healthcare Chatbots: Best Practices and Implementations."
5. White, D. "Emergency Response Features in Symptom-checking Chatbots: Ensuring User Safety."