A Project report on

AI-Driven Disease Insights-Doctor Referral and Appointment Utility

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

Bachelor of Technology

in

Computer Science and Engineering

Submitted by

A.Bhanu Prasad
(20H51A0528)

Ruheena Naaz
(20H51A0548)

A.Bhagya Sree
(20H51A0557)

Under the esteemed guidance of

Mr. B. K. Chinna Maddileti
(Assistant Professor)

Department of Computer Science and Engineering

CMR COLLEGE OF ENGINEERING& TECHNOLOGY

(UGC Autonomous)
*Approved by AICTE *Affiliated to JNTUH *NAAC Accredited with A$^+$ Grade
KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2020- 2024

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Major Project Phase II report entitled **"AI-Driven Disease Insights-Doctor Referral and Appointment Utility"** being submitted by A.Bhanu Prasad Reddy(20H51A0528) , Ruheena(20H51A0548) , A.Bhagya Sree (20H51A0557) in partial fulfillment for the award of **Bachelor of Technology in Computer Science and Engineering** is a record of bonafide work carried out his/her under my guidance and supervision.

The results embodies in this project report have not been submitted to any other University or Institute for the award of any Degree.

Mr.B.K.Chinna Maddileti
Assistant Professor
Dept. of CSE

Dr. Siva Skandha Sanagala
Associate Professor and HOD
Dept. of CSE

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

With great pleasure we want to take this opportunity to express my heartfelt gratitude to all the people who helped in making this project work a grand success.

We are grateful to **Mr.B.K.Chinna Maddileti, Assistant Professor** , Department of Computer Science and Engineering for his valuable technical suggestions and guidance during the execution of this project work.

We would like to thank **Dr. Siva Skandha Sanagala,** Head of the Department of Computer Science and Engineering, CMR College of Engineering and Technology, who is the major driving forces to complete my project work successfully.

We are very grateful to **Dr. Ghanta Devadasu**, Dean-Academics, CMR College of Engineering and Technology, for his constant support and motivation in carrying out the project work successfully.

We are highly indebted to **Major Dr. V A Narayana,** Principal, CMR College of Engineering and Technology, for giving permission to carry out this project in a successful and fruitful way.

We would like to thank the **Teaching & Non- teaching** staff of Department of Computer Science and Engineering for their co-operation

We express our sincere thanks to **Shri. Ch. Gopal Reddy**, Secretary, CMR Group of Institutions, for his continuous care.

Finally, We extend thanks to our parents who stood behind us at different stages of this Project. We sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project work.

A.Bhanu Prasad	20H51A0528
Ruheena Naaz	20H51A0548
A.Bhagya Sree	20H51A0557

TABLE OF CONTENTS

List of Figures

ABSTRACT

Many deadly diseases are preventable and can be cured if treated at an early stage. However, people tend to ignore the early symptoms of a disease as they are mild. And till the time they become aware of the disease, the infection has already spread into their body. Hence, it is always advisable to have a regular body check-up to avoid health issues later. The Online Doctor Application proposed here is an android based smart application that will help the users to identify a disease based on the symptoms. When a patient enters his /her symptoms the system will incorporates custom made algorithms like random forest algorithm and KNN classifier to identify the disease. Then based on the disease identified the system will recommend a list of doctors having expertise on that particular disease. The patients can book an appointment with the doctors as per their preferred time & date. The patients can also chat with a doctor to get consultations

The primary goal of this project is to harness the power of AI to transform the healthcare landscape by empowering both patients and healthcare providers with valuable information and resources. The platform leverages machine learning algorithms and data analysis techniques to deliver personalized disease insights to patients based on their medical history, symptoms, and available healthcare data. These insights serve as a valuable resource for patients to better understand their condition, treatment options, and potential risks.

CHAPTER 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1. Problem Statement

In today's fast-paced world, access to quality healthcare is essential. We're excited to introduce an innovative solution that puts your health in your hands.It is always advisable to have a regular body check-up to avoid health issues later.Introducing the AI-Driven Disease Insights, Doctor Referral & Appointment Utility —an indispensable tool that seamlessly combines cutting-edge AI technology with healthcare access. This application offers users the ability to gain deep insights into their health conditions through symptom analysis, receive tailored doctor referrals, and effortlessly schedule appointments, revolutionizing the way we approach healthcare in a user-friendly, efficient, and accessible manner.

Overall, this project aims to leverage the capabilities of AI to provide patients with meaningful disease insights, simplify the process of finding the right healthcare professional, and enhance the efficiency of appointment scheduling. By empowering patients with knowledge and facilitating seamless access to healthcare services, this platform contributes to the improved quality of care and patient satisfaction, ultimately promoting the overall well-being of individuals within the healthcare ecosystem.

1.2. Research Objective

Our objective is to develop and evaluate an AI-driven healthcare application that effectively identifies diseases based on symptoms.Assess the effectiveness of the symptom analysis algorithm in providing reliable disease insights.Based on the disease identified this aims to provide accurate doctor referrals. Create a user-friendly interface for seamless appointment booking with recommended doctors.with the aim of improving healthcare accessibility and patient outcomes.

1.3 Project Scope and Limitations

Project Scope:

- Users can select the symptoms which they are suffered and able to identify the disease caused by them.
- The app allows users to search for doctors in their area based on specialty, location, and availability. Users can also read reviews from other patients to help them make informed decisions about their healthcare providers.
- Users can easily book appointments with their chosen healthcare providers directly through the app. The app also offers reminders and notifications to help users stay on top of their appointments and manage their healthcare needs more effectively.

Limitations:

- **Data Quality:** The accuracy of disease insights and doctor referrals heavily relies on the quality and completeness of patient data. Inaccurate or incomplete data can lead to erroneous recommendations.
- **Privacy Concerns:** The project should adhere to stringent data privacy regulations, which may limit the extent of data sharing and analysis, potentially impacting the quality of insights.
- **AI Algorithm Accuracy:** AI algorithms, while powerful, are not infallible. There may be limitations in the algorithms' ability to accurately diagnose or match patients with doctors, particularly for rare or complex conditions.
- **User Adoption:** The success of the system relies on user adoption. If patients or healthcare providers are reluctant to use the platform, its impact may be limited.
- **Healthcare System Integration:** Integrating with existing EHR systems and databases can be complex and may require cooperation from various healthcare institutions, which could be challenging to achieve.

CHAPTER 2

BACKGROUND WORK

CHAPTER 2

BACKGROUND WORK

2.1 Disease Prediction using Machine Learning based on User Symptoms

2.1.1 Introduction:

The healthcare and medical sector are more in need of data mining today. When certain processing methods are used in the right way, valuable information is often extracted from a large database which may help the medical practitioner to wish for early decision and improve health services. Machine Learning helps in prediction by its emerging approach for the diagnosis of a disease. This paper depicts the prediction of diseases by using patient symptoms. Vamshi and Sai developed “Prediction of disease using ML” in International Journal for Research in Applied Science &Engineering Technology (IJRASET) Volume 10 Issue VI June 2022[1].Its implementation is completed through the python programming language. Diseases and health-related problems like malaria, Impetigo, Diabetes, Migraine, Jaundice, Chickenpox, etc., cause a significant effect on one’s health and sometimes might also lead to death if ignored. The healthcare industry can make a decent deciding making by, “mining” the large database they possess i.e. by extracting the hidden patterns and relationships within the database. Data processing algorithms like Decision Tree, Random Forest, and Naive Bayes algorithms can provide a remedy to the current situation. Hence, we have developed an automatic system that can discover and extract hidden knowledge associated with the diseases from a historical (diseases-symptoms) database according to the rule set of the algorithms..

2.1.2 Merits,Demerits and Challenges

Merits:

- Early Detection: Machine learning can aid in the early detection of diseases, allowing for timely intervention and treatment, potentially improving patient outcomes.
- Personalized Healthcare: Machine learning models can provide personalized disease predictions based on individual symptoms, medical history, and other data, resulting in more tailored treatment plans.
- Efficiency: Automation of disease prediction can help healthcare providers streamline the diagnostic process, reducing manual workloads and minimizing errors.

- Data-Driven Insights: Machine learning models can analyze large datasets to identify patterns and correlations in symptoms, enabling more accurate disease predictions.
- Improved Accuracy: Machine learning algorithms can process and analyze a vast amount of data, leading to potentially more accurate disease predictions compared to traditional methods.

Demerits:

- Data Quality: The accuracy of predictions heavily relies on the quality and completeness of input data. Inaccurate or incomplete symptom data can lead to incorrect predictions.
- Lack of Human Expertise: Machine learning models, while powerful, may lack the nuanced understanding and clinical judgment of healthcare professionals. They are not a replacement for doctors but should be used as tools to aid decision-making.
- Privacy Concerns: Handling sensitive healthcare data for prediction purposes requires stringent data privacy measures to protect patient information.
- Algorithm Bias: Machine learning models can inherit biases from the training data, potentially leading to disparities in disease predictions, particularly in underrepresented populations.
- Limited Data: Machine learning models may struggle to provide accurate predictions for rare diseases or conditions with limited historical data.

Challenges:

- Ethical Concerns: The ethical use of patient data for disease prediction, consent, and data security are paramount challenges in implementing machine learning in healthcare.
- Regulatory Compliance: Adhering to healthcare regulations and privacy laws, such as HIPAA in the United States, is crucial and can be a significant challenge.
- Education and Acceptance: Educating healthcare professionals and patients about the benefits and limitations of machine learning in healthcare is important for successful adoption and acceptance.

2.1.3 Implementation:

A. Input from the user (Symptoms) While designing the model we've assumed that the user contains a clear idea about the symptoms he's experiencing. The Prediction developed considers 95 symptoms amidst which the user can give the symptoms his processing because of the input.

B. Data preprocessing The data mining technique that transforms the information or encodes the information to a form that may be easily interpreted by the algorithm is named data preprocessing. The preprocessing techniques utilized in the presented work are x Data Cleaning: Data is cleansed through processes such as filling in missing value, thus resolving the inconsistencies within the data. X Data Reduction: We have also replaced missing value with actual value and we also remove orthographical errors from dataset.

C. Models selected This system aims to predict the diseases with the help of three algorithms:

● Disease Tree Classifier

● Random Forest Classifier

● Naïve Bayes Classifier A study is presented to inspect the performance of every algorithm of the considered database.

D. Output (diseases) Once the system is instructed with the training set using the mentioned algorithms a rule set is formed and when the user the symptoms are given as an input to the model, those symptoms are processed according to the rule set developed, thus making classifications and predicting the foremost likely disease.

The knowledge germination concerning algorithms and algorithms utilization in the medical domain it can be noted that methods and methodologies have developed that have allowed complicated data reports by mild and honest use of algorithms like disease tree classifier, random forest classifier, naive Bayes classifier. This paper introduces a comprehensive similar study of three algorithms completion on a medical record the execution analyzed is done with an accuracy score artificial intelligence will perform an even more significant role in data interpretation in the future due to the availability of enormous data produced and deposited by advanced technology.

2.2 Practo App

2.2.1 Introduction:

Practo is a widely recognized and popular healthcare technology platform that offers a variety of services aimed at improving patient care, enhancing healthcare provider efficiency, and enabling easier access to healthcare services. Founded in 2008, Practo has since expanded its presence to several countries and provides a wide range of features, including doctor discovery, appointment booking, telemedicine consultations, and health information resources. Shashank ND and Abhinav Lal started the operations of healthtech platform Practo in 2007[2].The Practo app provides users with features like booking doctor appointments, consulting with healthcare professionals online, ordering medicines, storing and managing medical records, and reading health-related articles. It aims to make healthcare more accessible and convenient for users by leveraging technology. Additionally, Practo also provides software solutions for healthcare providers to manage their practices efficiently. As of my last update in January 2022, Practo was widely used in several countries, particularly in India, where it originated, and it continues to be a popular choice for individuals seeking healthcare services. However, for the most accurate and up-to-date information, I recommend checking the latest reviews and information on the Practo app from reliable sources or directly from the app store.

2.2.2 Merits,Demerits and Challenges

Merits:

1. Doctor Discovery: Practo simplifies the process of finding healthcare professionals by offering an extensive database of doctors, clinics, and hospitals. Patients can search based on specialties, locations, user ratings, and other filters.
2. Appointment Booking: Patients can easily book, reschedule, or cancel appointments through Practo's platform, reducing the need for lengthy phone calls or in-person visits. This convenience enhances the overall patient experience.
3. Telemedicine: Practo offers telemedicine services, enabling patients to consult with healthcare providers remotely. This feature is particularly valuable in situations where in-person visits are challenging, as seen during the COVID-19 pandemic.
4. Health Information: Practo provides users with access to health information, articles, and tools to better understand their medical conditions and symptoms. This empowers patients to be more informed about their health.

5 Patient Feedback and Reviews: Users can read patient feedback and reviews to make informed decisions when choosing healthcare providers, enhancing transparency and trust.

6 Practice Management Tools: Healthcare practitioners can use Practo's practice management software to efficiently manage appointments, patient records, billing, and other administrative tasks.

Demerits:

1. Data Privacy Concerns: Like any healthcare technology platform, Practo must handle sensitive patient data. Privacy and security concerns can arise, especially given the increasing importance of data protection regulations.
2. Accuracy of Information: The quality of information, such as user reviews and doctor profiles, may vary. Users must exercise caution and verify information to ensure they receive accurate and reliable healthcare services
3. SMINQ was founded in 2015 by Shachin Bharadwaj, Santhosh Nagarajan and Sheldon D’Souza this can provide more efficient outcome by showing appointment waiting time[3].
4. User Adoption: Encouraging patients and healthcare providers to adopt the platform can be a challenge. Some individuals, particularly in less technologically advanced regions, may be resistant to adopting digital healthcare services.
5. Inadequate Follow-Up Care: The app may not adequately facilitate follow-up care or continuity of treatment, particularly for complex medical conditions requiring ongoing monitoring and management.
6. Overreliance on Technology: There is a risk of overreliance on technology-driven healthcare solutions, which may overlook the importance of human interaction, empathy, and holistic care in the patient-provider relationship.
7. Limited Scope of Services: While Practo offers a range of healthcare services, there may be gaps or limitations in coverage, particularly for specialized medical needs or ancillary services not offered through the platform.

Challenges:

1. Regulatory Compliance: Practo operates in a highly regulated industry with strict data privacy and healthcare regulations. Complying with these regulations, whichcan vary by country, can be a significant challenge.
2. Quality Control: Maintaining the quality and accuracy of information on the platform, including doctor profiles and user reviews, is an ongoing challenge.
3. Competition: The healthcare technology industry is highly competitive, with numerous other platforms vying for market share and user attention.
4. User Trust and Data Security: Earning and maintaining user trust in handling their sensitive healthcare data is crucial, as any security breach or data mishandling can have severe consequences.
5. Quality of Care: Maintaining consistent and high-quality healthcare services, especially in the case of online consultations, requires stringent monitoring, training, and adherence to professional standards.

2.2.3 Implementation:

The implementation of a healthcare platform like Practo involves a systematic approach. It begins with identifying specific needs and regulatory requirements, selecting the right technology stack, and designing user-friendly interfaces. Integration with healthcare providers and institutions is crucial, allowing patients to find doctors, book appointments, and access telemedicine services. Robust security measures are implemented, and mobile apps are developed for accessibility. Testing and continuous user feedback drive improvements, and marketing strategies aim to attract users and healthcare providers. Regional adaptation is important to meet local healthcare practices and regulations. This multifaceted process requires meticulous planning and a commitment to user privacy and data security.User feedback and support are crucial for ongoing improvement and maintaining a positive user experience.

2.3 Disease Predictor Based on Symptoms Using Machine Learning

2.3.1 Introduction:

A well-functioning healthcare system is critical to the economy and the well-being of humanity. Between the world, we live in now and the world we lived in a few decades ago, there has been a substantial amount of change. Everything has gotten more disorderly and unattractive. In this situation, doctors and nurses are doing everything they can to save people's lives, even if it means putting their own lives in danger. Virtual doctors are board-certified doctors who choose to practice online using video and phone consultations rather than in-person consultations, albeit this is not always practicable in an emergency. In the absence of human error, machines are thought to be superior to humans because they can do jobs faster while maintaining a consistent level of precision. A disease predictor, often known as a virtual doctor, A disease predictor can save a person's life in extreme instances, such as COVID-19 and EBOLA, by recognizing their health without requiring physical touch. There are virtual doctors on the market now, but they lack the capacity to provide the kind of precision that is required. This Condition's Prognosis To forecast sickness, we'll use hospital data and Machine Learning methods based on the Python programming language and the Tkinter interface. Doctors may make errors when diagnosing a patient's disease, however, disease prediction systems with machine learning algorithms can help produce accurate results in these situations.

2.3.2 Merits,Demerits and Challenges

Merits:

1. High Accuracy:The 97% accuracy rate is a significant merit of the project. It indicates that the machine learning models have been trained effectively and can make accurate predictions based on symptoms.

2. Early Detection: The project can help in early detection of diseases based on symptoms, which can be crucial in preventing and treating diseases at an early stage, potentially saving lives.

3. Ease of Use: Such a system can be user-friendly and accessible to the general public. Patients can input their symptoms easily, and the system can provide quick predictions, which can be especially helpful for individuals in remote areas or those who do not have immediate access to healthcare professionals.

4. Cost-Efficiency: This project can help reduce healthcare costs by efficiently allocating resources and avoiding unnecessary medical tests or appointments.

5Reduced Healthcare Costs: Early detection and prevention of diseases through ML-based predictors can lead to reduced healthcare costs by avoiding expensive treatments and hospitalizations associated with advanced disease stages.

Demerits:

1. Data Quality: The accuracy of the predictions is highly dependent on the quality of the training data. If the data used for training is not representative or contains biases, the model's performance can be compromised.

2. Overfitting: There's a risk of overfitting, where the models perform well on the training data but fail to generalize to unseen data. This can lead to inaccurate predictions in real-world scenarios.

3.Limited Scope: The project might be limited to a specific set of diseases or symptoms. Expanding it to cover a broader range of conditions can be challenging.

4. Privacy Concerns: Gathering and storing patient symptom data can raise privacy concerns, and it's important to ensure that data is handled securely and in compliance with relevant regulations like GDPR or HIPAA.

Challenges:

1.Maintenance:Machine learning models require regular updates and maintenance to stay accurate as medical knowledge evolves. Incorporating new data and adjusting the models can be an ongoing challenge.

2. Scalability: As the system gains popularity, it may need to handle a large volume of symptom data. Scalability can be a challenge in terms of computing resources and response times.

2.3.3. Implementation

To construct a disease prediction based on symptoms, we applied four machine learning algorithms: Decision Tree, Random Forest, KNN, and Naive Bayes.There is an accurate forecast for our model using these tactics. The Prognosis of the Illness Currently, the effort is in full swing. Machine Learning is being used to diagnose and prevent disease in its infancy. As we all know, humanity has become so engrossed in the competitive environment of economic advancement that it has lost sight of its own well-being. Studies show that 40% of people ignore small symptoms, which might lead to more serious problems in the future. The project's interface is also built with Tkinter, a Python library interface. The user must first enter their name, then select symptoms from a drop-down menu; alternatively, the user must enter all symptoms, after which the system will return an exact result. Four machine learning approaches were used to create this forecast: Decision Tree, Random Forest, KNN, and Naive Bayes. When the user enters all of the symptoms and simply presses the Random Forest button result opens. The result is computed using that method; similarly, we've utilized four ways to provide a more thorough perspective of the data, and the user must be satisfied with the anticipated conclusion.

CHAPTER 3
PROPOSED SYSTEM

CHAPTER 3

PROPOSED SYSTEM

3.1 Objective of Proposed Model:

- Our objective is to develop and evaluate an AI-driven healthcare application that effectively identifies diseases based on symptoms.
- Assess the effectiveness of the symptom analysis algorithm in providing reliable disease insights.
- Based on the disease identified this aims to provide accurate doctor referrals.
- Create a user-friendly interface for seamless appointment booking with recommended doctors. with the aim of improving healthcare accessibility and patient outcomes

3.2 Algorithm Used For Proposed Model:

Random Forest:

The RF classifier is a supervised machine learning algorithm and is mostly will use for regression and classification problems. The RF will predict the results based on multiple outcomes of a number of decision trees and which result is voter more time that one becomes as a predicted class. In this system, we used *RandomForestClassifier* which is imported from *sklearn. ensemble* package for disease prediction. In the experimental results, the RF classifier is provided with *88* percent accuracy.

Decision Tree:

The DT classifier is a tree-based predicted classifier and it is also a supervised learning classifier used for resolving the classification problems. The DT classifier will predict the disease based on the IF-THEN methodology. While training the classifier with help of a dataset, the DT classifier will form the tree structure, so based on testing inputs, it will compare with all child nodes and it will reach the leaf node that is a predicted class. In this system, the *Decision Tree Classifier* is used for disease prediction and this predefined classifier is imported from *sklearn. tree* package. Based on experimental results, the DT provided an accuracy of 82.18 percent.

KNN : The K-Nearest Neighbors (KNN) algorithm is a versatile and straightforward approach used for classification and regression tasks in machine learning. Unlike other algorithms, KNN does not build an explicit model during training; instead, it stores the entire training dataset and makes predictions based on the similarity of new data points to existing ones. This lazy learning approach allows KNN to adapt to complex decision boundaries and handle nonlinear relationships in the data effectively**.**In the experimental results, the KNN provided with *93* percent accuracy.

3.3 Designing:

3.3.1 UML Diagram:

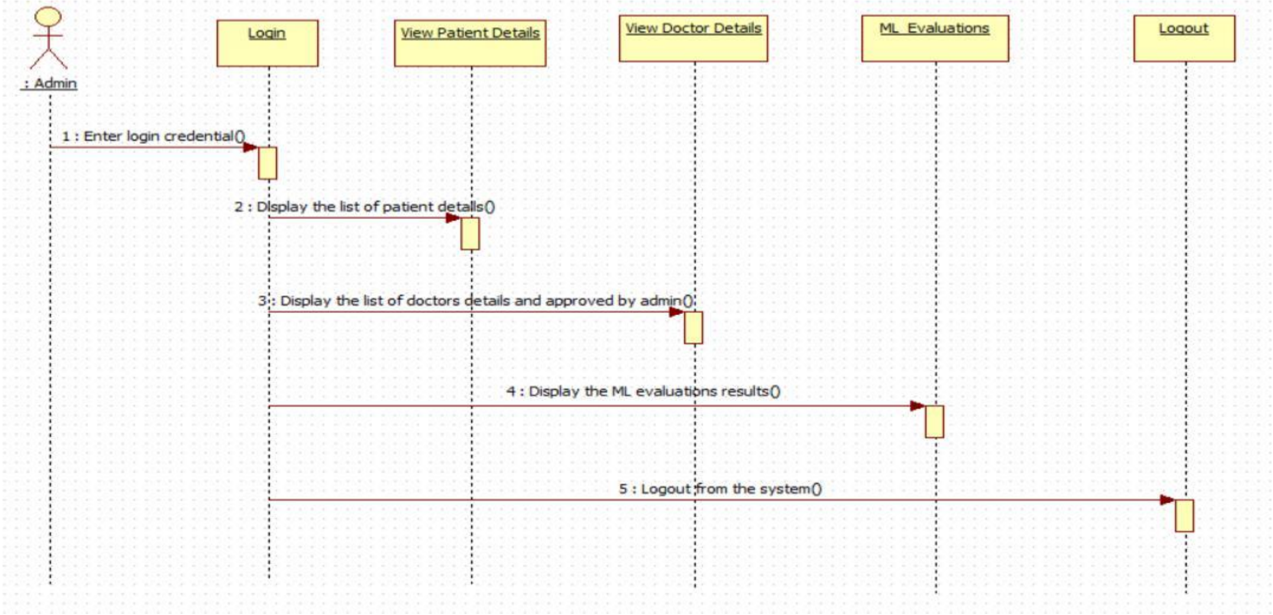

: Admin
Login
View Patient Details
View Doctor Details
ML_Evaluations
Logout
1 : Enter login credential()
2 : Display the list of patient details()
3 : Display the list of doctors details and approved by admin()
4 : Display the ML evaluations results()
5 : Logout from the system()

Fig.3.3.1.1 : Admin uml diagram

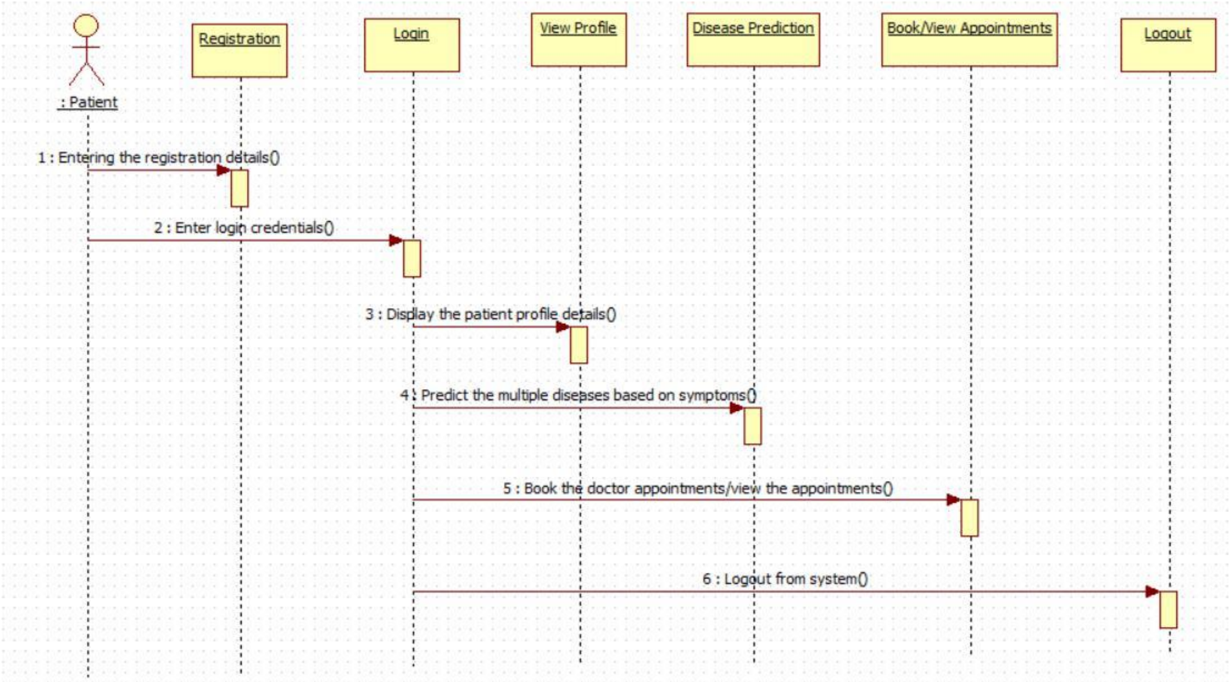

: Patient
Registration
Login
View Profile
Disease Prediction
Book/View Appointments
Logout
1 : Entering the registration details()
2 : Enter login credentials()
3 : Display the patient profile details()
4 : Predict the multiple diseases based on symptoms()
5 : Book the doctor appointments/view the appointments()
6 : Logout from system()

Fig.3.3.1.2 : Patient uml diagram

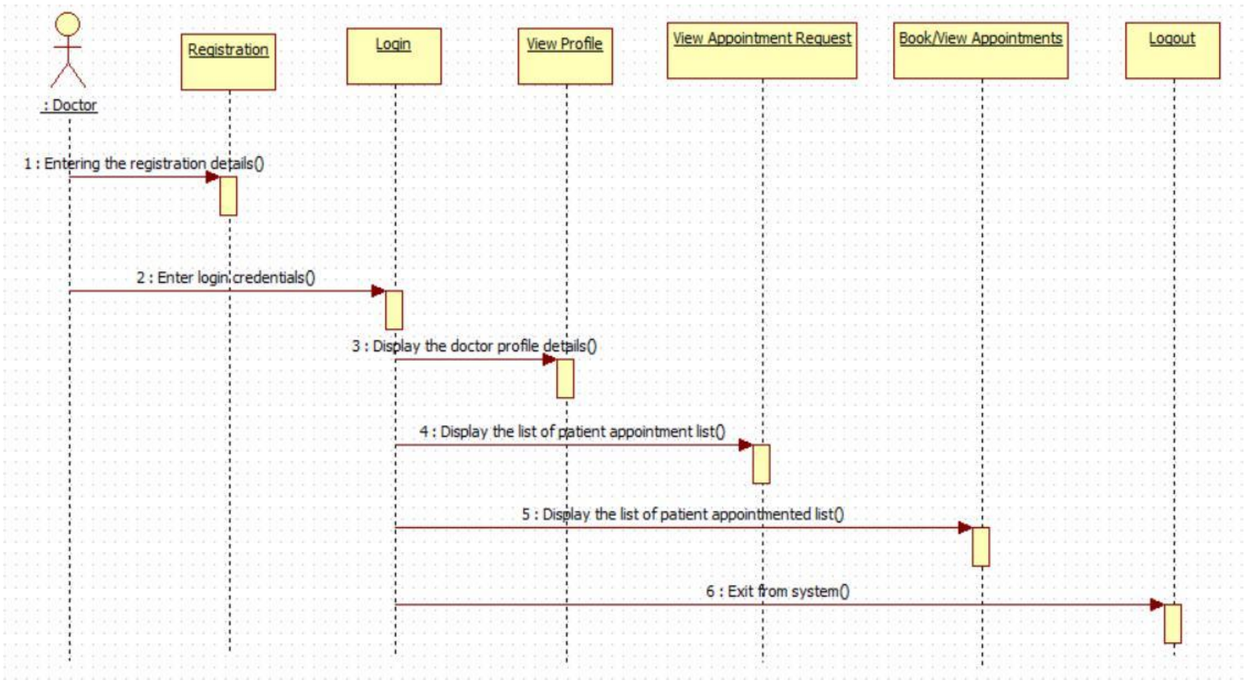

: Doctor
Registration
Login
View Profile
View Appointment Request
Book/View Appointments
Logout
1 : Entering the registration details()
2 : Enter login credentials()
3 : Display the doctor profile details()
4 : Display the list of patient appointment list()
5 : Display the list of patient appointmented list()
6 : Exit from system()

Fig.3.3.1.3 : Doctor uml diagram

3.4 Stepwise Implementation and Code:

3.4.1 Disease Prediction

Input:

- Users enter their symptoms (e.g., fever, cough, fatigue).

Processing:

- Utilize machine learning models (e.g., Decision Trees, Random Forest) trained on symptom-disease datasets.
- Predict the most likely disease based on input symptoms.

Output:

- Displays the predicted disease (e.g., “You may have Typhoid.”).

3.4.2 Doctor Recommendation

Input:

- The predicted disease from the previous step.

Processing:

- Maps the disease to relevant specialists (e.g., cardiologist for heart-related issues).

- Suggest doctors based on the predicted disease.

Output:

- Displays recommended doctors (e.g., “Consult with Dr. Smith, a cardiologist.”).

3.4.3 Appointment booking

- After disease prediction Patients can view available doctors and their schedules and book appointments by selecting a recommended doctor and preferred time slot.
- The above-given patient details are sent to the doctor for appointment booking in the form of notifications.
- After patient book an appointment Doctors can view their upcoming appointments and mark appointments as confirmed, rescheduled, or cancelled.

3.4.4 Architecture

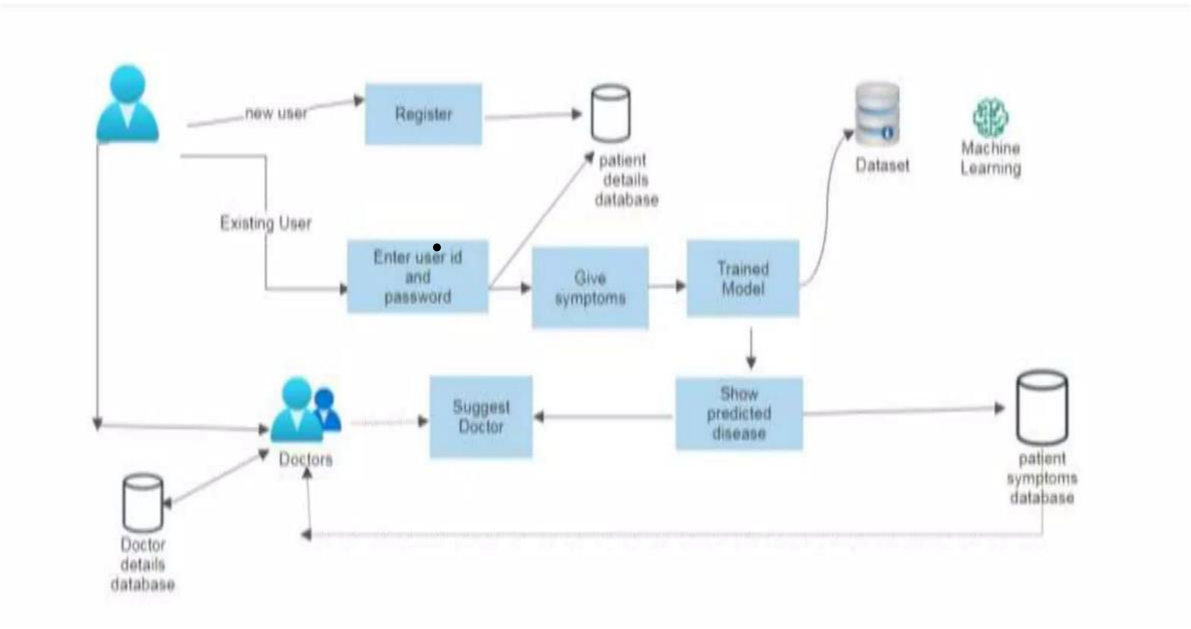


Fig.3.4.4.1 : Architecture

3.4.5. System Requirements

Operating System	: Windows family
Technology	: Python 3.6
Front-end Technology	: HTML, CSS, JS
IDE	: PyCharm
Web framework	: Flask

Install python 3.6 and pycharm IDE and create a new project.

Install Python Modules or Libraries in pychram.

Sklearn: It is used for importing all machine learning classifiers and feature selection.

Pandas: It is used for reading the data from the CSV file.

Matplotlib: It is used for generating bar charts.

Flask: It is a web framework and is used for developing web applications.

Source Code:

Database Connection

```
import mysql.connector
class DBConnection:
   @staticmethod
   def getConnection():
       database = mysql.connector.connect(host="localhost", user="root",
passwd="root", db='disease_prediction')
       return database
if __name__=="__main__":
   print(DBConnection.getConnection())
```

Index.py

```
from flask import Flask, render_template, request,flash
import pandas as pd
import csv
from matplotlib.backends.backend_agg import
   FigureCanvasAgg as FigureCanvas
from matplotlib.figure import Figure
import io
import random
from flask import Response
from DBConnection import DBConnection
import re
from flask import session
from werkzeug.utils import secure_filename
import cv2
import sys
from symptoms_list import symptoms
from patient_registration import patient_store
from doctor_registration import doctor_store
import os
from PIL import Image
from disease_detection import prediction
from ML_Evaluations import evaluation_ml
import io
import base64
import shutil
import matplotlib.pyplot as plt
import matplotlib.pyplot as plt2
import matplotlib.pyplot as plt3
import matplotlib.pyplot as plt4

from random import randint
import numpy as np
from matplotlib import pyplot as pyplt
```

```
app = Flask(__name__)
app.secret_key = "abc"
@app.route('/')
def index():
    return render_template('index.html')
@app.route("/admin")
def admin():
    return render_template("admin.html")
@app.route("/patient")
def patient():
    return render_template("patient.html")
@app.route("/doct_rating/<doct_id>/")
def doct_rating(doct_id):
    print("doct_id=",doct_id)
    database = DBConnection.getConnection()
    cursor = database.cursor()
    sql = "select *from doctor where
    userid='"+doct_id+"' "
    cursor.execute(sql)
    records = cursor.fetchall()
    for row in records:
        doct_id=row[7]
        spe= row[3]
    return render_template("/ratings.html",
    doct_id=doct_id,spe=spe)
@app.route("/patienthome") def patienthome():
    return render_template("patienthome.html")
@app.route("/adminhome")
def adminhome():
    return render_template("adminhome.html")
@app.route("/view_patients")
def view_patients():
    database = DBConnection.getConnection()
    cursor = database.cursor()
    sql = "select *from patient "
    cursor.execute(sql)
    records = cursor.fetchall()
    return
    render_template("view_patients.html",rawdata=reco
    rds)
@app.route("/view_doctors")
def view_doctors():
    database = DBConnection.getConnection()
    cursor = database.cursor()
    sql = "select *from doctor where status='wait'"
    cursor.execute(sql)
    records = cursor.fetchall()
    return
    render_template("view_doctors.html",rawdata=reco
    rds)
@app.route("/doctorhome")
def doctorhome():
    return render_template("doctorhome.html")
@app.route("/disease_detection")
def disease_detection():
    symptoms_list=symptoms()
    return
    render_template("disease_detection.html",symptom
    s=symptoms_list)
@app.route("/patient_reg")
def patient_reg():
    return render_template("patient_reg.html")
@app.route("/patient_profile")
def patient_profile():
    database = DBConnection.getConnection()
    cursor = database.cursor()
    uid=session['uid']
    sql = "select *from patient where userid='" + uid +
    "'"
    cursor.execute(sql)
    records = cursor.fetchall()
    for row in records:
        name=row[0]
        gen=row[1]
        age=row[2]
        contact = row[5]
        imgdata = row[6]
        data = base64.b64decode(imgdata)
        with
    open("../Disease_Prediction/static/picture.jpg", 'wb')
    as f:
            f.write(data)
    return
    render_template("patient_profile.html",name=name,
    gen=gen,age=age,contact=contact)
@app.route("/rating",methods =["GET", "POST"])
def rating():
        doct_uid = request.form.get("doct_uid")
        rating = request.form.get("rating")
        database = DBConnection.getConnection()
        cursor = database.cursor()
        sql = "update ratings set "+rating+"="+rating+"+1
    where doct_id='" + doct_uid + "' "
```

```
        cursor.execute(sql)
        database.commit()
        sql = "select *from ratings where doct_id='" +
    doct_uid + "'"
        cursor.execute(sql)
        records = cursor.fetchall()
        for row in records:
            one = row[1]
            two = row[2]
            three = row[3]
            four = row[4]
            five = row[5]
                records=session["bkapntmnts"]
        return
    render_template("view_appointments.html",records
    =records,msg="done")
@app.route("/book_appointment/<spec>/<doct_id>")
def book_appointment(spec,doct_id):
    return
    render_template("book_appointment.html",spec=sp
    ec,doct_id=doct_id)
@app.route("/doctor_accept/<sno>")
def doctor_accept(sno):
    database = DBConnection.getConnection()
    cursor = database.cursor()
    sql = "select *from bkappointments where sno='" +
    sno + "'"
    cursor.execute(sql)
    records = cursor.fetchall()
    for row in records:
        pname=row[2]
        page=row[3]
        pgen = row[4]
        date = row[8]
    return
    render_template("doctor_accept.html",sno=sno,pna
    me=pname,page=page,date=date,pgen=pgen)
@app.route("/doctor_reject/<sno>")
def doctor_reject(sno):
    database = DBConnection.getConnection()
    cursor = database.cursor()
    sql = "update bkappointments set status='Rejected'
    where sno='" + sno + "'"
    cursor.execute(sql)
    database.commit()
    database2 = DBConnection.getConnection()
    cursor2 = database2.cursor()
    dct_uid = session['dct_uid']
    sql = "select *from bkappointments where
    doctor_uid='" + dct_uid + "' and status='wait' "
    cursor2.execute(sql)
    records = cursor2.fetchall()
    bkapntmnts = []
    for row in records:
    sno = row[0]
        pid = row[1]
        pname = row[2]
        age = row[3]
        gen = row[4]
        symptoms = row[5]
        disease = row[6]
        dt_time = row[8]
        bkapntmnts.append([pid, pname, age, gen,
    symptoms, disease, dt_time, dct_uid, sno])
    return render_template("appointments.html",
    bkapntmnts=bkapntmnts,msg2='rejected')
@app.route("/book_appointment_doctor",methods
    =["GET", "POST"])
def book_appointment_doctor():
    symptom=""
    doct_uid = request.form.get('doct_uid')
    date = request.form.get('date')
    p_uid=session['uid']
    sql = "select *from patient where userid='" + p_uid
    + "'"
    database = DBConnection.getConnection()
    cursor = database.cursor()
    cursor.execute(sql)
    records = cursor.fetchall()
    for row in records:
        name = row[0]
        gen = row[1]
        age = row[2]
    selected_symptoms=session['selected_symptoms']
    predicted_disease= session['predicted_disease']
    date_time=date

    for row in selected_symptoms:
        symptom=symptom+row+","
    symptom=symptom[0:-1]
 database2 = DBConnection.getConnection()
    cursor2 = database2.cursor()
    sql = "insert into
    bkappointments(patient_uid,patient_name,patient_a
    ge,patient_gender,symptoms,disease,doctor_uid,apd
    ate,aptime,status)
    values(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"
    values = (p_uid,name,age,gen,
    symptom,predicted_disease,doct_uid,date_time,"","
    wait")
    cursor2.execute(sql, values)
    database2.commit()

    symptoms_list=symptoms()
    return render_template("disease_detection.html",
    msg="done",symptoms=symptoms_list)
@app.route("/doctors_list", methods =["GET",
    "POST"])

def doctors_list():
    try:
        pname = request.form.get('pname')
        page = request.form.get('page')
        pgen = request.form.get('pgen')

        medication = request.form.get('medication')
        city = request.form.get('city')

        prof = request.form.get('prof')

    selected_symptoms=session['selected_symptoms']
        predicted_disease=session['predicted_disease']
database = DBConnection.getConnection()
        cursor = database.cursor()
        #sql="select *from doctor where specialties='" +
    consultdoctor + "' "
        sql = "select *from doctor where
    specialties='"+prof+"' and
    medication='"+medication+"' and city='"+city+"'
    and status='active'"
        cursor.execute(sql)
        rec_doct_list = cursor.fetchall()

        sql2= "select *from doctor where status='active'"
        cursor.execute(sql2)
        available_doct_list = cursor.fetchall()
        patient_info=[]

        patient_info.append(pname)
        patient_info.append(page)
        patient_info.append(pgen)
        patient_info.append(city)
        return
    render_template("doctors_list.html",patient_info=pa
    tient_info,selected_symptoms=selected_symptoms,
    consultdoctor=prof,predicted_disease=predicted_dis
    ease,rec_doct_list=rec_doct_list,available_doct_list
    =available_doct_list)
    except Exception as e:
        print(e)
        tb = sys.exc_info()[2]
        print(tb.tb_lineno)
    return ""
@app.route("/evaluations" )
def evaluations():


    metrics,accuracy_list,precision_list,recall_list,f1scor
    e_list = evaluation_ml()

    session["accuracy_list"]=accuracy_list

    session["precision_list"] = precision_list

    session["recall_list"] = recall_list

    session["f1score_list"] = f1score_list

    session["metrics"] = metrics

    ml_accuracy()
    ml_precision()
    ml_recall()
    ml_f1score()

    return render_template("ml_evaluations.html",
    metrics=metrics)

#@app.route("/ml_accuracy")
def ml_accuracy():
    accuracy_list=session["accuracy_list"]
    bars = ('RF', 'DT', 'KNN')
    plt.figure(1)
    y_pos = np.arange(len(bars))
    plt.bar(y_pos, accuracy_list, color=['red', 'green',
    'brown'])
    plt.xticks(y_pos, bars)
    plt.xlabel('Classification Algorithms')
    plt.ylabel('Accuracy')
    plt.title('Accuracy of ML Techniques')
    plt.savefig('static/accuracy.png')
    #plt.show()
   #metrics = session["metrics"]
    #return render_template("ml_evaluations.html",
    metrics=metrics)
def ml_accuracy():
    accuracy_list=[]
    accuracy_list.clear()
    accuracy_list=session["accuracy_list"]

    bars = ('RF', 'NB', 'DT', 'VotingClassifier')
    y_pos = np.arange(len(bars))
    plt.bar(y_pos,   accuracy_list,   color=['red',   'green',
    'blue','orange'])
    plt.xticks(y_pos, bars)
    plt.xlabel('Classification Algorithms')
    plt.ylabel('Accuracy')
    plt.title('Accuracy of ML Techniques')
    plt.savefig('static/accuracy.png')

    accuracy_list.clear()
    #fig = pyplt.gcf()

    #return fig

def ml_precision():
    precision_list = []
    precision_list.clear()
    precision_list=session["precision_list"]
    bars = ('RF', 'NB', 'DT', 'VotingClassifier')
    y_pos = np.arange(len(bars))
    plt2.bar(y_pos, precision_list, color=['blue', 'orange',
    'red','green'])
    plt2.xticks(y_pos, bars)
    plt2.xlabel('Classification Algorithms')
    plt2.ylabel('Precision')
    plt2.title('Precision of ML Techniques')
    plt2.savefig('static/precision.png')
    precision_list.clear()'''
```

```
def ml_recall():
    recall_list=session["recall_list"]
    bars = ('RF', 'DT', 'KNN')
    plt3.figure(3)
    y_pos = np.arange(len(bars))
    plt3.bar(y_pos,recall_list,
    color=['orange','blue','red'])
    plt3.xticks(y_pos, bars)
    plt3.xlabel('Classification Algorithms')
    plt3.ylabel('Recall')
    plt3.title('Recall of ML Techniques')
    plt3.savefig('static/recall.png')
    recall_list.clear()

def ml_f1score():
    f1score_list=session["f1score_list"]
    bars = ('RF','DT','KNN')
    plt4.figure(4)
    y_pos = np.arange(len(bars))
    plt4.bar(y_pos,f1score_list,
    color=['green','red','blue'])
    plt4.xticks(y_pos, bars)
    plt4.xlabel('Classification Algorithms')
    plt4.ylabel('F1-Score')
    plt4.title('F1-Score of ML Techniques')
    plt4.savefig('static/fscore.png')
    f1score_list.clear()

@app.route("/videos")
def videos():
    database = DBConnection.getConnection()
    cursor = database.cursor()
    qry = "select * from videos"
    cursor.execute(qry)
    videos = cursor.fetchall()
    return
    render_template("videos.html",videos=videos)

  # CHAT APPLICATION

@app.route("/chat_with_patient/<patinet_userid>")
def chat_with_patient(patinet_userid):
    dct_uid = session['dct_uid']
    database = DBConnection.getConnection()
    cursor = database.cursor()
    qry = "select * from msgs where chatof='" + dct_uid
    + "'" + patinet_userid + "' order by sno ";
    cursor.execute(qry)
    chatrec = cursor.fetchall()
    returnrender_template("chat.html",
    patinet_userid=patinet_userid, chatrec=chatrec,
    dct_uid=dct_uid)

@app.route("/schatbot2",methods=["get","post"])
def schatbot2():
    dct_uid = request.form.get('dct_uid')
    msg = request.form.get('text')
    patinet_userid = session['uid']
    #suid = session['suid']
    database = DBConnection.getConnection()
    cursor = database.cursor()
    qry="insert into msgs(msg, user_, time_, chatof,
    status) values ('%s','%s',now(),'%s','%s')"
    cursor.execute(qry % (msg,
    patinet_userid,dct_uid+patinet_userid, "new"))
    database.commit()
    qry = "select * from msgs where chatof='" + dct_uid
    + "'" + patinet_userid + "' order by sno ";
    cursor.execute(qry)
    chatrec =
    cursor.fetchall() return
    render_template("schat.html",patinet_userid=patinet
    _userid,chatrec=chatrec,dct_uid=dct_uid)

if __name__ == '__main__':
    app.run(host="localhost", port=2024, debug=True)
```

Patient Registration:

```
from DBConnection import DBConnection

def patient_store(name,gender,age,uid,pwd,mno,imgdata):
    sts=0;
    database = DBConnection.getConnection()
    cursor = database.cursor()

    sql = "select count(*) from patient where userid='" + uid + "'"
    cursor.execute(sql)

    res = cursor.fetchone()[0]
    if res > 0:

        sts=0
    else:
        sql = "insert into patient values(%s,%s,%s,%s,%s,%s,%s)"
        values = (name, gender, age, uid, pwd,mno,imgdata)
        cursor.execute(sql, values)
        database.commit()
        sts=1

    return sts;
```

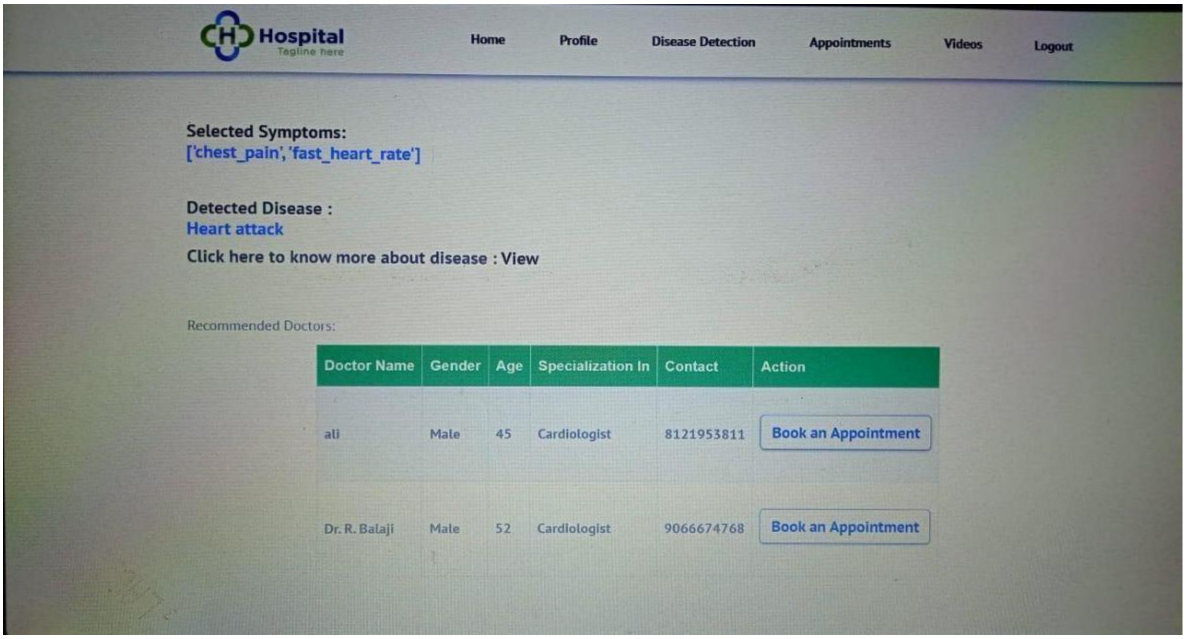
Symptoms

```
def symptoms():
    symptomslist = ['itching', 'skin_rash', 'nodal_skin_eruptions',
'continuous_sneezing', 'shivering', 'chills',
                    'joint_pain',
                    'stomach_pain', 'acidity', 'ulcers_on_tongue', 'muscle_wasting',
'vomiting', 'burning_micturition',
                    'spotting_ urination',
                    'fatigue', 'weight_gain', 'anxiety', 'cold_hands_and_feets',
'mood_swings', 'weight_loss',
                    'restlessness', 'lethargy',
                    'patches_in_throat', 'irregular_sugar_level', 'cough', 'high_fever',
                    'sunken_eyes', 'breathlessness', 'sweating',
                    'dehydration', 'indigestion', 'headache', 'yellowish_skin', 'dark_urine',
                    'nausea', 'loss_of_appetite', 'pain_behind_the_eyes',
                    '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',
               g',
                    'silver_like_dusting', 'small_dents_in_nails', 'inflammatory_nails', 'blister',
                    'red_sore_around_nose',
                    'yellow_crust_ooze']
    return symptomslist
```

CHAPTER 4
RESULTS AND DISCUSSION

4.1 Output Screens

By entering symptoms, we achieve accurate disease predictions. We gathered a lot of information from the disease symptom dataset on Kaggle, which helps our system work better.There will be list of symptoms in which patients can select their symptoms and check for their diseases. In this project we used Pycharm for the execution of our project and SQL for storing the database and SQLyog for checking the data updates in the project.After running in pycharm the admin has to login ,he can approve new doctors registrations.Patient has to login or register then can take actions as to know disease or appointment list.Doctor has to register for first time and then approve patients request.Using this project users can efficiently know about predicted disease at the same time can book appointments



Selected Symptoms:
['chest_pain', 'fast_heart_rate']

Detected Disease :
Heart attack
Click here to know more about disease : View

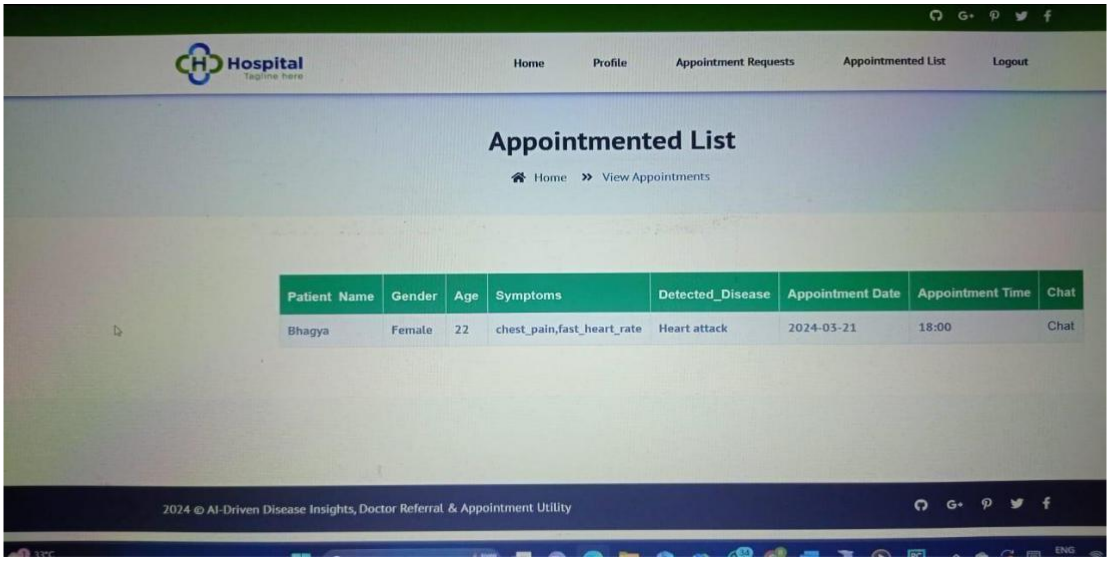
Recommended Doctors:

Doctor Name	Gender	Age	Specialization In	Contact	Action
ali	Male	45	Cardiologist	8121953811	Book an Appointment
Dr. R. Balaji	Male	52	Cardiologist	9066674768	Book an Appointment

Fig.4.1.1 : Disease Prediction

In fig1.patient is login and can select symptoms using ml algorithms disease will be predicted and book appointment with the list of specialized doctors.Then book appointment .this has to be approved by doctor.Other applications focusing on symptom-based disease prediction often fall short in terms of effectiveness, offering only a limited number of diseases with relatively low accuracy and lacking a convenient appointment booking system. In contrast, platforms like Practo aim to enhance overall healthcare experiences by providing comprehensive online medical services and solutions for individuals seeking better self-care.

However, our system stands out in its ability to not only predict diseases but also provide users with a curated list of specialized doctors for the identified illness. This unique feature allows users to select their preferred specialist and seamlessly book appointments forconsultation and treatment.



Appointmented List

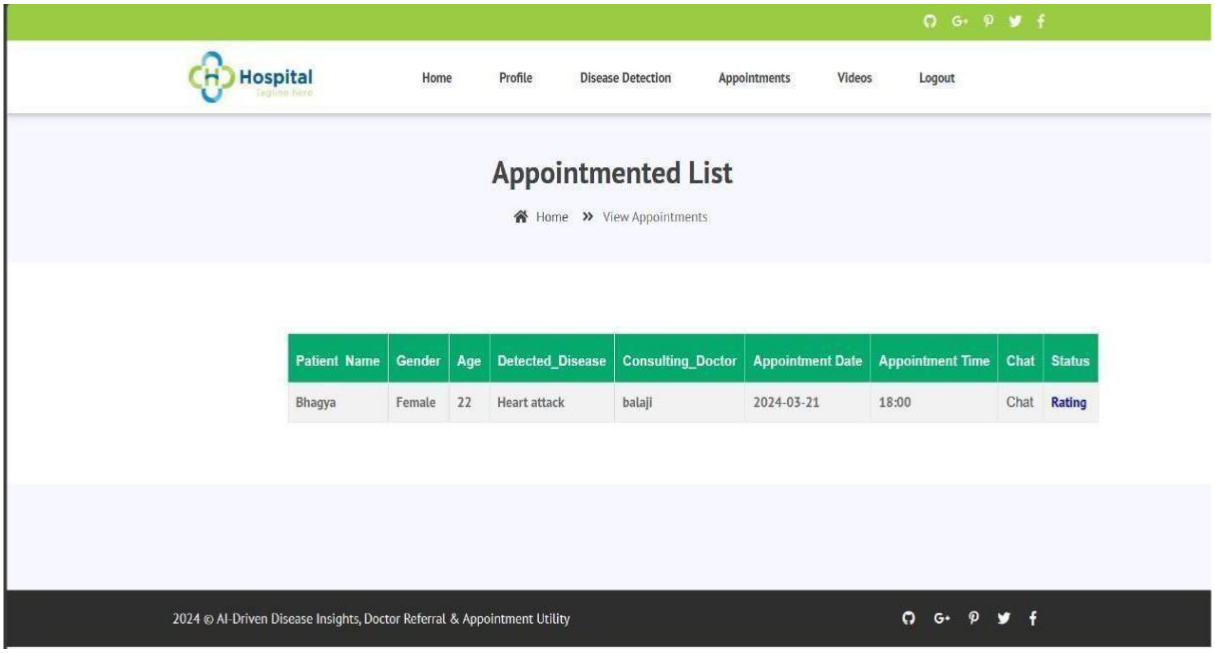
Home » View Appointments

Patient Name	Gender	Age	Symptoms	Detected_Disease	Appointment Date	Appointment Time	Chat
Bhagya	Female	22	chest_pain,fast_heart_rate	Heart attack	2024-03-21	18:00	Chat

2024 © AI-Driven Disease Insights, Doctor Referral & Appointment Utility

Fig.4.1.2 : Doctor’s Appointment List

In fig.2 shows the patient’s appointment acceptance by doctor. It is the interface for managing and arranging appointments depending on the doctors preferences or needs. Using appointments list option doctors check for their availability to provide the desired time to patients depending on their medical needs. After checking and accepting thepatient appointment the above figure is depicted as output.As a result, this project provides an excellent solution inwhich consumers may examine several available booking slots and select their chosen day and time. This minimizes physical waiting time, saving users time while also improving the appointment process' efficiency. This program allows the doctor to alert his own timetable. Hospitals can conveniently manage their registration and appointment processes, as well as track patient flow to doctors.



Appointmented List

Home » View Appointments

Patient Name	Gender	Age	Detected_Disease	Consulting_Doctor	Appointment Date	Appointment Time	Chat	Status
Bhagya	Female	22	Heart attack	balaji	2024-03-21	18:00	Chat	Rating

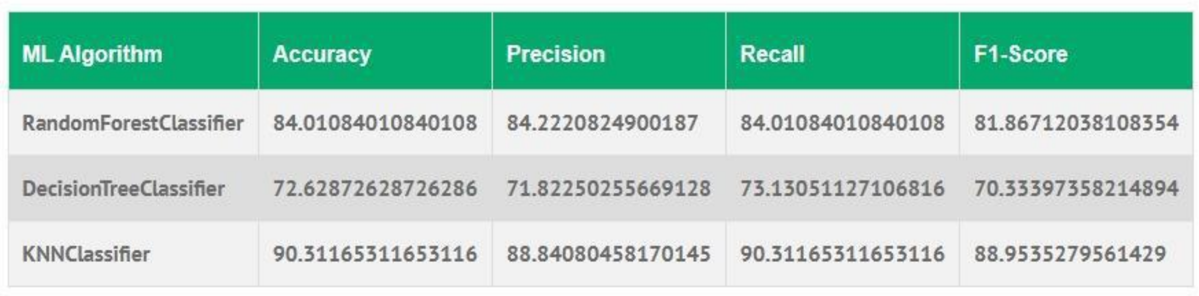
2024 © AI-Driven Disease Insights, Doctor Referral & Appointment Utility

Fig.4.1.3 : Patient’s Appointment List

It shows the appointments booked by patient.Here, patient can login ,check their appointment and give rating to doctor depending on his experience with medication.

4.2 Performance Metrics

Here is the accuracy and precision of all algorithms (KNN,Random Forest and Decision tree)



ML Algorithm	Accuracy	Precision	Recall	F1-Score
RandomForestClassifier	84.01084010840108	84.2220824900187	84.01084010840108	81.86712038108354
DecisionTreeClassifier	72.62872628726286	71.82250255669128	73.13051127106816	70.33397358214894
KNNClassifier	90.31165311653116	88.84080458170145	90.31165311653116	88.9535279561429

Fig.4.2.1. : Accuracy

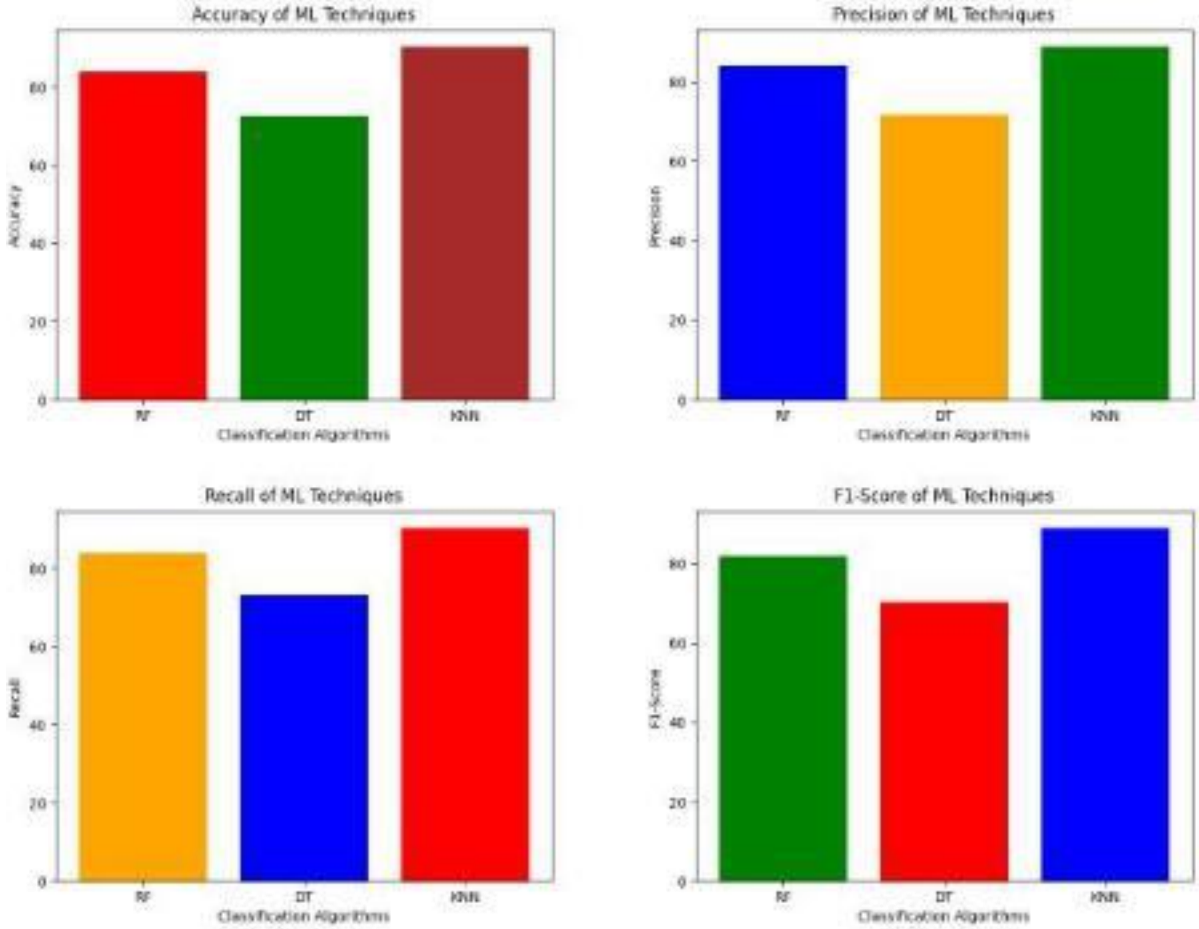


Fig.4.2.2 : Graphs

CHAPTER 5

CONCLUSION

CHAPTER 5

CONCLUSION

In conclusion, the project revolving around the development and implementation of an AI-Driven Disease Insights, Doctor Referral & Appointment Utility platform holds great promise for revolutionizing the healthcare landscape. By leveraging artificial intelligence, this platform aims to empower patients with valuable disease insights, streamline the process of finding healthcare providers, and simplify appointment scheduling, ultimately enhancing the patient experience and improving healthcare efficiency.

While there are merits to such a project, including improved patient convenience, access to healthcare, and data-driven insights, there are also important challenges and limitations to consider. These encompass issues related to data privacy, the accuracy of AI algorithms, regulatory compliance, and the need for user trust and adoption. Addressing these challenges is essential for the successful implementation of such a project.In a rapidly evolving healthcare technology landscape, the ability to adapt to regulatory changes, maintain data security, and continually improve the platform's features and user experience will be key to its long-term success. Overall, the AI-Driven Disease Insights, Doctor Referral & Appointment Utility project represents a significant step towards the future of healthcare, where technology plays a pivotal role in enhancing patient care and making healthcare services more accessible and efficient.

REFERENCES

REFERENCES

[1] Vamshi and Sai developed “Prediction of disease using ML” in International Journal for Research in Applied Science &Engineering Technology (IJRASET) Volume 10 Issue VI June 2022

[2] http:practo.com developed in 2008

[3] http:sminq.com developed in 2015

[4] Venkatesh and Hema Gallani, Gaytri Aasija researched on online appointment booking system.ISSN: 2248-9622, Vol. 12, Issue 4, (Series-III) April 202

[5] “Disease Prediction using Machine Learning Algorithms” Sneha Grampurohit, Chetan Sagarnal 2020 International Conference for Emerging Technology

(INCET) Belgaum, India. Jun 5-7, 2020

GitHub Link

1. attell/AI-Driven-Disease-Insights-Doctor-Referral-and-Appointment-utility-Phase-2 (github.com)

Volume: 12 Issue: III Month of publication: March 2024

DOI: https://doi.org/10.22214/ijraset.2024.59528

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue III Mar 2024- Available at www.ijraset.com

AI-Driven Disease Insights-Doctor Referral and Appointment Utility

A. Bhagya Sree[1], A. Bhanu Prasad[2], Ruheena Naaz[3], Mr. B.K. Chinna Maddileti[4]
[1, 2, 3]UG Student, Department of CSE, CMR College of Engineering & Technology, Hyderabad, Telangana
[4]Assistant Professor, Department of CSE, CMR College of Engineering & Technology, Hyderabad, Telangana

Abstract: Many deadly diseases are preventable and curable if addressed early. However, because the early signs of an illness are mild, people often overlook them. As a result, it is always recommended to get a regular body check-up in order to avoid future health difficulties. Early illness prediction has become the most critical work, and accurate disease prediction is the most difficult task. We presented a web application to help people detect diseases based on their symptoms. There are numerous applications for online consultations with doctors, however there are less applications for prediction of the disease. There are systems that can only predict one health problems. however our approach can predict forty-one common diseases. For projections, we require symptom data. Using the Kaggle platform's disease symptom dataset, we discover a massive quantity of information to analyze in our system to ensure correct prediction. The system uses random forest, decision tree, and knn algorithms to identify diseases. After predicting disease, the system will offer a list of specialists who are experts in that particular condition. Patients can schedule an appointment with the doctors at their desired time and day. Patients can also communicate with a doctor to receive consultations.

Keywords: Machine Learning, Diseases, Symptoms, Decision tree, Random forest tree, KNN.

I. INTRODUCTION

The use of data mining is increasingly important in the healthcare and medical sectors. Effective processing methods can extract significant information from enormous databases, enabling medical practitioners to make early decisions and improve health services. Machine Learning is a promising method for disease prediction and diagnosis. This paper proposes to predict diseases using patient symptoms . Machine learning algorithms, such as KNN, Decision Tree, and Random Forest, are used to forecast diseases. Python is used to complete the implementation. Ignoring diseases and health issues, such as malaria, impetigo, diabetes, migraine, jaundice, and chickenpox, can have serious consequences for one's health and even death. The healthcare business can improve decision-making by "mining" its enormous database to identify hidden patterns and linkages. Data processing methods such as Decision Trees, Random Forests, and Naive Bayes can address the current situation. We created an automated approach to retrieve disease-related knowledge from a historical database using algorithmic rules.Our approach effortlessly integrates cutting-edge AI technology and healthcare access. This allows users to acquire deep insights into their health conditions through symptom analysis, receive tailored doctor referrals, and easily schedule appointments, transforming the way we approach healthcare in a user-friendly, efficient, and accessible manner.

II. RELATED WORK

In the quest for innovation and efficiency, modern projects frequently rely on existing solutions as fundamental building blocks for development. This approach not only recognizes the expertise and advancements of those who came before us but also nurtures a collaborative ecosystem where ideas can evolve and confront new challenges. In our project, we wholeheartedly embrace this ethos, conscientiously integrating elements from existing solutions to enrich our endeavor. These existing solutions serve as guiding lights, offering insights and frameworks that shape the direction of our project.

1) *Disease Prediction using ML on users Symptoms:* Vamshi Krishna and Sai Nath, in 2022, delved into disease detection using machine learning, particularly targeting the healthcare industry. Their objective was to predict patients' using symptoms.By K-Nearest Neighbors (KNN) algorithm. Through meticulous model training and symptom analysis, they achieved an impressive 87% accuracy rate in disease prediction..
2) *Online Doctor Appointment System:* Venkatesh Rallapalli and Dipti Menghani in 2016 developed an Online Doctor Appointment System, a web-based platform enabling patients to choose doctors according to their medical requirements and view their profiles. This system primarily facilitates appointment scheduling with different doctors, without incorporating disease prediction functionality.

3) *Practo which Provides Doctor Search and AppointmentBooking:* Shashank ND and Abhinav Lal founded Practo in 2008. Practo provides medical services online. It's all about your healthcare needs. With Practo, you can find the right doctors, book appointments, keep track of your health records, and even get medical advice through telemedicine.
4) *Sminq App:* Shachin Bharadwaj and Santhosh Nagarajan launched the Sminq app in 2015. It lets users check live queue status for doctor appointments and offers real- time updates on trending places worldwide. Sminq focuses on managing queues and cutting wait times for different services. Users can join virtual queues and get instant updates on their queue status through the app.

III. PROPOSED SYSTEM

Our solution is a web application that allows users to input their symptoms. Once a disease is identified, the system suggests specialized doctors and facilitates easy appointment booking and real-time consultations. It employ decision tree, random forest,and KNN algorithms for disease identification. Users can conveniently book appointments if they suspect they have any of the identified diseases.

In Fig 1 administrators have the capability to oversee and manage details regarding doctors, patients, and diseases, as well as receive valuable patient feedback. Doctors can directly communicate with patient's health status. The system efficiently handles all aspects of doctor-related information, including schedules, fees, and appointment management. Users have access to video consultations and can engage in chat sessions with doctors

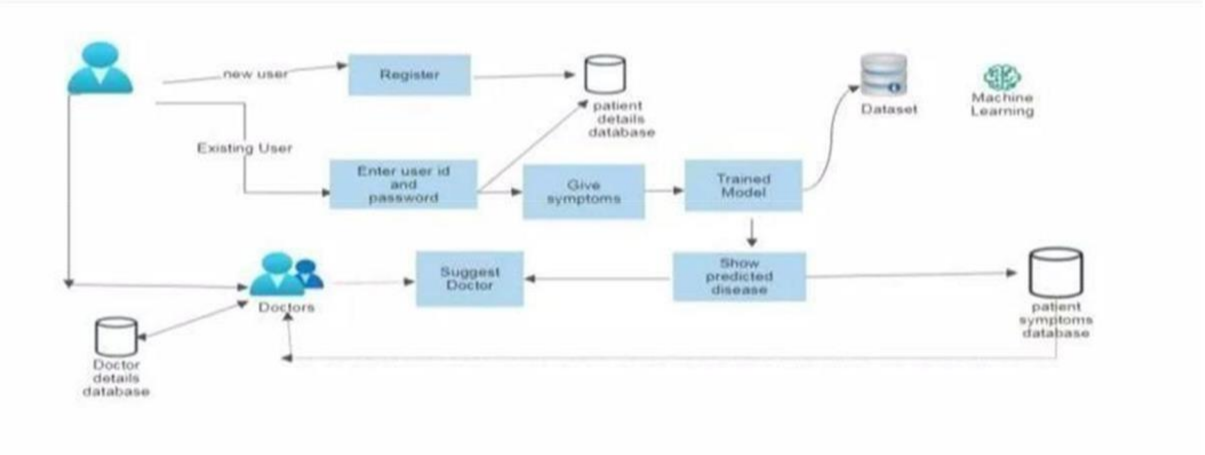


Fig 1. System Design

IV. METHODS AND EXPERIMENTAL DETAILS

A. Decision Tree

It partitions a set of data repeatedly into subgroups based on the numerical values of the input features. At each node, the algorithm chooses the optimum feature and threshold to partition the data, seeking to optimize homogeneity within each subset. This splitting process continues until a stopping requirement is satisfied, such as achieving a maximum depth or a minimum quantity of samples per node. The resulting tree structure contains a set of decision rules forecast new data points by traversing the tree from the root to the appropriate leaf node. In classification tasks, each leaf node provides a class label, but in regression tasks it depicts a predicted value.

B. KNN

The K-Nearest Neighbors (KNN) algorithm operates by data points including classifications or regression values during the training phase.When presented with new data for classification or prediction, KNN calculates the distances between the input data point and all other data points in the training set, typically using the Euclidean distance metric. By selecting a predefined number of nearest neighbours (K), KNN identifies the K closest data points to the input and determines their classifications or regression values. Finally, it assigns highest class label or averages the regression values among the K neighbours to predict value for the input data point, making it a versatile and intuitive algorithm for various machine learning tasks.

C. Random Forest

It is an ensemble learning technique that combines the predictions of many different decision trees to improve accuracy and robustness in classification and regression. Random Forest teaches through developing a multiple decision trees at each node using bootstrap sampling and random feature selection. Each tree separately predicts the target variable, and output is combining the various tree forecasts using majority voting (for classification) or averaging (for regression).

D. Methodology

In our model development, we proceed with the expectation that users have a comprehensive understanding of their symptoms.Data preprocessing is the name of the data mining approach that converts or encodes the data into a format that the algorithm can understand with ease. The preparation methods used in the work that is being presented include: Data Cleaning: Data is cleaned up by procedures like adding missing values, which eliminates data discrepancies. Data reduction: We have also eliminated orthographic errors from the dataset and substituted the true value for any missing data.Three algorithms are used by this system to try and forecast the diseases:

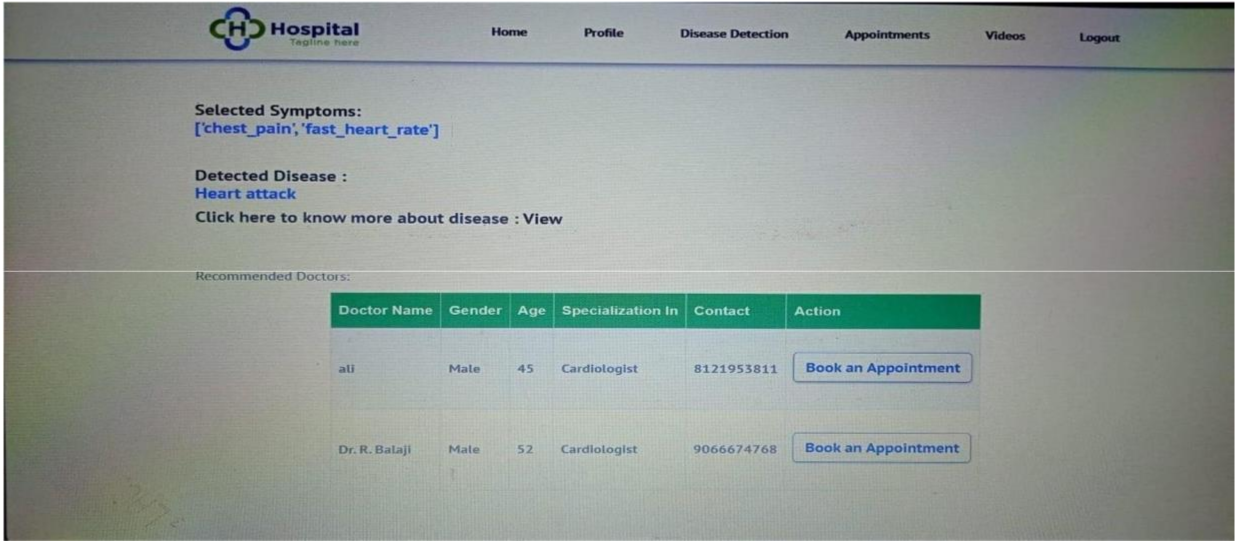
1) KNN Classifier
2) Random Forest Classifier
3) Decision Tree Classifier

A research that examines each algorithm's performance inside the database under consideration is offered.When the user provides symptoms as input to the model, the system processes those symptoms in accordance with the rule set created, resulting in classifications and the prediction of the most likely disease.

V. RESULTS AND DISCUSSIONS

By entering symptoms, we achieve accurate disease predictions. We gathered a lot of information from the disease symptom dataset on Kaggle, which helps our system work better. There will be list of symptoms in which patients can select their symptoms and check for their diseases. In this project we used Pycharm for the execution of our project and SQL for storing the database and SQLyog for checking the data updates in the project. After running in pycharm the admin has to login, he can approve new doctors registrations. Patient has to login or register then can take actions as to know disease or appointment list. Doctor has to register for first time andthen approve patients request. Using this project users can efficiently know about predicted disease at the same time can book appointments. In fig2.patient is login and can select symptoms using ml algorithms disease will be predicted and book appointment with the list of specialized doctors. Then book appointment .this has to be approved by doctor.

Other applications focusing on symptom-based disease prediction often fall short in terms of effectiveness, offering only a limited number of diseases with relatively low accuracy and lacking a convenient appointment booking system. In contrast, platforms like Practo aim to enhance overall healthcare experiences by providing comprehensive online medical services and solutions for individuals seeking better self-care.



Selected Symptoms:
['chest_pain', 'fast_heart_rate']

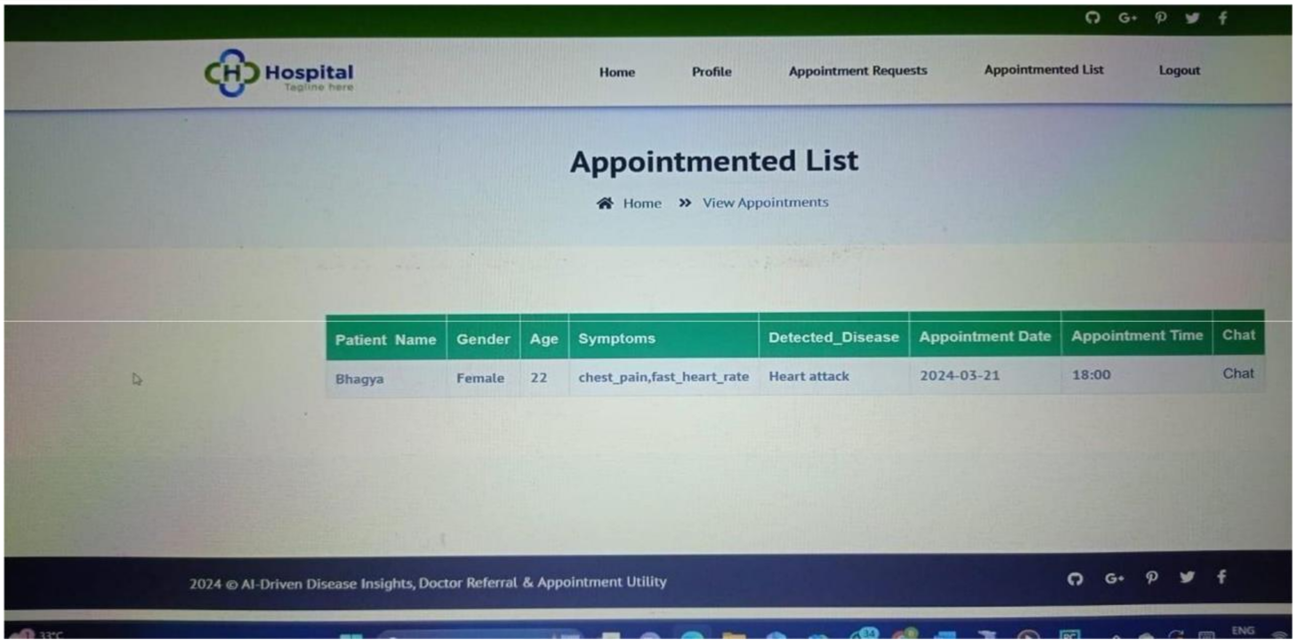
Detected Disease :
Heart attack

Click here to know more about disease : View

Recommended Doctors:

Doctor Name	Gender	Age	Specialization In	Contact	Action
ali	Male	45	Cardiologist	8121953811	Book an Appointment
Dr. R. Balaji	Male	52	Cardiologist	9066674768	Book an Appointment

Fig 2.Patient Interface



Appointmented List

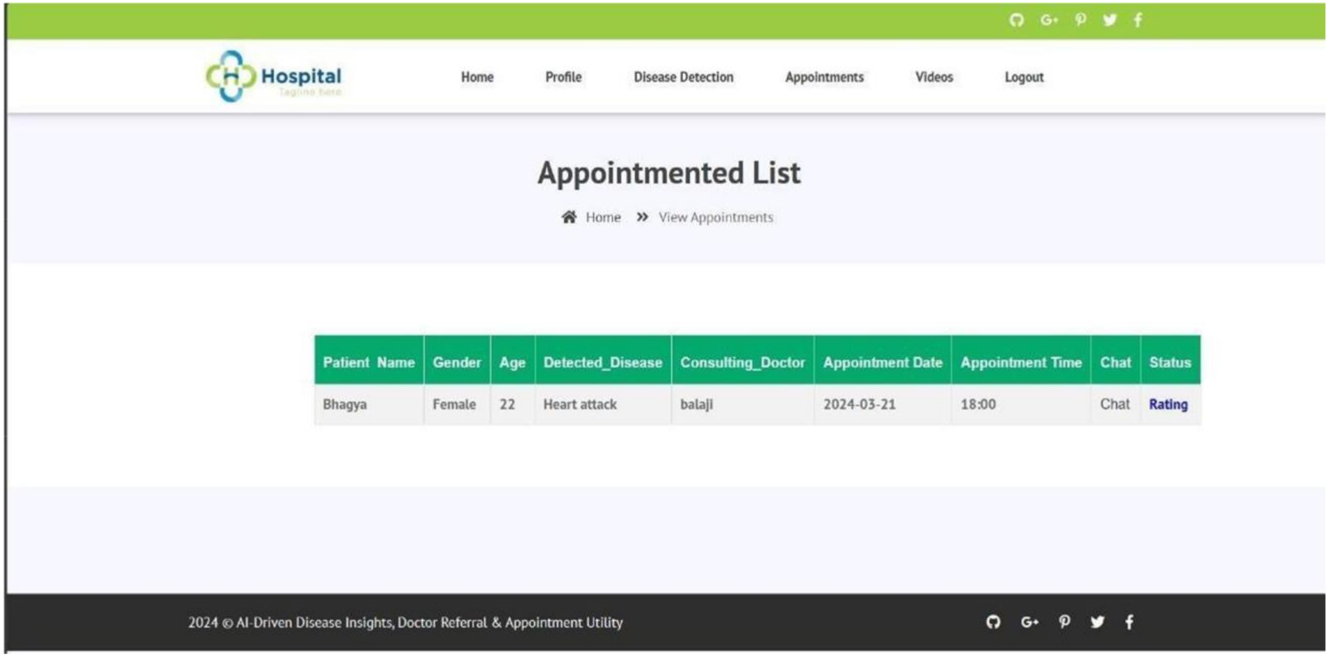
Home » View Appointments

Patient Name	Gender	Age	Symptoms	Detected_Disease	Appointment Date	Appointment Time	Chat
Bhagya	Female	22	chest_pain,fast_heart_rate	Heart attack	2024-03-21	18:00	Chat

2024 © AI-Driven Disease Insights, Doctor Referral & Appointment Utility

Fig 3.Doctor Interface

In fig.3 shows the patient's appointment acceptance by doctor. It is the interface for managing and arranging appointments depending on the doctors preferences or needs. Using appointments list option doctors check for their availability to provide the desired time to patients depending on their medical needs. After checking and accepting the patient appointment the above figure is depicted as output.As a result,project helps consumers may examine several available booking slots and select their chosen day and time. This minimizes physical waiting time, saving users time while also improving the appointment process' efficiency. This program allows the doctor to alert his own timetable. Hospitals can conveniently manage their registration and appointment processes, as well as track patient flow to doctors.



Appointmented List

Home » View Appointments

Patient Name	Gender	Age	Detected_Disease	Consulting_Doctor	Appointment Date	Appointment Time	Chat	Status
Bhagya	Female	22	Heart attack	balaji	2024-03-21	18:00	Chat	Rating

2024 © AI-Driven Disease Insights, Doctor Referral & Appointment Utility

Fig 4.Patient Appointment Interface

In fig.4 It shows the appointments booked by patient.Here, patient can login and check their appointment and give rating to doctor depending on his experience with medication

VI. CONCLUSION

We'd like to demonstrate how vital this initiative is in everyone's daily life, who rely on these algorithms to anticipate the needs of patients problems based on their basic information and symptoms. It is frequently helpful for the health business to enlighten the user because it has become such a significant part intreating patients illnesses is also beneficial for the user if he or she does not want to travel to the hospital or other clinics, because the user can learn about the disease . This method may enhance the health sector by allowing patients to enter symptoms and other relevant information

The proposed system will be more effective to find doctors nearby with disease speciality for patients rather than finding them manually or going clinic to clinic.

This system is not intelligent enough to provide a 100% precise disease hence further we provide the user with nearby doctors or hospitals or any specialist doctor regarding that particular disease. Another feature that will predict the diseases for the users or patients who provide the symptoms as input to ease the process of doctor recommendation directly with face to face appointment with doctors.Another feature which can be added in future is to provide a user interface friendly to the user with the help of AI chatbots and image processing. We will try to provide a user based report with respect to its past information such as predicted diseases of patient/user and the doctors appointed.

This system providing a seamless experience for everyone involved. This lowers tiredness and frustration while also providing an easy way to schedule appointments in today's society.

REFERENCES

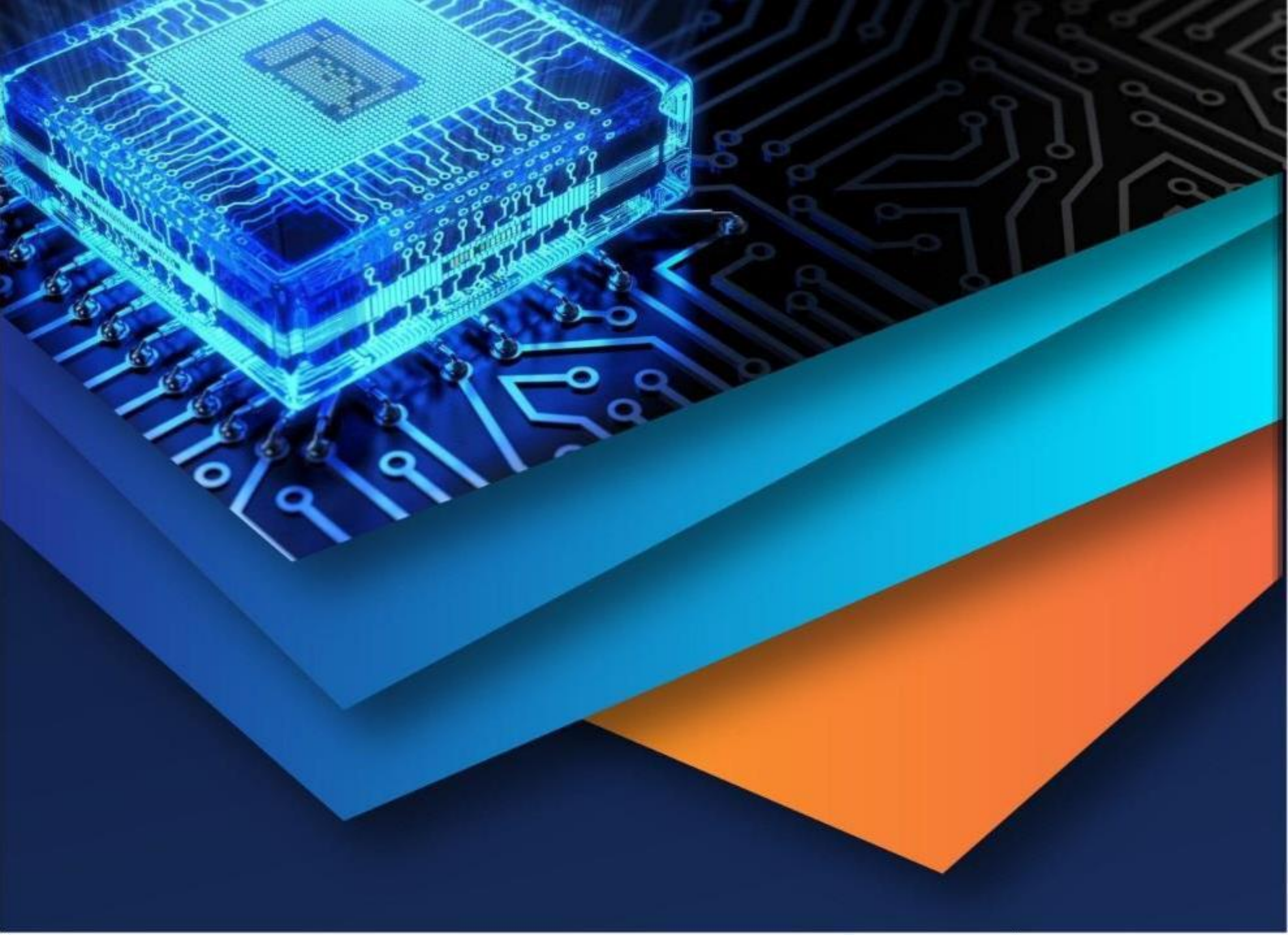
[1] Vamshi and Sai developed “Prediction of disease using ML” in International Journal for Research in AppliedScience & Engineering Technology (IJRASET) Volume 10 Issue VI June 2022

[2] http:practo.com developed in 2008

[3] http:sminq.com developed in 2015

[4] Venkatesh and Hema Gallani, Gaytri Aasija researched on online appointment booking system.ISSN: 2248-9622, Vol. 12, Issue 4, (Series-III) April 202

[5] “Disease Prediction using Machine Learning Algorithms” Sneha Grampurohit, Chetan Sagarnal 2020 International Conference for Emerging Technology (INCET) Belgaum, India. Jun 5-7, 2020



ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

Certificate

It is here by certified that the paper ID : IJRASET59528, entitled

AI-Driven Disease Insights-Doctor Referral and Appointment Utility

by

A. Bhagya Sree

after review is found suitable and has been published in
Volume 12, Issue III, March 2024
in
International Journal for Research in Applied Science &
Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors

Editor in Chief, **IJRASET**

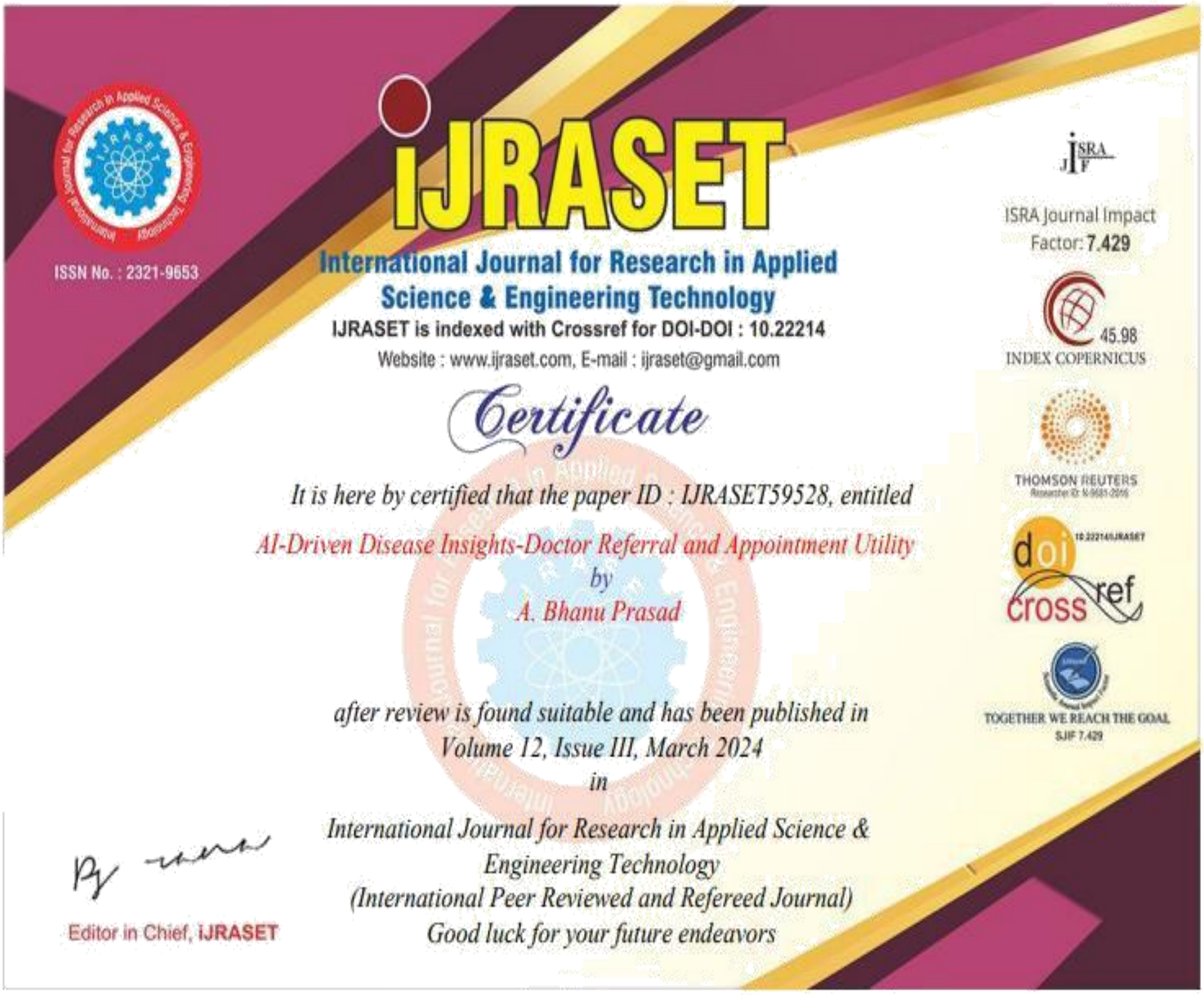
ISRA Journal Impact Factor: **7.429**

45.98
INDEX COPERNICUS

THOMSON REUTERS
Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL
SJIF 7.429



ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS

10.22214/IJRASET

TOGETHER WE REACH THE GOAL

SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET59528, entitled

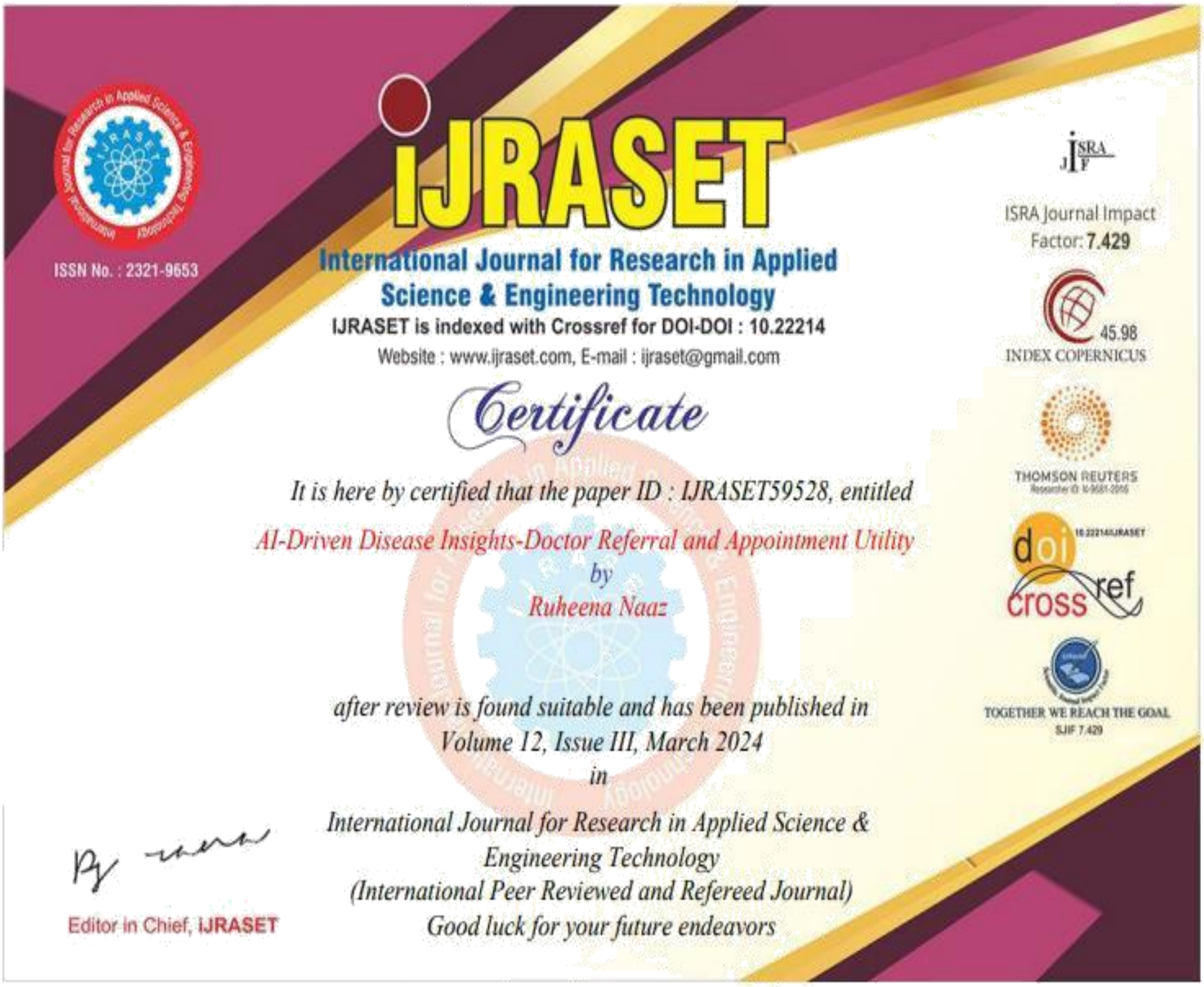
AI-Driven Disease Insights-Doctor Referral and Appointment Utility

by

A. Bhanu Prasad

after review is found suitable and has been published in
Volume 12, Issue III, March 2024
in
International Journal for Research in Applied Science &
Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors

Editor in Chief, IJRASET



ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS

10.22214/IJRASET

TOGETHER WE REACH THE GOAL

SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET59528, entitled

AI-Driven Disease Insights-Doctor Referral and Appointment Utility

by

Ruheena Naaz

after review is found suitable and has been published in

Volume 12, Issue III, March 2024

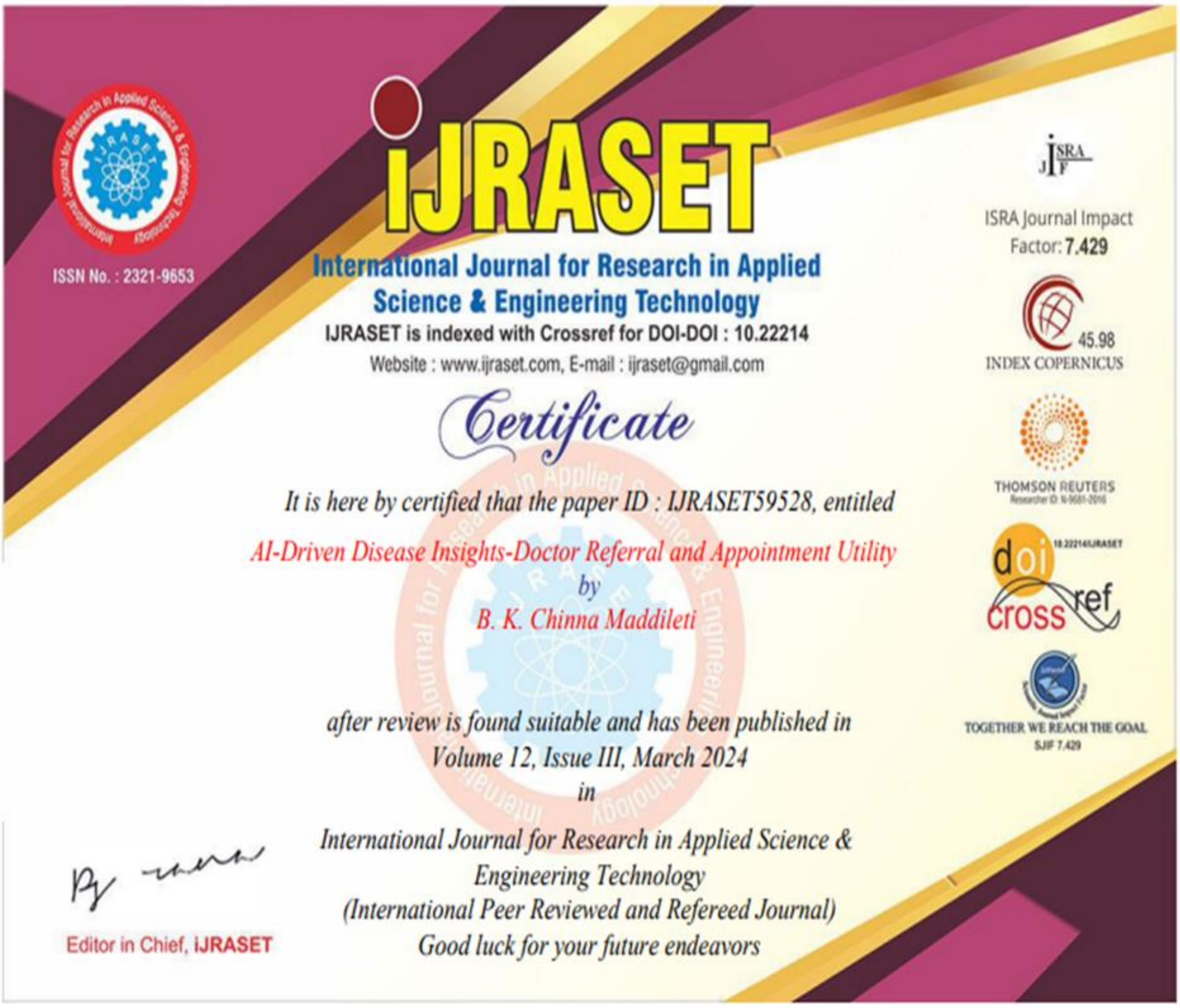
in

International Journal for Research in Applied Science & Engineering Technology

(International Peer Reviewed and Refereed Journal)

Good luck for your future endeavors

Editor in Chief, IJRASET



ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS Researcher ID: N-9681-2016

10.22214/IJRASET

TOGETHER WE REACH THE GOAL SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET59528, entitled

AI-Driven Disease Insights-Doctor Referral and Appointment Utility

by

B. K. Chinna Maddileti

after review is found suitable and has been published in
Volume 12, Issue III, March 2024
in
International Journal for Research in Applied Science &
Engineering Technology
(International Peer Reviewed and Refereed Journal)
Good luck for your future endeavors

Editor in Chief, IJRASET