AN AI BASED CHATBOT MODEL FOR INFECTIONS DIESASE PREDICTION

A Project Report Submitted to

In partial fulfillment of requirements for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

By

T.CHANDU

(Regd.No.21F11F0060)

Under the esteemed Guidance of

Mr.V.KASTHURAIAH, MCA

Asst. Professor,HOD

Narayana Engineering College (Autonomous), Gudur

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

NARAYANA ENGINEERING COLLEGE :: GUDUR (AUTONOMOUS)

(2021-2023)

NARAYANA ENGINEERING COLLEGE :: GUDUR (AUTONOMOUS)

BONAFIDE CERTIFICATE

This is to certify that the project entitled "***AN AI BASED CHATBOT MODEL FOR INFECTIONS DISEASE PREDICTION***" that is being submitted by **Mr.T.CHANDU (21F11F0060)** in partial fulfillment of the requirements for the award of degree of **MASTER OF COMPUTER APPLICATIONS** in to Narayana Engineering college, GUDUR is recorded to be the Bonafide work carried out by her under my guidance and supervision.

Signature of project guide

MR. V. KASTHURAIAH, MCA

Asst. professor,

Department of MCA,

NEC, Gudur

Signature of HOD

MR. V. KASTHURAIAH, MCA

Department of MCA,

NEC,Gudur

INTERNAL EXAMINER

EXTERNALEXAMINER

The Project Viva-Voice held on…….

ACKNOWLEDGMENT

I am extremely thankful to **Dr. P. Narayana Garu, Founder** of **Narayana Engineering College, GUDUR** for providing the infrastructural facilities to work in, without this work would not have been possible.

I express my sincere thanks to **Mr. Y. Vinay Kumar Garu**, **Managing Secretary** of our college for granting the permission to carry out this project work.

I would like to express my deep sense of gratitude to the principal **Dr. Ch. V. S. Ravi Prasad Garu, Principal, Narayana Engineering College, GUDUR** for his continuous effort in creating a competitive environment in our college and encouraging throughout this course.

I would like to express my deep sense of gratitude to the principal **Dr. K .Vishwaksena Reddy ,Vice Principal, Narayana Engineering College, GUDUR** for his continuous effort in creating a competitive environment in our college and encouraging throughout this course.

I would like to convey my heartfelt thanks to **Mr. V. Kasthuraiah Garu, HOD of MCA** for giving the opportunity to embark upon this topic and for his continued encouragement throughout the preparation of this presentation.

I would like to thank sincerely to my project Coordinator **Mr. A . Sanjeev kumar** and my guide **Mr. V. Kasthuraiah garu**, **Assistant Professor** for his invaluable guidance, constant assistance, support, endurance and constructive suggestions for the betterment of the project.

I also wish to thank all the **Staff Members of the Department of Master of Computer Applications** for helping us directly or indirectly in completing this work successfully.

Finally, I thankful to **My Parents and Friends** for their continued moral and material support throughout the course and in helping us to finalize the report.

T. CHANDU

(21F11F0060)

ORGANIZATION PROFILE

Supreme Info Solutions is technology solutions company that has focus in the 'e-Business' space. Founded and steered by professionals from India, Supreme Info Solutions eBusiness understanding is backed by its domain expertise in industries covering Banking and

Financial Services, Chemicals, Pharmaceuticals, Construction, Retail, Electronics, Mobile, Food & Agriculture, and Automobiles. Supreme Info Solutions has leveraged its domain expertise to create industry vertical solutions in the digital economy.

The company has over the years evolved into an end –to – end IT solutions provider with over 50 employees and having robust sales and profits that are showing sustainable growth. Supreme Info Solutions is currently one among the best IT Solutions provider for large corporate and government establishments. Supreme Info Solutions current Clients are spread over India and abroad.

Supreme Info Solutions has executed large and complex projects based on the web based technologies such as Java based technology platform & .NET based technology platform. Supreme Info Solutions has highly qualified resources in web technologies that have gained immensely through working on large complex projects. These resources are skilled in the all the required technologies that are utilized while executing a web based project and have over 300 man years of experience in developing & implementing web based solutions for clients that are spread over the world.

PRODUCTS AND SERVICES:

A. Hospital Management Software

Supreme info Solutions is one the solution provider for Hospital Management Software in the health care domain. We provide cloud based and local server based solutions to hospitals to automate their key processes - clinical, financial, and operational and management processes. As a result, it quits easy facilitating day to day operations with a strong reporting system.

Supreme Info Solutions consists of a team of experienced and dedicated professionals that each carries extensive experience in their own field. Their passion to implement highly successful strategies, provide affordable health care solutions, and introduce innovative technologies inspired them to automate the entire system.

This software helps to manage the workflow processes among different departments within an enterprise and coordinate departmentalize activities in a tremendous way. It is capable of both retrieving and disseminating patient specific data (clinical & financial).The main aim of Solvent India HRS is to achieve the better Patient Care & Safety.

B. Mobile Application Development

Are you looking for great mobile app product or branding your website with mobile interface? If answer is yes, Then Solvent India here in the market to make your dreams come true. Get featured industry application to benefit your company with high audience appearance and brand value.

With the introduction of Smartphone, the need for the Mobile Application Development has also increased rapidly in today's world. Mobile apps have become necessity in today's era. And when we talk about mobile application development, we focus on various platforms like iPhone, Android and Windows.

We at Supreme Info experts in Consulting and assisting you in architecture, development, deployment and maintaining your open standard infrastructure solutions for successful implementation in your organization. The app is an Engineering and Terminology reference app that assists engineers in the selection of high quality fastener products.

We help you to Select an application that provides easy access to product details, dimensional and seating torque values for standard applications with the addition of part number reference and contact details.

Supreme Info Solutions has built in-house capabilities and expertise in developing iPhone applications. Our expert iPhone application developers/programmers transform your app idea into integrated app design with unique application development experience.

We are path breaking due to our unfolded quality, affordable cost, project deliverance on time, client approach and sheltered development with best environment. Solvent India

thrives upon its watchwords of ‘Quality beyond contract’ and ensures it in every sphere of its operations on any mobile platform.

iPad Development

iPad application developer needs more technical knowledge than any other smart phones. Based on its years of experience and in-depth knowledge in iPad application development, we are well-positioned to undertake app development on any iOS device. Solvent India gives you a platform to raise any enquiry and provide you a robust system for your mobile device. We have developed various iPad applications in different sectors like travel, restaurant, , health care, news, weather, GPS, marketing, job portal, business directory, real estate, social network, multimedia, daily utility software, touch screen applications, and so on.

Android Development

Android, a software platform and operating system for mobile devices developed by Google and the Open Handset Alliance, is the new craze of the Mobile application development industry. We engaged in the field of Mobile Application Development for quite some time back and being expert in all the leading platforms, we quickly jump into the development of Android Application development.

We have expertise that can develop custom android mobile based applications for your business.

C. Software Development:

Here at Supreme Info, we focus on designing the ERP Solutions for your business. With a resourceful designing panel, Solvent India extends its services of creating dominant and vibrant ERP solutions in different domain to cater the needs of big as well as small enterprises.

OUR ERP Solutions help you in forecasting and scheduling the resources of all kinds of businesses.

Our organization particularly designs the ERP software to supply the robust solutions to a business about mounting its output in all the divisions through appropriate reserve management.

The model of most ERP (Enterprise Resource Planning) systems can be described in the following way: the data on the company's activity gets into the single database based on which the program makes various reports, forecasts, diagrams, etc. The ERP system simplifies and speeds up the performance of routine work, reduces the risk of misprints during the data entry, increases productivity in general. The main objective is to reduce paper work and to automate the entire business system; In return you will get work efficiency at your work place.

Unfortunately, the ERP software development doesn't meet customer's needs all the time. The reason can be the following: the ERP system development was trusted to low qualified specialists, as a result, the developed system can have big defects and failures. Very often it's so complicated that the company has to get rid of its usage.

Main reasons for unsuccessful ERP system implementation:

Mismatch between ERP Solutions & Goals:-The system was designed without taking into account the company's long term goals. The wrong designing can lead to big financial expenses as well as the necessity to redesign the system in the nearest future. During the ERP software development it's necessary to foresee the company's activities in the next three years.

Excessive Business Process Re-engineering. Re-engineering is creation of principally new effective business processes in management that the company didn't have before. There are two extremes in this situation: some companies agree to re-engineer of all company's business processes and their subordination to the requirements of the chosen system or insist on keeping the existing way of performance and the principal reconstruction of the chosen system. Both variants have negative points to consider. The best solution is to find the correct normal balance between business process re-engineering and system modification.

Wrong assessment of Infrastructure tools:-The development of the ERP system requires big expenses for general automation. It includes: computers, servers, licenses, network equipment, consultation services, etc.

Thus, it's necessary to correlate the expenses for the automation of various business processes with the final results of the project, otherwise, there's a risk the expenses won't be paid off. In order not to have excessive expenses, the company needs to analyze the economic effectiveness of each element of the ERP system already at the stage of development of its concept.

D. Web Development

Supreme Info Solutions excels in the designing field and provide the best website development and promotions. We have been developing website that do real business. And we offer our services in an affordable cost.

Firstly, we study what exactly the client need. After detailed analysis a project line is made which is given to the client for their approval. Secondly, if the concept is approved, the next step comprises of the issues that deals with layouts, templates, graphics, user friendliness, targeting the audience and the access speed.

Then in total using everything in mind a prototype is made that is again send to the client for the approval. Soul of the website is then made. Keeping in mind that each and every site should act as the sales tool as well as the purchasing tool.

Content Management System

Content Management System (CMS) Made Simple provides a fast and easy way to create a professional web site and manage its content, whether it is for a small business or a multinational corporation.

E. IT Consulting

Supreme Info Solutions does business analysis to translate your business ideas into technical requirements. We design system architecture to find the right technological strategy, minimize the risks of future changes, speed up the development process and scale system

with minimal effort. We deliver technology consulting services to companies across multiple domain areas. Different platform concern to our Technology consulting is given below:-

- Business Process Analysis and Formalization
- Design Business strategies
- IT Infrastructure Audit
- IT Strategy Development
- Design Frame work/layout for multiple business lines
- Software Architecture Review and Planning
- Application Security Consulting
- Project Requirements Definition and Design
- Provide optimal solutions on different Technology platform
- How we help our clients for long-term benefits with Smart investments

In this competitive arena, any Businesses need to invest in tools/Technologies that can create robust structures, processes and strategies – strong enough to react to these changes accurately without compromising on quality or cost.

Supreme Info Solutions offers complete range of end-to-end IT consulting services that are designed to fulfil the needs of businesses of every size. We help our clients to attain strategic objectives, increase productivity, maximize profitability and streamline processes. We have talented pool at our work place and we access consultancy services along with onsite/offsite and offshore model. Our main objective is to provide low risk solutions, helping clients in reducing infrastructure overheads.

Supreme Info provides a concrete business solution for any internal or external partners to work in smooth functional flow. Our IT consulting services will help you achieve goals and allow you to make your future plan under predictable circumstances.

Our strength:

- Our Consulting services helps you speed up your business
- To convert your business ideas into action plan
- To make innovation at faster pace
- Maximize returns on your investment

- Timely delivery
- Optimal solutions provider
- Spectrum of advanced technologies
- Talented Poor on different technological skill set

CONTENTS OF INDEX

SEEKHA GROUP

SUPREME®
INFO SOLUTIONS

Date: 23rd Aug 2023,

Nellore.

TO WHOM IT MAY CONCERN

This is to certify that Mr. **THOTAKURA CHANDU** has done his internship in Python Technology at **SUPREME INFO SOLUTIONS**, Nellore, from **1st May 2023 to 16th Aug 2023.**

He has worked on a project titled **"AI BASED MEDICAL CHATBOT FOR INFECTION DISEASE PREDICTION"**. This project was aimed at design Medical Chatbot to predict Disease. As part of the project, he has designed interactive web pages, database design, Python programming.

During his internship he has demonstrated his skills with self-motivation to learn new skills. His performance exceeded our expectations and he was able to complete the project on time.

We wish him all the best for his upcoming career.

Supreme Info Solutions

23/08/23

(S SRIKANTH BABU)
Managing Director

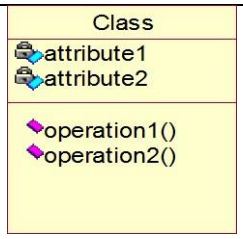
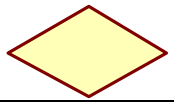
Supreme Info Solutions Dt..... Nellore

Door No: 23-218,2nd Floor, Vasantha Nilayam, Raavi Chettu Center, Fathekan pet, Nellore – 524 003.Email Id: supremeinfosolutions@gmail.com, Cell: 9063 64 3665.

ABSTRACT

The purpose of this paper is to show concisely how we can promote chatbots in the medical sector and cure infectious diseases. We can create awareness through the users and the users can get proper medical solutions to prevent disease. We created a preliminary training model and a study report to improve human interaction in databases in 2021. Through natural language processing, we describe the human behaviors and characteristics of the chatbot. In this paper, we propose an AI Chatbot interaction and prediction model using a deep feed forward multilayer perceptron. Our analysis discovered a gap in knowledge about theoretical guidelines and practical recommendations for creating AI chatbots for lifestyle improvement programs. A brief comparison of our proposed model concerning the time complexity and accuracy of testing is also discussed in this paper. In our work, the loss is a minimum of 0.1232 and the highest accuracy is 94.32%. This study describes the functionalities and possible applications of medical chatbots and explores the accompanying challenges posed by the use of these emerging technologies during such health crises mainly posed by pandemics. We believe that our findings will help researchers get a better understanding of the layout and applications of these revolutionary technologies, which will be required for continuous improvement in medical Chabot functionality and will be useful in avoiding COVID-19.

SYMBOLS

Symbol Name	Symbol	Description
Class		Class represents a collection of similar grouped together.
Associations		Association represent static relationship between classes.
Aggregation		Aggregation is a form of association; it aggregates several classes into single one.
Usecase		A usecase is an interaction between the system and the external environment.
Actor		Actors are the users of a system.
Decision		Diamond indicates the decision.
Communication		It is the communication between various usecases.
State		It represents the state of a process, each state goes through various flows.
Initial State		It represents the initial state of the project.

Final State		It represents the final state of the project.

LIST OF ABBREVATIONS

S.NO.	ABBREVATION	DESCRIPTION
1.	**WHO**	World Health Organization
2.	**SDLC**	System Development Life Cycle
3.	**SRS**	Software Requirement Specifications
4.	**ERD**	Entity Relationship Diagram
5.	**GUI**	Graphical User Interface
6.	**LSTM**	Long Short-Term Memory
7.	**UML**	Unified Modelling Language
8.	**NLP**	Natural language processing

9.	**LSA**	Latent Semantic Analysis
10.	**AIML**	Artificial Intelligence Markup Language
11.	**K-NN**	K-nearest neighbor
12.	**NB**	Naïve Bayesian

INTRODUCTION

1.1 Motivation

Covid-19 is an ailment because of the SARS-CoV-2 virus. `World Health Organization (WHO)' has been declared a pandemic on March 11, 2020. Around 15 million human beings have been affected worldwide by more than one million deaths by covid-19. From the start, the affordability and sustainability of oxygen have been a problem in poor and underdeveloped countries. Oxygen is a very important medicine for treating hospitalized patients with Covid-19. According to PATH, the organization that works with global institutions and businesses to tackle health problems, the demand for Oxygen cylinders has been growing between 6%-8% each daily India. When someone gets severe Covid19, the oxygen levels in the body get low. Patients suffered from enormous fevers, coughs, and lost sense of taste and smell, all of which became problematic when the SARS-CoV-2 virus invaded them. As a result, to prevent such massive problems associated with COVID-19 and facilitate a quick cure method, we have developed a Chatbot that facility an interconnection between users and computers in a natural manner. In the twenty-first century, artificial intelligence algorithms have been used to create a revolutionary medium with which users can interact with their needs to prevent and resolve acoustic problems easily; one of these is chatbots, a modern form of interaction. Chatbot systems use human interactions such as decision-making performing daily tasks, replying to users quickly, and solving problems like a human. Chatbots are also called communicational agents or answering engines. Developers and programmers train these chatter bots with the help of artificial intelligence and machine learning systems to interact with users through voice commands, over communication, or text-based messages. We focused on developing the core of AI to interact more efficiently with users and get to understand the user's queries, and provide the user with an appropriate solution. This application works in a very simple way because the data is already put in the system. A few things like matching patterns, NLP (Natural Language Processing), and data mining are used in the application to train the system. Chatbots match the user input texts or voices with the previously included data and give replies according to the data. This knowledge or data has been taken from various sources. Everything in this generation is getting in touch

with the web. It is effective to use a method to manage to advantage anything at your doorway. AI chatbots can be deployed on the web or in mobile applications as software or on computers. It works 24/7 without any delay, only a good internet connection and power have to provide to work smoothly. Chatbots have a significant role in various fields, especially in the medical field. The first established medical chatbot ELIZA was programmed in 1966 to simulate a text-based conversation with a Psychotherapist. The top platform that lends invaluable support to that bot development, Amazon Alexa, has more than 0.1 million applications, many of which are for health. The `World Health Organization (WHO) on the renowned social media Facebook Messenger to counter wrong information and provide accurate information about COVID-19 without any delay, launched a chatbot. As Chatbots are becoming communicational agent in the digital world, it opens many ways for demonstrating a change in behavior for disease prevention and promoting health-related actions on a large scale.

1.2 Problem Definition

We can create awareness through the users and the users can get proper medical solutions to prevent disease. We created a preliminary training model and a study report to improve human interaction in databases in 2021. Through natural language processing, we describe the human behaviors and characteristics of the chatbot.

1.3 Objective

This project is aims to develop a new hybrid food recommender-system to overcome the shortcomings of previous systems, such as ignoring food ingredients, time factor, cold start users, cold start food items and community aspects.

1.4 Limitation of this Project

- The society claim that their hectic schedule gives them no time for periodic medical check-ups.
- That they disregard any uneasiness shown by their body until it is too severe.

1.5 Organization of Documentation

Chapter 2: Literature Survey

Literature survey is that the most vital step in software package development method. Before developing the tool it is necessary to see the time issue, economy n company. Once this stuff square measure, 10 next steps square measure to see that package and language might be used for developing the tool.

Once the programmers begin improving the tool the programmers want heap of external support. This support might be obtained from senior programmers, from different websites. Before building 5 the system on the top of thought square measure taken into consideration for developing the enforced system.

Chapter 3: System Analysis

Chapter three provided to hide the enforced system Analysis.

Chapter 4: System Design

Chapter 4 provided to cover the system modules and design.

Chapter 5: System Implementation and Results

Chapter 5 provided to cover software environment and Results.

Chapter 6: System Testing

This chapter is to cover testing of system.

Chapter 7: Conclusion and Future Scope

This chapter contains conclusion and further work with light scope

2. LITERATURE SURVEY

2.1 Introduction

In the project the author described the whole procedure of developing a chatbot by dividing the process into segments such as speech-to-text conversion, natural language processing, response generation, knowledge base creation, dialogue management, text-tospeech, etc. The author has also included security considerations such as security flaws in chatbot platforms and malicious chatbots. In the paper the authors reviewed topics surrounding the chatbot's knowledge domain, response generation, text processing, machine learning model, and the dataset usage and evaluation strategy topics. In the paper the authors have described the building of a chatbot that can provide an authentic and accurate answer for any type of query using Artificial Intelligence Markup Language (AIML) and Latent Semantic Analysis (LSA) by using the application of the python platform. In the paper the author covers different approaches to chatbot development, including key points on chatbot integration and deployment, and employs machine learning to configure a chatbot. In the paper the author discussed the method of data analysis, which allows the analytical system to learn through the way of solving problems and applying similar methods in its working process. In the paper the authors have analyzed how artificial intelligence and machine learning are implemented in popular use to make advancements in chatbot services specifically in helping users to access college websites. The problem selects the closest matching response from the closest matching statement that matches the input and then chooses a response from an available selection of statements for that response. In the paper, the author has elaborated on the use of diverse neural framework exhibits as the learning technique for setting up the chatbot to make it continuously like human enlistment authority. NLP techniques such as NLTK (python) can be applied for speech analysis and intelligent responses can be generated by designing a model to provide appropriate human-like responses. In the paper the authors have discussed their research on how to design, develop and evaluate a health assistant chatbot application that helps users to ask any personal query related to healthcare without physically availing any hospital facilities. In the paper by the authors have described an AI chatbot whose work depends on Natural language processing.

The chatbot users can upload their queries related to healthcare without physically availing of any healthcare facilities. It uses Google API for voice-text and text-voice conversations. The query is sent to the chatbot and the related answer is displayed on an android app. The system's main concern behind developing this web-based platform is analyzing customers' sentiments. In the paper the authors have described a proposed idea to create a system with AI which meets the user's requirements. The AI can predict diseases based on symptoms and give a list of available treatments. The system can also give the composition of the medicines and their prescribed uses. In our chatbot model, we have incorporated the symptoms of Covid19 since it is based on only this one disease currently. It also shows the list of medications and precautions that the users might take if they are infected. In the paper the authors have focused to show the implementation of a retrieval-based chatbot with voice support. They have investigated other standing chatbots and how it is useful in helping the patients to fetch all the necessary details about Covid 19. In the paper the authors have described the modern chatbot functioning and incorporating institution-specific responses in chatbots related to Covid 19 related queries. The main necessity is for unique response mapping, complex contextualization, and dynamic validations which are led by human resources of content-led industry leaders to develop a chatbot through collaborative communication with companies that are experienced in machine learning and natural language processing. In the paper the authors have discussed the tests they conducted on 701 French participants. They found that interacting with their chatbot for a few minutes significantly increases people's intention to get vaccinated and have a positive attitude towards Covid 19 vaccination. The results suggest that a properly scripted and regularly updated chatbot could offer a powerful resource to help ght hesitancy toward COVID-19 vaccines. In the paper the author has described how to identify chatbot use cases deployed for public health response activities during the Covid 19 pandemic. The authors filtered articles basing them on the abstracts and keywords in their texts and made their assessment. Chatbots, their applications of usage, and chatbot design techniques were extracted from these articles.

In the papers the authors have discussed his study to review the current status of Covid 19 related chatbots in the healthcare sector, identify and categorize the upcoming and new technologies and their applications for Covid 19 and find solutions to related challenges. In the paper the authors have discussed their research where they took an interview study with 29 participants to study the daily positive and negative aspects that are experienced with CAs. By assessing how users presently think about CAs, the authors have identified one of the best criteria that could transform their future design of the model. This can contribute to the end user's perspective by evaluating these functionalities for existing research topics about the guidelines for efficient and seamless user experience for CAs. In the paper the author has outlined the creation of a Penn Medicine chatbot collaboratively created with Verily, Google Cloud, and Quantiphi, a Google Cloud strategic partner. The author has described how interactions with users that can be updated and transformed, such as checkers for disease symptoms, must be consistently made with the capacity, capabilities, and different types of path- ways of the existing health system using it to communicate important information to patient actions required by the user while efficiently managing constrained and contained resources. In the paper the authors have proposed a conversational chatbot on Google Cloud Platform (GCP) to deliver tele health in India to increase the user's access to knowledge related to healthcare and be able to levy the potentials of Artificial Intelligence to bridge the currently existing gap of demand and supply of human healthcare providers.

In the paper the authors have presented the design of a highly efficient Artificial Intelligence Chatbot for evaluation based on diagnostic technologies and recommended efficient and quick measures when patients are exposed to the deadly Covid-19. Along with this, creating a virtual assistant can also help in the measurement of the severity of the infection and connect with registered doctors when the given symptoms become serious. In the paper the author provides chatbot structures ALICE and Elizabeth, illustrating the speak information illustration and sample matching gadget of each. It discusses the problems, which can arise when Dialogue Diversity Corpus is used to retain a functional chatbot system with examples of dialogues spoken by general human beings. A basic implementation of corpus-based chatbot training can be found when a Java program is used to convert from dialogue transcript to AIML.

In the paper the authors have provided statistics concerning epidemiology, serological and molecular diagnosis, the starting place of SARS-CoV-2 and its capacity to contaminate human cells and protection issues. Then it focuses on the available therapies to combat Covid 19, the development of different vaccines, and the role of AI in managing the pandemic and limiting the spread of the virus. In the paper the author discusses the challenges posed by Covid 19 to the education system. In the paper the authors used Chabot technology to implement the medical consultant system service. It was implemented using information from the Doctor Me application's symp- toms and treatment records. The test results demonstrate the proposed system's capability. In the paper the authors have introduced a sketch for a clinical chatbot that gives diagnoses and Treatments based totally on signs provided to the system. The device will be capable to measure the seriousness of the analysis and if needed, it will connect the user to a doctor available online. In the paper the authors have brought to light the usefulness of chatbots in human resource management systems. They have illustrated a detailed analysis of chatbots in HRM, which are also known as HR-bots. It has been studied to emphasize its usefulness in real-time considering the different relevant challenges such as cost factors, complex business domains, limited responsiveness, etc. In the paper the authors have demonstrated their deep learning model, which is named the Long Short-Term Memory (LSTM) network-based patient- dependent model that is adopted for FOG detection. In the paper the authors discuss the sudden impact and severity of Covid 19 around the world and how to fight it by enabling the following possibilities: autonomous everything, pervasive knowledge, assistive technology, and rational decision support. In the paper the authors have discussed the essential roles of some AI-driven techniques (machine learning, deep learning, etc.) and AI-empowered imaging techniques to analyze, predict, and diagnose COVID-19 disease. In the paper the authors have demonstrated the various machine learning models which have been built to predict the PPIs between the virus and human proteins that are further vali- dated using biological experiments. A special chatbot with the ability of visual question answering with the integration of scene-text using PHOCs and fisher vectors is introduced in the paper.

The paper discusses on the impact of algorithmic information processing have on users' attitudes and actions while using artificial intelligence (AI). A paper [33] creates a

cognitive model to describe user inter- actions with conversational journalism (CJ) in the setting of chatbot news using the anthropomorphism and explain ability constructs. In a study, an AI-based machine learning model was developed to forecast the effects of interactions between Paget's disease treatment and pharmaceuticals used to treatosteoporosis. This model reduces the cost and time required to apply the most effective medication combination in medical practic. A deep learning model that could locate FMN interacting residues using a 2D convolutional neural network and positionspecific score matrices.

2.2Existing system

- ✓ Nowadays, people are less aware of their health. In their busy life, they forget to take suitable measures to maintain their health and are less aware of their health status.
- ✓ In the latest news by TOI, we can see that people give no importance to their health and find it time consuming to undergo check-ups at hospitals.
- ✓ The busy-scheduled life has got no place for health.

2.3 Disadvantages

- ✓ The society claim that their hectic schedule gives them no time for periodic medical check-ups.
- ✓ That they disregard any uneasiness shown by their body until it is too severe.

2.4 Proposed system

- ✓ In this proposed system, a medical chatbot is built to be a conversational agent that motivates users to discuss about their health issues and based on the symptoms provided by them; chatbot returns the diagnosis.
- ✓ This chatbot system will be able to identify symptoms from user interaction.
- ✓ Using these extracted symptoms, chatbot predicts the disease and recommends treatment.
- ✓ The machine learning algorithm employed K-nearest neighbor algorithm (KNN). This clearly shows that a medical chatbot can somewhat accurately diagnose patients with simple symptom analysis and a conversational approach done with the help of natural language processing.

2.5 CONCLUSION

I have done survey on my project. In that survey I found some of the disadvantages of the existing project which does not support the existing system. Here I am introduced on my project is very power full compare to the existing system.

3. SYSTEM ANALYSIS

3.1 Introduction

System Analysis is first stage according to System Development Life Cycle model. This System Analysis is a process that starts with the analyst.

Analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system. One aspect of analysis is defining the boundaries of the system and determining whether or not a candidate system should consider other related systems. During analysis, data are collected on the available files, decision points, and transactions handled by the present system.

Logical system models and tools that are used in analysis. Training, experience, and common sense are required for collection of the information needed to do the analysis.

3.2 Software Requirement Specifications

3.2.1 User Requirements

A Software Requirements Specification (SRS) – a requirements specification for a software system is a complete description of the behaviour of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. Non-functional requirements are requirements which impose constraints on the design or implementation

3.2.2 Hardware Requirements:

- ✓ System : Pentium Dual Core.
- ✓ Hard Disk : 120 GB.
- ✓ Ram : 2 GB

3.2.3 Software Requirements:

- ✓ Operating System : Window 7 or above
- ✓ Programming Language : PYTHON
- ✓ Front End : HTML, CSS
- ✓ Database : MySQL 5.0
- ✓ Server : Apache Tomcat
- ✓ Tools : Xampp

3.3 Content Diagram of the project

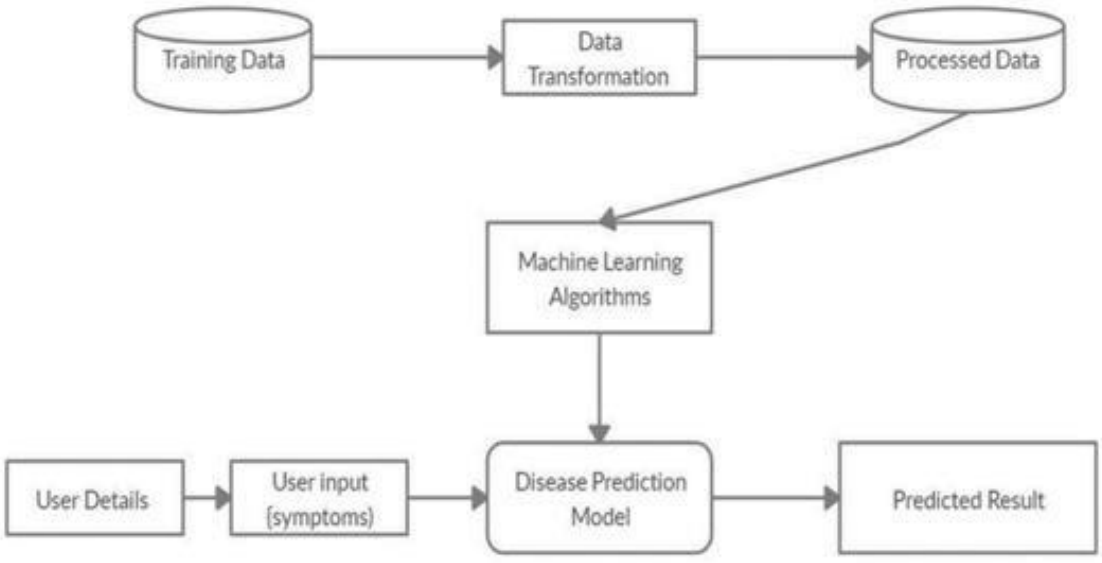


Fig. 2 System architecture

3.4 Algorithms

Navie Bayes

- Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.
- It is mainly used in text classification that includes a high-dimensional training dataset.
- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

Random Forest

- Steps involved in random forest algorithm:
- Step 1: In Random forest n number of random records are taken from the data set having k number of records.
- Step 2: Individual decision trees are constructed for each sample.
- Step 3: Each decision tree will generate an output.
- Step 4: Final output is considered based on Majority Voting or Averaging for Classification and regression respectively.

3.5 CONCLUSION

By observing the above algorithm and diagrams represent that the entire functionality and flow of control of entire project. Here by using the proposed algorithms I am going to design, implement, testing, and also provide the security for my project. By understanding the flow chart diagram each and every one can understand the behavior of the system very efficiently and effectively.

4. SYSTEM DESIGN

4.1 INTRODUCTION

Software vogue sits at the technical kernel of the pc code engineering methodology and is applied despite the event paradigm and house of application. vogue is that the gap among the event section for any designed product or system.

The designer's goal is to supply a model or illustration of associate entity which will later be built. Beginning, once system demand are like and analysed, system vogue is that the first of the three technical activities -design, code and take a glance at that is required to form and verify code.

The importances are declared with one word "Quality". vogue is that the place where quality is fostered in code development. vogue provides North yankee country with representations of code which will assess for quality.

Design is that the alone manner that we are going to accurately translate a customer's browse into a finished product or system. code vogue could be a foundation for all the pc code engineering steps that follow. whereas not a strong vogue we've got an inclination to risk building associate unstable system – one which will be difficult to envision, one whose quality cannot be assessed until the last stage.

During vogue, progressive refinement of knowledge structure, program structure, and procedural details unit developed reviewed and documented. System vogue are viewed from either technical or project management perspective.

From the technical purpose of browse, vogue is comprised of four activities – field vogue, system vogue, interface vogue and procedural vogue.

4.2 ENTITY RELATIONSHIP DIAGRAM

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases.

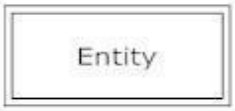
At first glance an entity relationship diagram looks very much like a flowchart. It is the specialized symbols, and the meanings of those symbols, that make it unique.

The History of Entity Relationship Diagrams

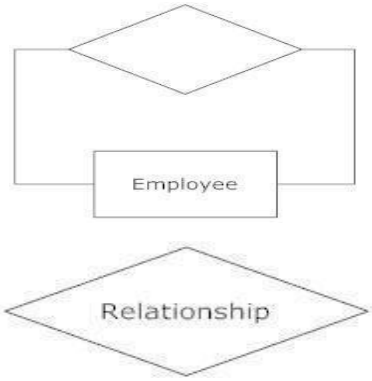
Peter Chen developed ERDs in 1976. Since then Charles Bachman and James Martin have added some slight refinements to the basic ERD principles.

Structure of an Entity Relationship Diagram with Common ERD Notations An entity relationship diagram is a means of visualizing how the information a system produces is related. There are five main components of an ERD: Entities, which are represented by rectangles. An entity is an object or concept about which you want to store information.

A weak entity is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.



Actions, which are represented by diamond shapes, show how two entities share information in the database. In some cases, entities can be self-linked. For example, employees can supervise other employees.



Attributes, which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity. For example, an employee's social security number might be the employee's key attribute.

A multi valued attribute can have more than one value. For example, an employee entity can have multiple skill values.

A derived attribute is based on another attribute. For example, an employee's monthly salary is based on the employee's annual salary.

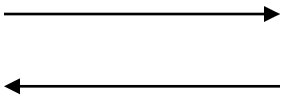
Connecting lines, solid lines that connect attributes to show the relationships of entities in the diagram. Cardinality specifies how many instances of an entity relate to one instance of another entity. Ordinarily is also closely linked to cardinality. While cardinality specifies the occurrences of a relationship, ordinarily describes the relationship as either mandatory or optional. In other words, cardinality specifies the maximum number of relationships and ordinarily specifies the absolute minimum number of relationships.

DATA FLOW DIAGRAM

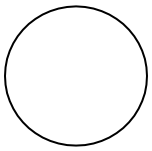
A graphical tool used to describe and analyze the moment of data through a system manual or automated including the process, stores of data, and delays in the system. Data Flow Diagrams are the central tool and the basis from which other components are developed. The transformation of data from input to output, through processes, may be described logically and independently of the physical components associated with the system. The DFD is also know as a data flow graph or a bubble chart.

DFDs are the model of the proposed system. They clearly should show the requirements on which the new system should be built. Later during design activity this is taken as the basis for drawing the system's structure charts. The Basic Notation used to create a DFD's are as follows:

1. Dataflow: Data move in a specific direction from an origin to a destination.



2. Process: People, procedures, or devices that use or produce (Transform) Data. The physical component is not identified.



3. Source: External sources or destination of data, which may be People, programs, organizations or other entities.

4. Data Store: Here data are stored or referenced by a process in the System.

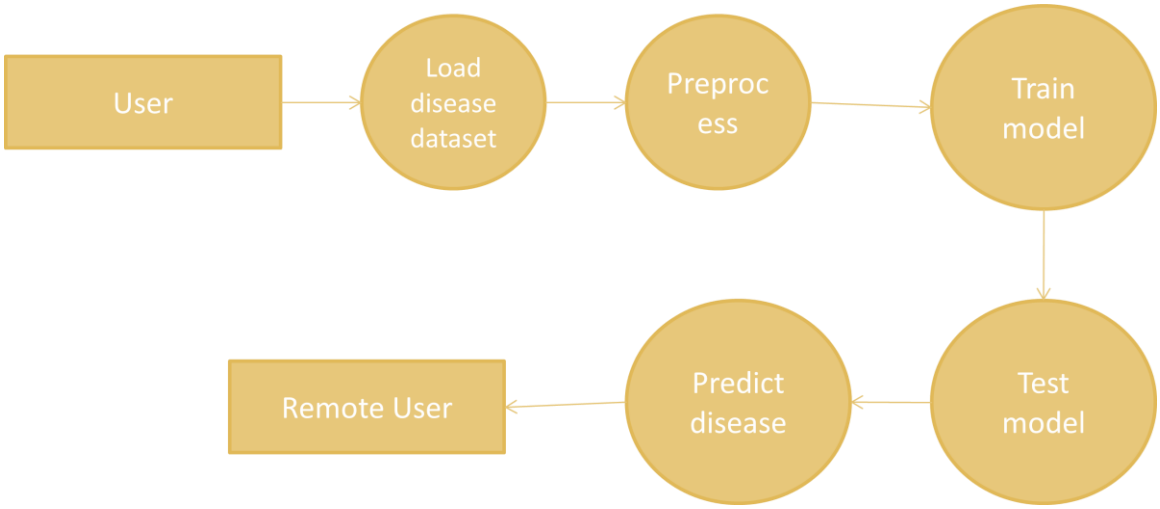


Fig. 4.1: Data Flow Diagram

4.3 UML BASICS

The Unified Modelling Language™ (UML®) can be a customary visual modelling language purported to be used for modelling business and similar processes, analysis, design, and implementation of software-based systems

UML can be a typical language for business analysts, software system package architects and developers accustomed describe, specify, design, and document existing or new business processes, structure and behavior of artifacts of software system package systems.

UML could also be applied to varied application domains (e.g., banking, finance, internet, aerospace, healthcare, etc.) it's going to be used with all major object and half software system package development methods and for various implementation platforms (e.g., J2EE, .NET).

UML can be a customary modelling language, not a software system package development methodology. UML 1.4.2 Specification explained that process:

- ✓ Provides steerage on the order of a team's activities,
- ✓ Specifies what artifacts ought to be developed,
- ✓ Directs the tasks of individual developers and therefore the team as an entire, and offers criteria for observation and activity a project's merchandise and activities.

UML is designedly methodology freelance and can be applied inside the context of assorted processes. Still, it's best suited to be used case driven, unvarying and progressive development processes. Associate in Nursing example of such methodology is Rational Unified methodology (RUP).

UML is not complete and it isn't totally visual. Given some UML diagram, we are going to not ensure to understand drawn [*fr1] or behavior of the system from the diagram alone. Some information are often designedly omitted from the diagram, some information pictured on the diagram might need altogether completely different interpretations, and a number of concepts of UML have no graphical notation within the least, so there is no due to depict those on diagrams.

For example, linguistics of multiplicity of actors and multiplicity of use cases on use case diagrams is not made public precisely among the UML specification and can mean either coincident or successive usage of use cases.

Name of academic degree abstract classifier is shown in italics whereas final classifier has no specific graphical notation, so there is no because of ensure whether or not or not classifier is final or not from the diagram.

There square measure a pair of broad caetgories of diagrams thus square measure all over again divided into sub-categories:

- ✓ Structural Diagrams
- ✓ Behavioral DiagramS

Structural Diagrams:

The structural diagrams represent the static facet of the system. These static aspects represent those components of a diagram that forms the most structure and so stable.

These static components are represents by categories, interfaces, objects, parts and nodes. The four structural diagrams are:

- ✓ Class diagram
- ✓ Object diagram
- ✓ Component diagram
- ✓ Deployment diagram

Behavioural Diagrams:

Any system will have 2 aspects, static and dynamic. Therefore a model is taken into account as complete once each the aspects area unit lined totally.

Behavioural diagrams essentially capture the dynamic side of a system. Dynamic side are often any delineated because the changing/moving components of a system.

UML has the subsequent 5 forms of activity diagrams:

- ✓ Use case diagram
- ✓ Sequence diagram
- ✓ Collaboration diagram
- ✓ State chart diagram
- ✓ Activity diagram

4.3.1 Use case Diagram

Use case diagrams are a group of use cases, actors and their relationships. They represent the employment case read of a system.

So use case diagram is employed to explain the relationships among the functionalities and their internal/external controllers. These controllers are called actors.

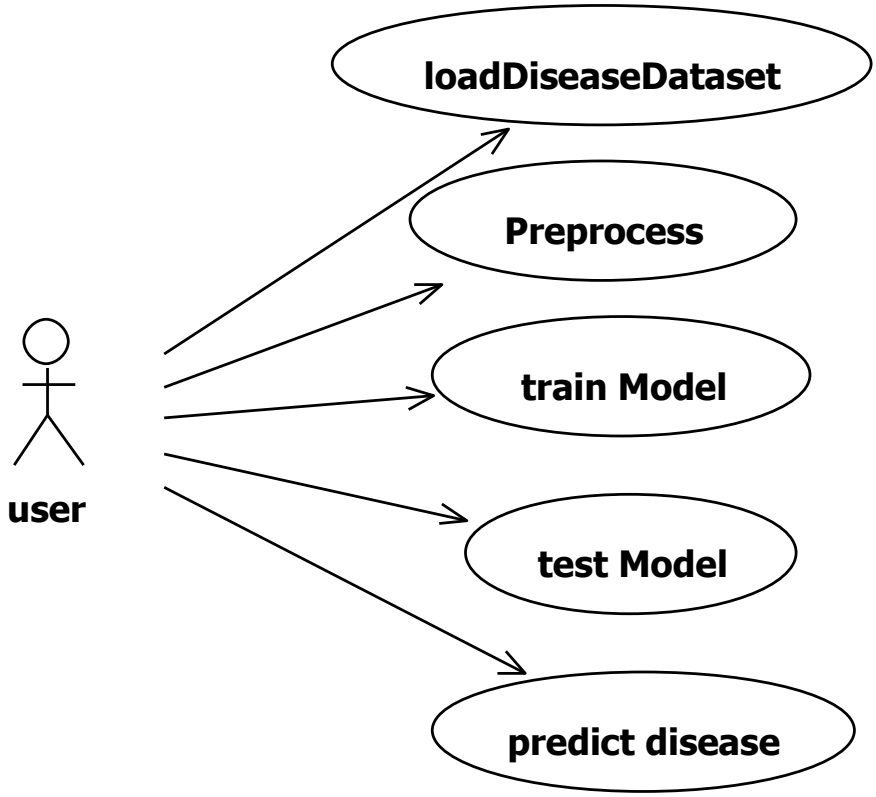


Fig. 4.4: Use case Diagram

4.3.2 Class Diagram

Class diagrams area unit the foremost common diagrams employed in UML. category diagram consists of categories, interfaces, associations and collaboration. Category diagrams primarily represent the thing directed read of a system that is static in nature. Active category is employed in a very category diagram to represent the concurrency of the system.

Class diagram represents the thing orientation of a system. Therefore it's usually used for development purpose. This can be the foremost wide used diagram at the time of system construction.

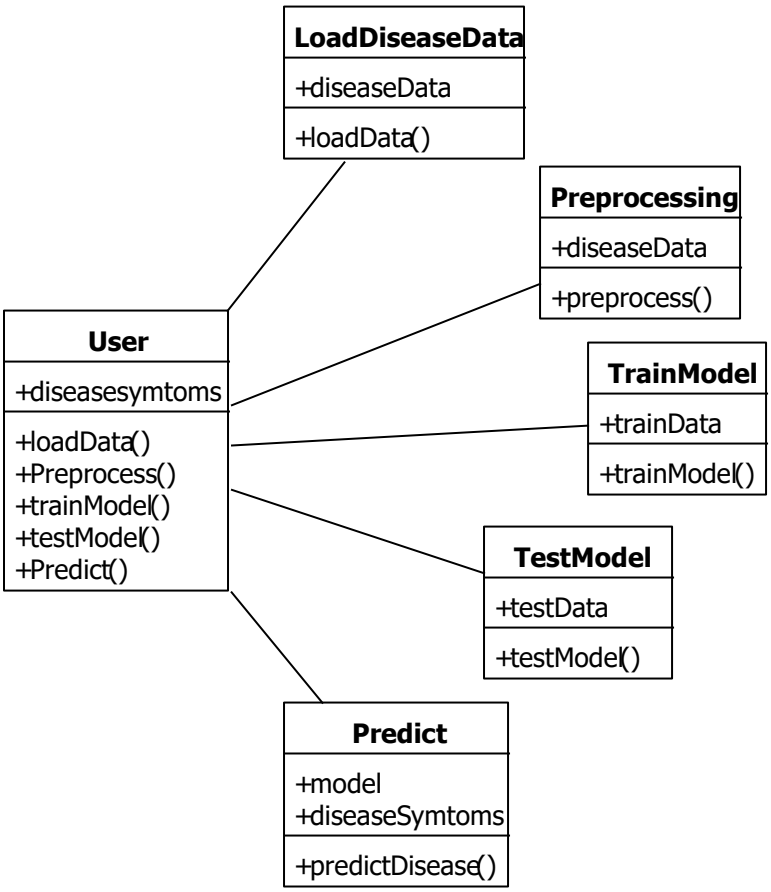


Fig. 4.6: Class Diagram

4.3.3 Sequence Diagram

A sequence diagram is Associate in nursing interaction diagram. From the name it's clear that the diagram deals with some sequences, that area unit the sequence of messages flowing from one object to a different. Interaction among the elements of a system is incredibly necessary from implementation and execution perspective. Thus Sequence diagram is employed to see the sequence of calls in an exceedingly system to perform a particular practicality.

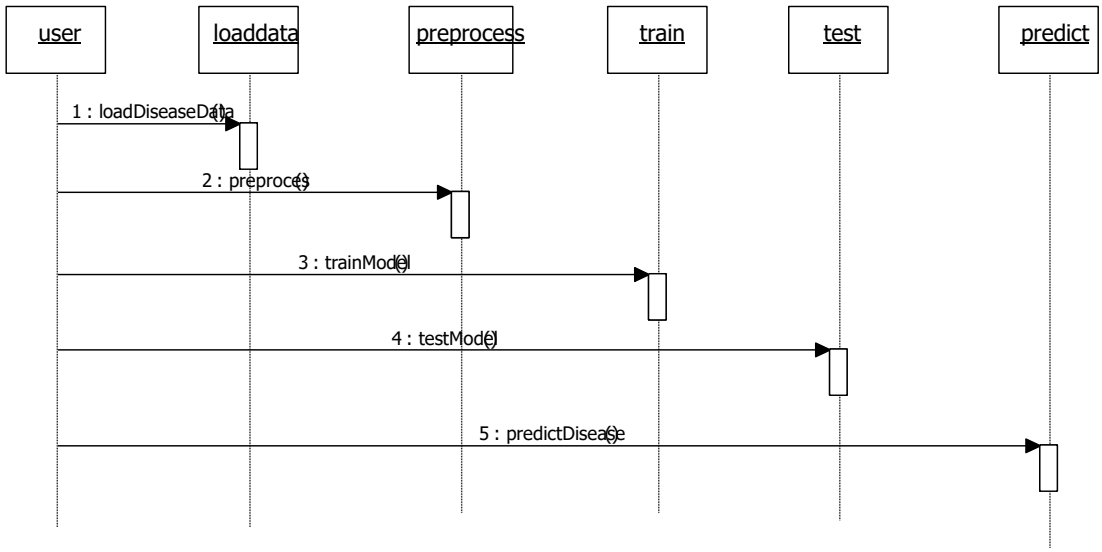


Fig. 4.7: Sequence Diagram

4.3.4 Activity Diagram

Activity diagram is another vital diagram in UML to explain dynamic aspects of the system. Activity diagram is largely a flow chart to represent the flow kind one activity to a different activity. The activity may be delineating as associate operation of the system.

So the management flow is drawn from one operation to a different. This flow may be serial, branched or coinciding. Activity diagrams deals with all style of flow management by victimization completely different parts like fork, join etc.

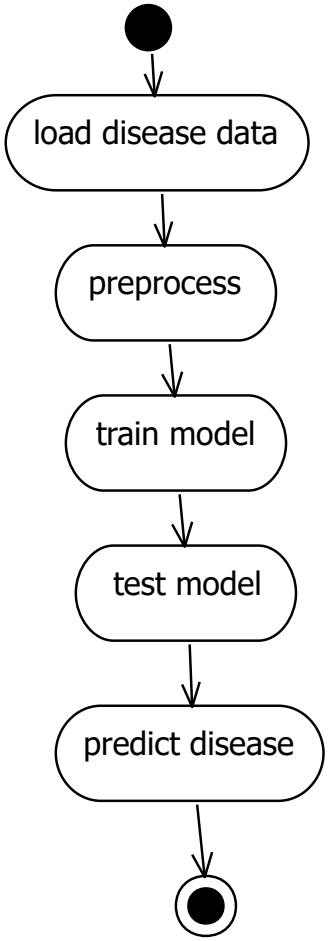


Fig.4.8: Activity Diagram

4.4 MODULES:

- Load Dataset
- Preprocess
- Split and Train and Test Model
- Prediction

4.4.1 MODULES DESCSRIPTION:

Load Dataset

- ✓ In this phase, load the dataset into program and extract the data from the .csv file.
- ✓ This data can be analyzed and extract the best features to preprocess the data.

Preprocess

- ✓ For the given data set, there are quite a few 'NA' values which are filtered in python. Furthermore, as the data set consists of numeric data, we used robust scaling, which is quite similar to normalization, but it instead uses the interquartile range whereas normalization is something which normalization shrinks the data in terms of 0 to 1.

Split and Train and Test Model

- ✓ In this module, the service provider split the Used dataset into train and test data of ratio 70 % and 30 % respectively. The 70% of the data is consider as train data which is used to train the model and 30% of the data is consider as test which is used to test the model.

Prediction

- ✓ In this module, the user enter the disease to predict the disease type. Random Forest and Naive Bayes from Sk learn for the disease prediction. The model has been pretrained on a dataset of 4920 trials with 132 symptoms and 41 diseases.

4.5 CONCLUSION

By using DFD, UML diagrams I this project I was represented the flow of data between system and user. I was represented the behaviour of the entire system through UML diagrams.

5 IMPLEMENTATION & RESULTS

5.1 Introduction

In this section, we describe the all the major functions of the system and we provide brief description about key functions. Finally we show the input forms and output forms of this project.

5.2 Explanation of Key Functions

- ✓ Load Dataset
- ✓ Pre-processing
- ✓ Split Dataset
- ✓ Train Model
- ✓ Test Model
- ✓ Prediction

5.2.1 Forms

Diabeties_prediction.py #

-*- coding: utf-8 -*-

"""diabetese_prediction.ipynb

Automatically generated by Colaboratory. Original file is located at

https://colab.research.google.com/drive/1Z5S75PULGFew2JX0qqXpzPk1Vn4L7vG8

Diabetes Prediction (Supervised Machine Learning)

"" import numpy as np import pandas as pd from

sklearn.preprocessing import StandardScaler

from sklearn.model_selection import

train_test_split from sklearn import svm from

sklearn.metrics import accuracy_score data =

```
pd.read_csv('.\data\diabetes.csv') data.head() #
data.shape data.describe()
#                                    data['Outcome'].value_counts()
data.groupby('Outcome').mean()
# separating the data and labels
X = data.drop(columns='Outcome', axis=1)
Y = data['Outcome']
# print(X)
# print(Y)
"""Data          Standardization"""
scalar = StandardScaler()
X = scalar.fit_transform(X)
#  X  x_train,  x_test,  y_train,  y_test  =  train_test_split(X,Y,  test_size=0.2,
random_state=1)
"""Training the model""" classifier =
svm.SVC(kernel='linear')
classifier.fit(x_train,              y_train)
"""Accuracy Score""" x_train_pred =
classifier.predict(x_train)
training_accuracy                    =
accuracy_score(x_train_pred,   y_train)
print('Accuracy Score of training data :
',  training_accuracy)  x_test_pred  =
classifier.predict(x_test) test_accuracy =
```

```
accuracy_score(x_test_pred,y_test)
print('Accuracy Score of test data : ',
test_accuracy) """#### Making a
predictive system""" input_data = (5,
166, 72, 19, 175, 25.8, 0.587, 51) #
changing input data to numpy array
input_data = np.asarray(input_data)
# reshaping the array ( as we are predicting for one instance)
input_data = input_data.reshape(1, -1) # standardize the
input data std_data = scalar.transform(input_data) pred =
classifier.predict(std_data) if pred :
   print('The person is diabetic')
else :
   print('The person is not diabetic') """###
Saving the training model""" import pickle
filename = 'diabetes_model.sav'
pickle.dump(classifier,
open(filename,'wb'))
```

Heart_disease_prediction.py

```
# -*- coding: utf-8 -*-
"""heart_disease_prediction.ipynb
Automatically generated by Colaboratory. Original
file is located at
```

tps://colab.research.google.com/drive/13WMtld0S6B54CUmE8VaER3HN1Z9eXDO1

Heart Disease Prediction Model

""" import numpy as np import pandas as pd from

sklearn.model_selection import train_test_split from

sklearn.linear_model import LogisticRegression from

sklearn.metrics import accuracy_score data =

pd.read_csv('.\data\heart_disease_data.csv')

data.head()

data.shape

data.info()

data.isnull().sum()

data.describe()

data['target'].value_counts()

X = data.drop(columns='target', axis=1) Y = data['target'] x_train, x_test, y_train,

y_test = train_test_split(X,Y, test_size=0.2, random_state=1)

"""##### Logistic Regression"""
model = LogisticRegression() model.fit(x_train, y_train) #

Model Evaluation training_pred = model.predict(x_train)

training_accuracy = accuracy_score(training_pred, y_train)

print('Accuracy Score of training data : ',

training_accuracy) test_pred = model.predict(x_test)

test_accuracy = accuracy_score(test_pred,y_test)

print('Accuracy Score of test data : ', test_accuracy)

input_data = (52, 0, 2, 136, 196, 0, 0, 169, 0, 0.1,
1, 0, 2)

```
# changing input data to numpy array input_data

= np.asarray(input_data)

# reshaping the array ( as we are predicting for one instance)

input_data = pd.DataFrame(input_data.reshape(1, -1), columns=['age','sex','cp','trestbps','chol','fbs','restecg','thalach','exang','oldpeak','slope','ca','thal'])

pred = model.predict(input_data) if

pred :

  print('The person has Heart Disease')

else :

  print('The person does not have a Heart Disease')

import pickle pickle.dump(model,

open('heart_disease_model.sav','wb'))
```

Parkinsons Disease Detection.py

```
# -*- coding: utf-8 -*-

"""parkinson's_disease_detection.ipynb

Automatically generated by Colaboratory. Original file is located at

https://colab.research.google.com/drive/16g4Pvdn84xzFvcqOWZl0L_IDEqoNjpZc

### Parkinson's Disease Detection Model

""" import numpy as np import pandas as pd from

sklearn.preprocessing import StandardScaler from

sklearn.model_selection import train_test_split

from sklearn import svm from sklearn.metrics
```

```
import accuracy_score data =
pd.read_csv('.\data\parkinsons.csv') data.head()
# data.shape
# data.info()
# data.isnull().sum() #
data['status'].value_counts()
data.groupby('status').mean()
X = data.drop(columns=['name','status'], axis=1) Y = data['status'] x_train, x_test,
y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=1) scalar =
StandardScaler() scalar.fit(x_train) x_train = scalar.transform(x_train) x_test =
scalar.transform(x_test) x_train
"""#### SVM model""" model = svm.SVC(kernel='linear')
model.fit(x_train, y_train) """Accuracy Score"""
training_pred = model.predict(x_train) training_accuracy =
accuracy_score(y_train,training_pred) print('Accuracy
Score of training data : ', training_accuracy) test_pred =
model.predict(x_test) test_accuracy =
accuracy_score(test_pred,y_test) print('Accuracy Score of
test data : ', test_accuracy)
"""#### Building a predictive system"""
input_data = (162.568, 198.346, 77.63, 0.00502, 0.00003, 0.0028,
        0.00253, 0.00841, 0.01791, 0.168, 0.00793, 0.01057,
        0.01799, 0.0238, 0.0117, 25.678, 0.427785,
        0.723797, -6.635729, 0.209866, 1.957961, 0.135242
)
```

```
input_data = np.asarray(input_data) input_data = input_data.reshape(1, -1) input_data =
pd.DataFrame(input_data, columns=['MDVP:Fo(Hz)', 'MDVP:Fhi(Hz)',
'MDVP:Flo(Hz)', 'MDVP:Jitter(%)','MDVP:Jitter(Abs)', 'MDVP:RAP', 'MDVP:PPQ',
'Jitter:DDP','MDVP:Shimmer', 'MDVP:Shimmer(dB)', 'Shimmer:APQ3',
'Shimmer:APQ5','MDVP:APQ', 'Shimmer:DDA', 'NHR', 'HNR', 'RPDE', 'DFA','spread1',
'spread2', 'D2', 'PPE']) input_data =
scalar.transform(input_data) pred =
model.predict(input_data) if pred :
   print('The person has Parkinsons')
else :
   print('The person does not Parkinsons Disease')
import pickle pickle.dump(model,
open('parkinsons_model.sav','wb'))
```

app.py

```
import streamlit as st import pickle from streamlit_option_menu import
option_menu import numpy as np # Loading the saved model
diabetes_model = pickle.load(open('diabetes_model.sav', 'rb'))
heart_disease_model = pickle.load(open('heart_disease_model.sav',
'rb')) parkinsons_model = pickle.load(open('parkinsons_model.sav',
'rb'))
# sidebar for navigation with
st.sidebar:
   selected = option_menu('Multiple Disease Predictoin System', ['Diabetes Prediction',
'Heart Disease Prediction',
                                                'Parkinsons Prediction'],
                icons=['activity', 'heart', 'person-fill']
                , default_index=0)
```

```
# Diabetes Prediction Page if selected
== 'Diabetes Prediction':
st.title('Diabetes Prediction using ML')
col1, col2, col3 = st.columns(3) with
col1:
    Pregnancies = st.text_input('Number of Pregnancies')
with col2:
    Glucose = st.text_input('Glucose Level')
with col3:
    BloodPressure = st.text_input('Blood Pressure Value')
with col1:
    SkinThickness = st.text_input('Skin Thickness Value')
with col2:
    Insulin = st.text_input('Insulin Level')
with col3:
    BMI = st.text_input('BMI value')
with col1:
    DiabetesPedigreeFunction = st.text_input('Diabetes Pedigree Function value')
with col2:
    Age = st.text_input('Age of the Person')
diab_dignosis = '' if st.button('Diabetes
Test Result'): diab_pred =
diabetes_model.predict([[Pregnancies,
```

```
Glucose,        BloodPressure,
SkinThickness, Insulin,
                                BMI,  DiabetesPedigreeFunction,  Age]])
if diab_pred[0] == 1:
        diab_dignosis = 'The person is Diabetic'
    else:
        diab_dignosis = 'The person is Not Diabetic'
st.success(diab_dignosis) # Heart Disease
Prediction Page if selected == 'Heart Disease
Prediction':
  st.title('Heart  Disease  Prediction  using
ML')        col1,  col2,  col3  =  st.columns(3)
with col1:
    age   =   st.text_input('Age')
with col2:
    sex   =   st.text_input('Sex')
with col3:
    cp  =  st.text_input('Chest  Pain  types')
with col1:
    trestbps  =  st.text_input('Resting  Blood  Pressure')
with col2:
    chol  =  st.text_input('Serum  Cholestoral  in  mg/dl')
with col3:
    fbs = st.text_input('Fasting Blood Sugar > 120 mg/dl')
  with col1:
```

```
    restecg = st.text_input('Resting Electrocardiographic results')
with col2:
    thalach = st.text_input('Maximum Heart Rate achieved')
with col3:
    exang = st.text_input('Exercise Induced Angina')
with col1:
    oldpeak = st.text_input('ST depression induced by exercise')
with col2:
    slope = st.text_input('Slope of the peak exercise ST segment')
with col3:
    ca = st.text_input('Major vessels colored by flourosopy')
with col1:
    thal = st.text_input('thal: 0 = normal; 1 = fixed defect; 2 = reversable
defect')  heart_diagnosis = ''  if st.button('Heart Disease Test Result'):
    arr = np.array([[int(age), int(sex), int(cp), int(trestbps), int(chol), int(fbs),
int(restecg), int(thalach),                int(exang), float(oldpeak), int(slope), int(ca),
int(thal)]])      heart_prediction = heart_disease_model.predict(arr)      if
heart_prediction[0] == 1:
        heart_diagnosis = 'The person is having heart disease'
    else:
        heart_diagnosis = 'The person does not have any heart disease'
st.success(heart_diagnosis) # Parkinson's Prediction Page if selected == "Parkinsons
Prediction":  st.title("Parkinson's Disease Prediction using ML")  col1, col2, col3,
col4, col5 = st.columns(5)  with col1:
```

```
    fo = st.text_input('MDVP:Fo(Hz)')
with col2:
    fhi = st.text_input('MDVP:Fhi(Hz)')
with col3:
    flo = st.text_input('MDVP:Flo(Hz)')
with col4:
    Jitter_percent = st.text_input('MDVP:Jitter(%)')
with col5:
    Jitter_Abs = st.text_input('MDVP:Jitter(Abs)')
with col1:
    RAP = st.text_input('MDVP:RAP')
with col2:
    PPQ = st.text_input('MDVP:PPQ')
with col3:
    DDP = st.text_input('Jitter:DDP')
with col4:
    Shimmer = st.text_input('MDVP:Shimmer')
with col5:
    Shimmer_dB = st.text_input('MDVP:Shimmer(dB)')
with col1:
    APQ3 = st.text_input('Shimmer:APQ3')
with col2:
    APQ5 = st.text_input('Shimmer:APQ5')
with col3:
```

```
    APQ = st.text_input('MDVP:APQ')
with col4:
    DDA = st.text_input('Shimmer:DDA')
with col5:
    NHR = st.text_input('NHR')
with col1:
    HNR = st.text_input('HNR')
with col2:
    RPDE = st.text_input('RPDE')
with col3:
    DFA = st.text_input('DFA')
with col4:
    spread1 = st.text_input('spread1')
with col5:
    spread2 = st.text_input('spread2')
with col1:
    D2 = st.text_input('D2')
with col2:
    PPE = st.text_input('PPE')
parkinsons_diagnosis = "" if st.button("Parkinson's Test Result"):
    parkinsons_pred = parkinsons_model.predict([[fo, fhi, flo, Jitter_percent, Jitter_Abs, RAP, PPQ, DDP,
                                                 Shimmer, Shimmer_dB, APQ3, APQ5, APQ, DDA, NHR, HNR, RPDE,
```

```
                                    DFA, spread1, spread2, D2, PPE]])
if parkinsons_pred[0] == 1:
    parkinsons_diagnosis = "The person has Parkinson's disease"
else:
    parkinsons_diagnosis = "The person does not have Parkinson's disease"
st.success(parkinsons_diagnosis)
```

5.2.2 Output Screens

- First we open the Home in the Desk top

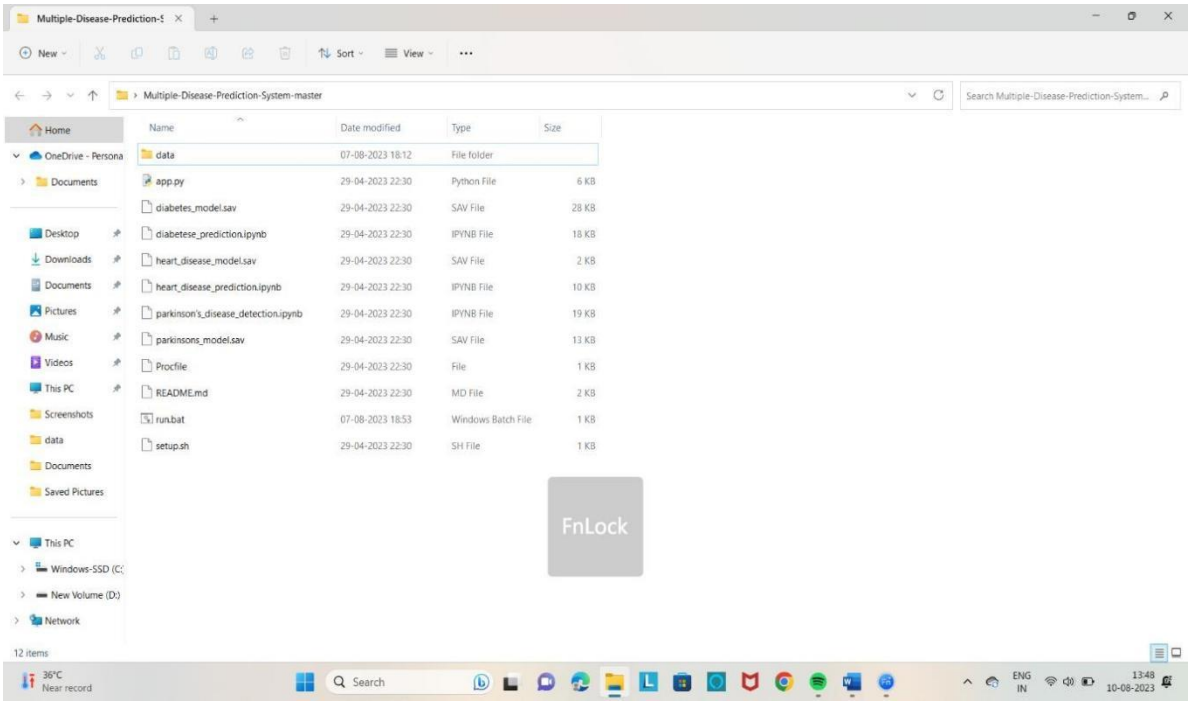


FIG-1

- After open the home page interface like this

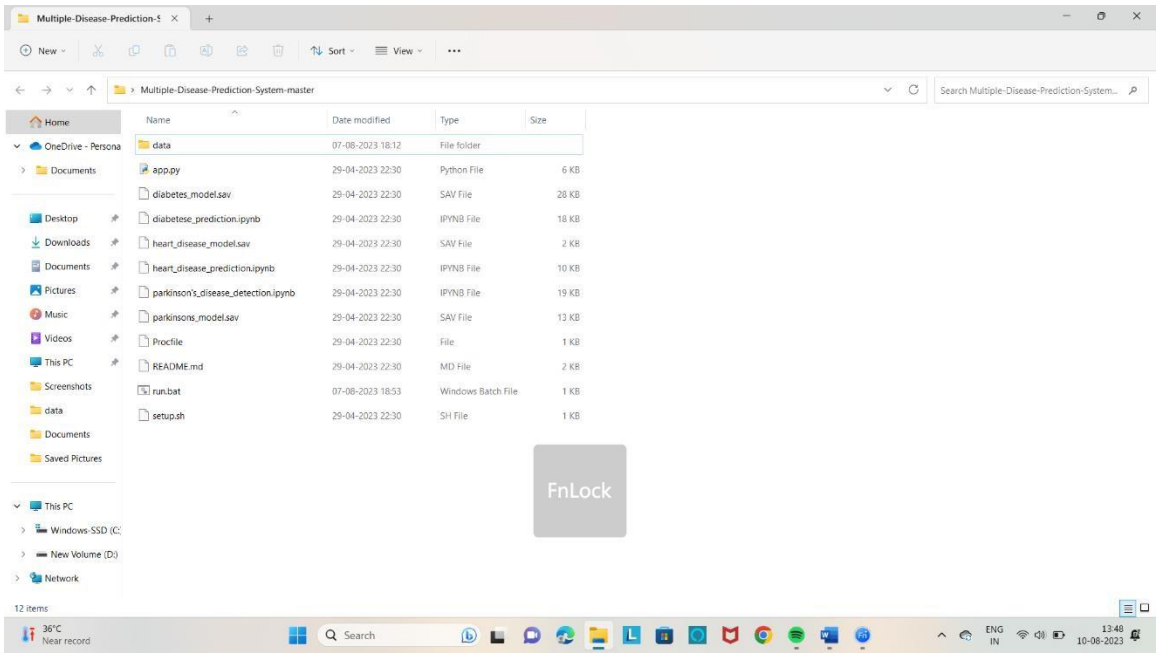


FIG:2

➢ Now we need to choose the data set file

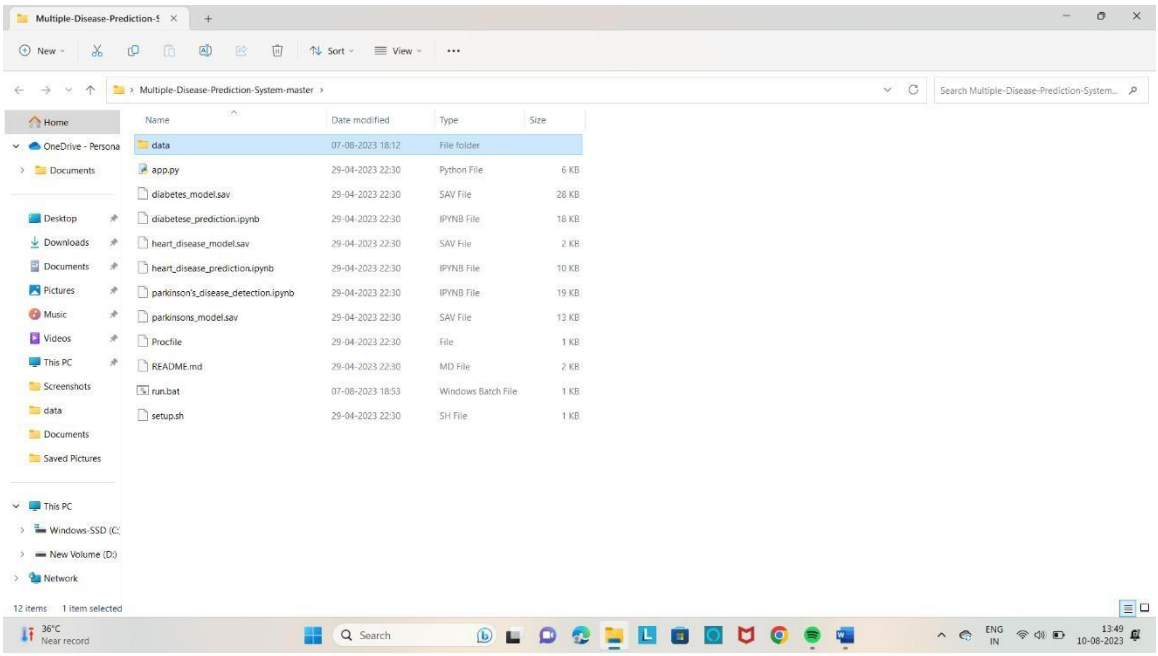

Multiple-Disease-Prediction-S
New
Sort
View
Multiple-Disease-Prediction-System-master
Search Multiple-Disease-Prediction-System...
Home
OneDrive - Persona
Documents
Desktop
Downloads
Documents
Pictures
Music
Videos
This PC
Screenshots
data
Documents
Saved Pictures
This PC
Windows-SSD (C:
New Volume (D:)
Network
Name
Date modified
Type
Size
data 07-08-2023 18:12 File folder
app.py 29-04-2023 22:30 Python File 6 KB
diabetes_model.sav 29-04-2023 22:30 SAV File 28 KB
diabetese_prediction.ipynb 29-04-2023 22:30 IPYNB File 18 KB
heart_disease_model.sav 29-04-2023 22:30 SAV File 2 KB
heart_disease_prediction.ipynb 29-04-2023 22:30 IPYNB File 10 KB
parkinson's_disease_detection.ipynb 29-04-2023 22:30 IPYNB File 19 KB
parkinsons_model.sav 29-04-2023 22:30 SAV File 13 KB
Procfile 29-04-2023 22:30 File 1 KB
README.md 29-04-2023 22:30 MD File 2 KB
run.bat 07-08-2023 18:53 Windows Batch File 1 KB
setup.sh 29-04-2023 22:30 SH File 1 KB
12 items 1 item selected
36°C
Near record
Search
ENG
IN
13:49
10-08-2023

FIG:3

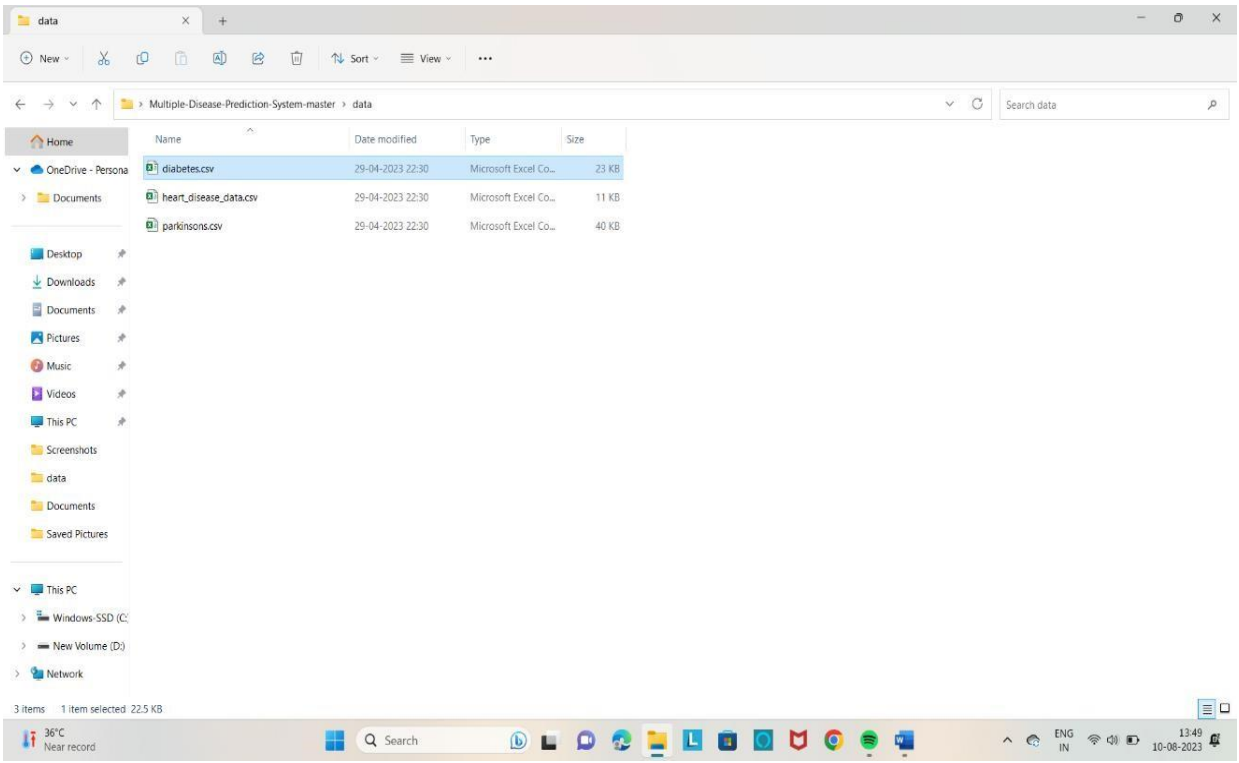
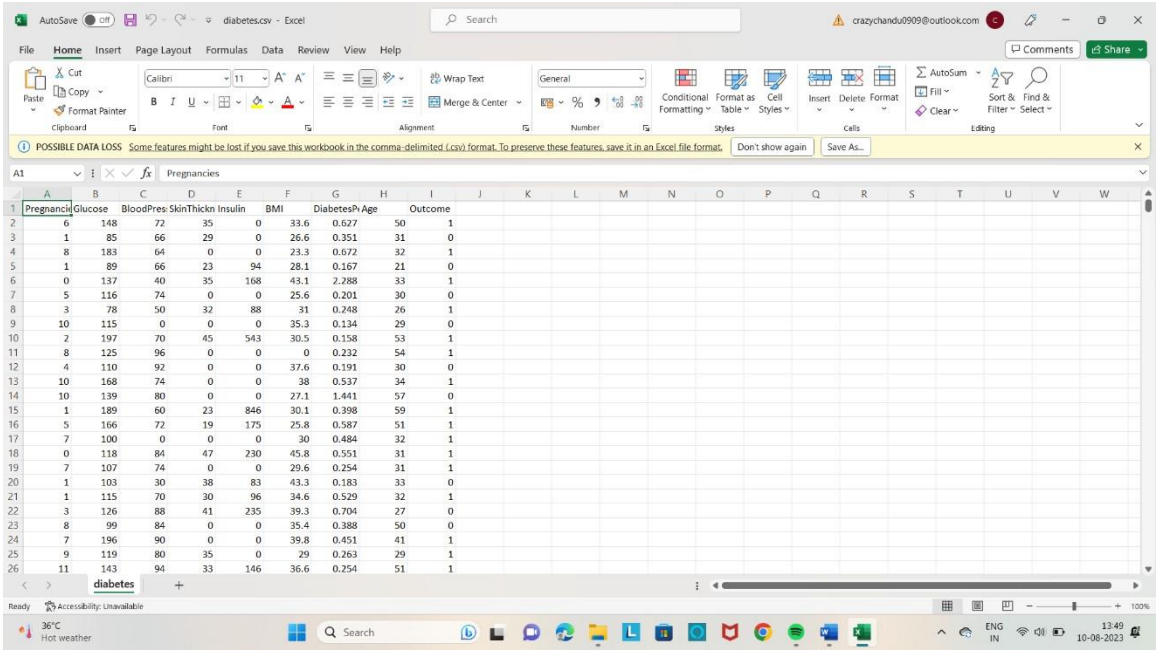


FIG:4

- We are using the three different diseases data set file they are
- Diabetes data set
- Heart disease data set
- Parkinsons data se
- From the above figure we open the diabetes data set information



Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	25.6	0.201	30	0
3	78	50	32	88	31	0.248	26	1
10	115	0	0	0	35.3	0.134	29	0
2	197	70	45	543	30.5	0.158	53	1
8	125	96	0	0	0	0.232	54	1
4	110	92	0	0	37.6	0.191	30	0
10	168	74	0	0	38	0.537	34	1
10	139	80	0	0	27.1	1.441	57	0
1	189	60	23	846	30.1	0.398	59	1
5	166	72	19	175	25.8	0.587	51	1
7	100	0	0	0	30	0.484	32	1
0	118	84	47	230	45.8	0.551	31	1
7	107	74	0	0	29.6	0.254	31	1
1	103	30	38	83	43.3	0.183	33	0
1	115	70	30	96	34.6	0.529	32	1
3	126	88	41	235	39.3	0.704	27	0
8	99	84	0	0	35.4	0.388	50	0
7	196	90	0	0	39.8	0.451	41	1
9	119	80	35	0	29	0.263	29	1
11	143	94	33	146	36.6	0.254	51	1

FIG:5

- in the above figure diabetes data set

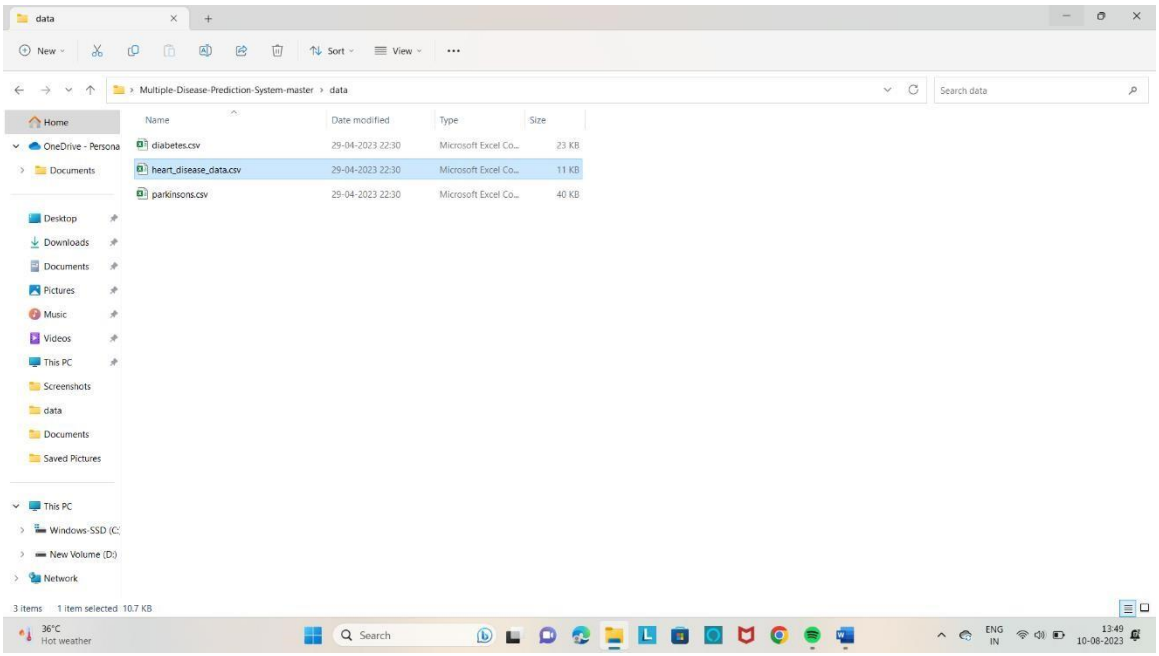
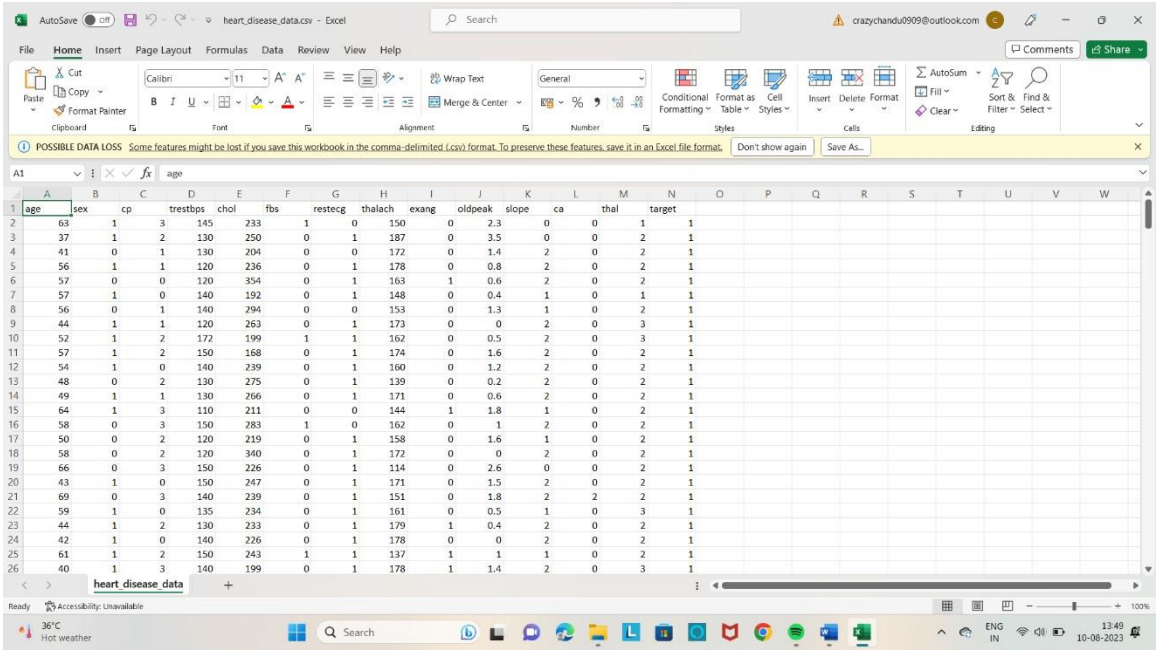


FIG:6

- In the above figure home page of the heart disease data set



age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
44	1	1	120	263	0	1	173	0	0	2	0	3	1
52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
57	1	2	150	168	0	1	174	0	1.6	2	0	2	1
54	1	0	140	239	0	1	160	0	1.2	2	0	2	1
48	0	2	130	275	0	1	139	0	0.2	2	0	2	1
49	1	1	130	266	0	1	171	0	0.6	2	0	2	1
64	1	3	110	211	0	0	144	1	1.8	1	0	2	1
58	0	3	150	283	1	0	162	0	1	2	0	2	1
50	0	2	120	219	0	1	158	0	1.6	1	0	2	1
58	0	2	120	340	0	1	172	0	0	2	0	2	1
66	0	3	150	226	0	1	114	0	2.6	0	0	2	1
43	1	0	150	247	0	1	171	0	1.5	2	0	2	1
69	0	3	140	239	0	1	151	0	1.8	2	2	2	1
59	1	0	135	234	0	1	161	0	0.5	1	0	3	1
44	1	2	130	233	0	1	179	1	0.4	2	0	2	1
42	1	0	140	226	0	1	178	0	0	2	0	2	1
61	1	2	150	243	1	1	137	1	1	1	0	2	1
40	1	3	140	199	0	1	178	1	1.4	2	0	3	1

FIG:7

In the above figure displays heart disease data set

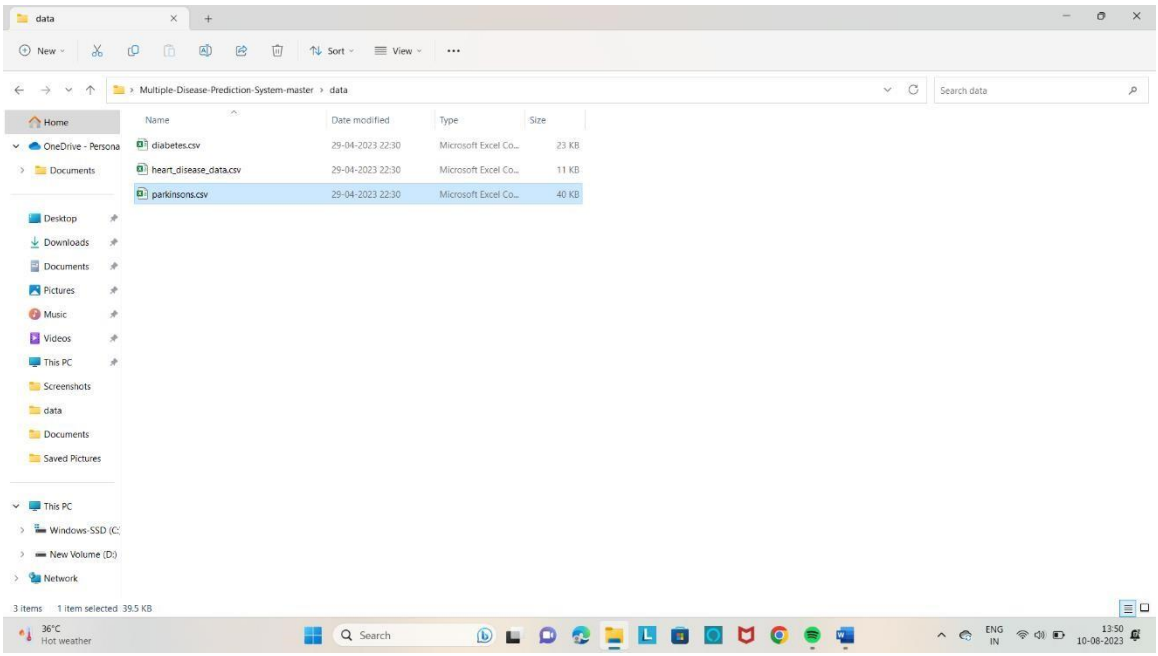


FIG:8

- In the above figure home page of Parkinsons data set

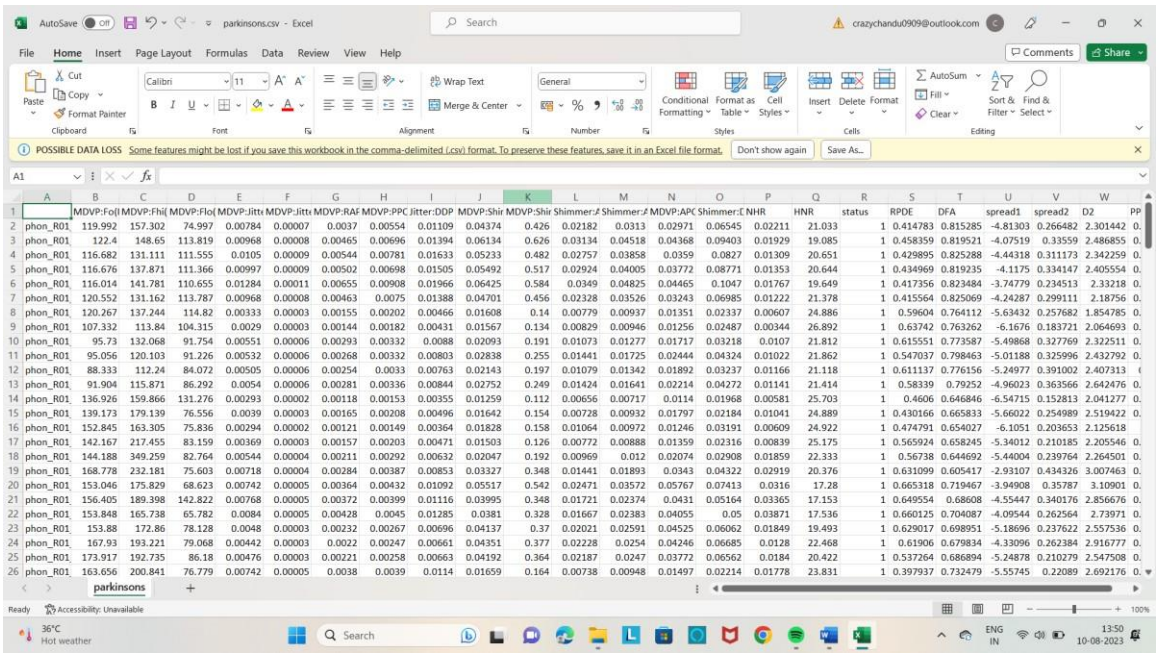


FIG:9

- In the above figure displays Parkinsons data set information

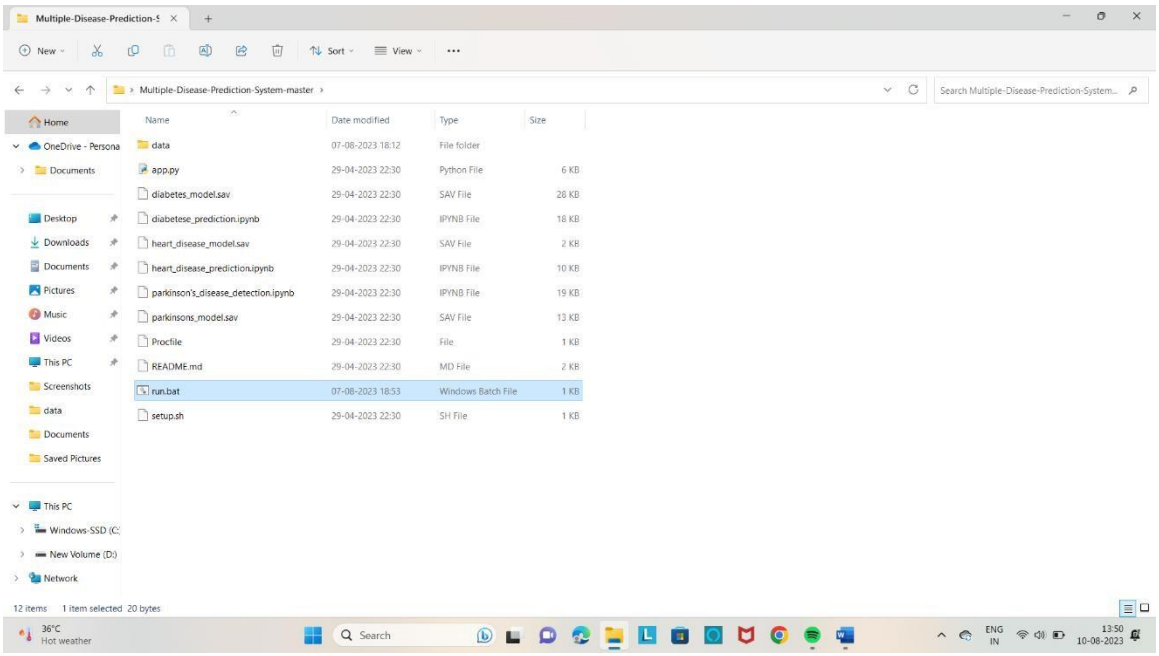


FIG:10

➢ Now click on the run battle directly enter in to the execution part

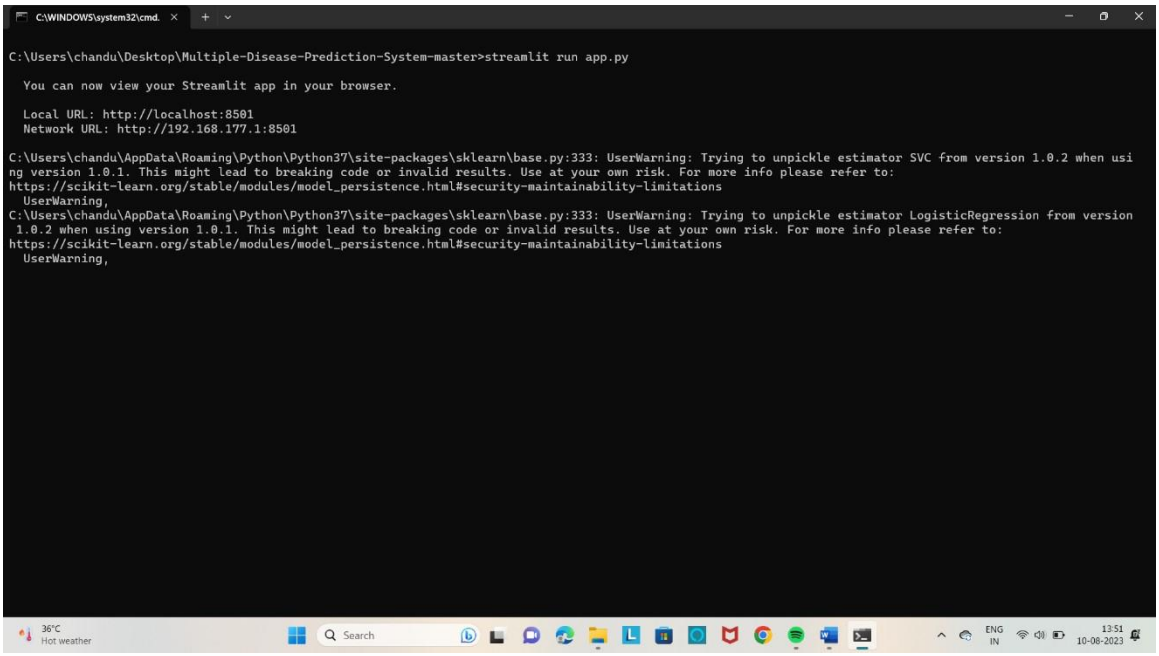


FIG:11

- Next predict the diabetes disease prediction using ML

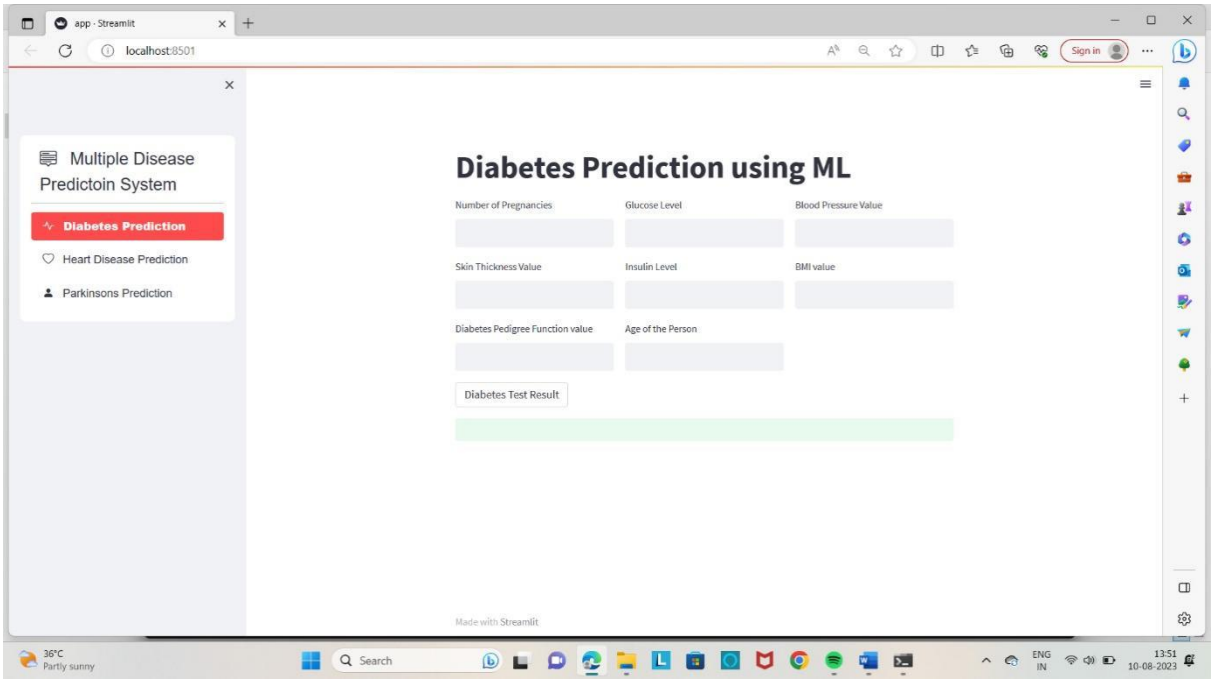


FIG:12

- In the above figure diabetes stream lit now entire the values in the data to predict the disease are not

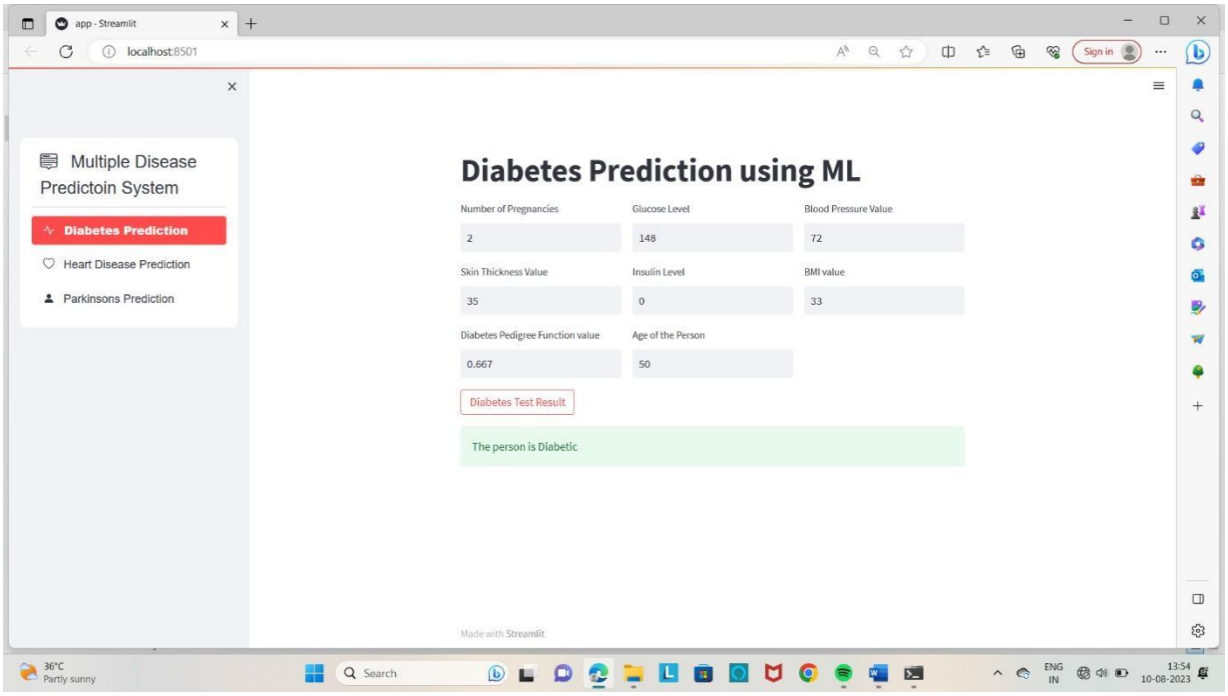


FIG:13

- From the above figure before predict the diabetes disease the patient must be register with his details like number of pregnancies, glucose level, BP……..
- Entire the patient details it help us to predict diabetes disease are not ➢ This person is diabetic

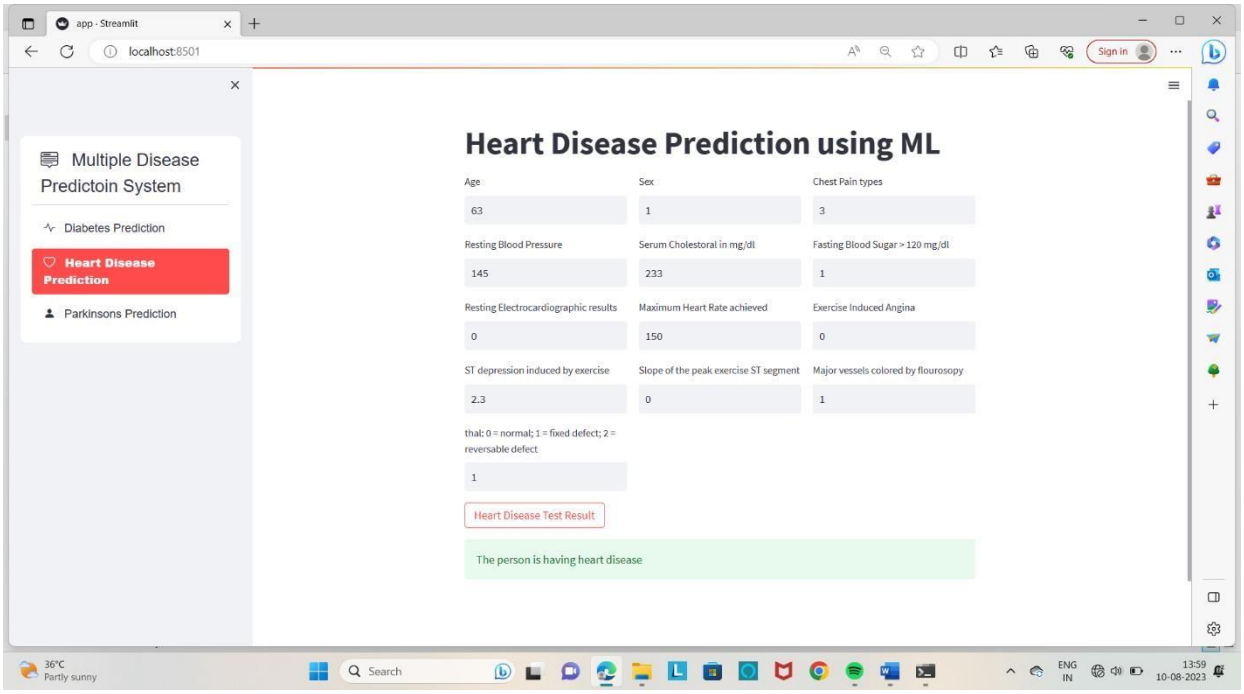


FIG:14

- From the above figure before predict the heart disease the patient must be register with his details like age, sex, chest pain types, resting BP…….
- Entire the patient details it helps to predict the heart disease
- The person is having heart disease

5.2.3 Result Analysis

The result analysis describes that the entire project was executed successfully and also having quality and performance by analyzing the flow of data and output screens. In my project the modules like Service Provider, Remote User, Pre-Proceesing, train model evaluate model and Prediction modules are independent modules. Because my project follows the Top down approach and bottom up approach.

5.3 Method of Implementation

Random Forest

- Steps involved in random forest algorithm:
- Step 1: In Random Forest n number of random records are taken from the data set having k number of records.
- Step 2: Individual decision trees are constructed for each sample.
- Step 3: Each decision tree will generate an output.
- Step 4: Final output is considered based on Majority Voting or Averaging for Classification and regression respectively.

Navie Bayes

- Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.
- It is mainly used in text classification that includes a high-dimensional training dataset.
- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

SOFTWARE AND TECHNOLOGY DESCRIPTION

Introduction to Python

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language

constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbagecollected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library. Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system capable of collecting reference cycles.

Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3.

The Python 2 language, i.e. Python 2.7.x, was officially discontinued on 1 January 2020 (first planned for 2015) after which security patches and other improvements will not be released for it.[32][33] With Python 2's end-of-life, only Python 3.5.x and later are supported. Python interpreters are available for many operating systems. A global community of programmers develops and maintains C Python, an open source[35] reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and C Python development.

SYNTAX AND SEMANTICS

Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation.

Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are optional. It has fewer syntactic exceptions and special cases than C or Pascal.

Indentation

Main article: Python syntax and semantics § Indentation

Python uses whitespace indentation, rather than curly brackets or keywords, to delimit blocks. An increase in indentation comes after certain statements; a decrease in

indentation signifies the end of the current block. Thus, the program's visual structure accurately represents the program's semantic structure. This feature is sometimes termed the off-side rule, which some other languages share, but in most languages indentation doesn't have any semantic meaning.

Statements and control flow

Python's statements include (among others):

The assignment statement (token '=', the equals sign). This operates differently than in traditional imperative programming languages, and this fundamental mechanism (including the nature of Python's version of variables) illuminates many other features of the language. Assignment in C, e.g., x = 2, translates to "typed variable name x receives a copy of numeric value 2". The (right-hand) value is copied into an allocated storage location for which the (left-hand) variable name is the symbolic address.

The memory allocated to the variable is large enough (potentially quite large) for the declared type. In the simplest case of Python assignment, using the same example, x = 2, translates to "(generic) name x receives a reference to a separate, dynamically allocated object of numeric (int) type of value 2." This is termed binding the name to the object. Since the name's storage location doesn't contain the indicated value, it is improper to call it a variable.

Names may be subsequently rebound at any time to objects of greatly varying types, including strings, procedures, complex objects with data and methods, etc.
Successive assignments of a common value to multiple names, e.g., x = 2; y = 2; z = 2 result in allocating storage to (at most) three names and one numeric object, to which all three names are bound.

Since a name is a generic reference holder it is unreasonable to associate a fixed data type with it. However at a given time a name will be bound to some object, which will have a type; thus there is dynamic typing.

- The if statement, which conditionally executes a block of code, along with else and else if (a contraction of else-if).

- The for statement, which iterates over an iterable object, capturing each element to a local variable for use by the attached block.
- The while statement, which executes a block of code as long as its condition is true.
- The try statement, which allows exceptions raised in its attached code block to be caught and handled by except clauses; it also ensures that clean-up code in a finally block will always be run regardless of how the block exits.
- The raise statement, used to raise a specified exception or re-raise a caught exception.
- The class statement, which executes a block of code and attaches its local namespace to a class, for use in object-oriented programming.
- The def statement, which defines a function or method.
- The with statement, from Python 2.5 released in September 2006, which encloses a code block within a context manager (for example, acquiring a lock before the block of code is run and releasing the lock afterwards, or opening a file and then closing it), allowing Resource Acquisition Is Initialization (RAII)-like behavior and replaces a common try/finally idiom.
- The break statement, exits from the loop.
- The continue statement, skips this iteration and continues with the next item.
- The pass statement, which serves as a NOP. It is syntactically needed to create an empty code block.
- The assert statement, used during debugging to check for conditions that ought to apply.
- The yield statement, which returns a value from a generator function. From Python 2.5, yield is also an operator. This form is used to implement coroutines.

The import statement, which is used to import modules whose functions or variables can be used in the current program. There are three ways of using import: import <module name> [as <alias>] or from <module name> import * or from <module name> import <definition 1> [as <alias 1>], <definition 2> [as <alias 2>],

The print statement was changed to the print() function in Python 3.

Python does not support tail call optimization or first-class continuations, and, according to Guido van Rossum, it never will. However, better support for coroutine-like functionality is provided in 2.5, by extending Python's generators.

Before 2.5, generators were lazy iterators; information was passed unidirectionally out of the generator. From Python 2.5, it is possible to pass information back into a generator function, and from Python 3.3, the information can be passed through multiple stack levels.

EXPRESSIONS

Some Python expressions are similar to languages such as C and Java, while some are not:

Addition, subtraction, and multiplication are the same, but the behavior of division differs. There are two types of divisions in Python. They are floor division (or integer division) // and floating point/division. Python also added the ** operator for exponentiation.

From Python 3.5, the new @ infix operator was introduced. It is intended to be used by libraries such as NumPy for matrix multiplication.

From Python 3.8, the syntax :=, called the 'walrus operator' was introduced. It assigns values to variables as part of a larger expression.

In Python, == compares by value, versus Java, which compares numerics by value and objects by reference. (Value comparisons in Java on objects can be performed with the equals() method.) Python's is operator may be used to compare object identities (comparison by reference). In Python, comparisons may be chained, for example a <= b <= c.

Python uses the words and, or, not for its boolean operators rather than the symbolic &&, ||,!used in Java and C.

Python has a type of expression termed a list comprehension. Python 2.4 extended list comprehensions into a more general expression termed a generator expression.

Anonymous functions are implemented using lambda expressions; however, these are limited in that the body can only be one expression.

Conditional expressions in Python are written as x if c else y (different in order of operands from the c ? x : y operator common to many other languages).

Python makes a distinction between lists and tuples. Lists are written as [1, 2, 3], are mutable, and cannot be used as the keys of dictionaries (dictionary keys must be immutable in Python). Tuples are written as (1, 2, 3), are immutable and thus can be used as the keys of dictionaries, provided all elements of the tuple are immutable. The + operator can be used to concatenate two tuples, which does not directly modify their contents, but rather produces a new tuple containing the elements of both provided tuples. Thus, given the variable t initially equal to (1, 2, 3), executing t = t + (4, 5) first evaluates t + (4, 5), which yields (1, 2, 3, 4, 5), which is then assigned back to t, thereby effectively "modifying the contents" of t, while conforming to the immutable nature of tuple objects. Parentheses are optional for tuples in unambiguous contexts.

Python features sequence unpacking wherein multiple expressions, each evaluating to anything that can be assigned to (a variable, a writable property, etc.), are associated in the identical manner to that forming tuple literals and, as a whole, are put on the left hand side of the equal sign in an assignment statement.

The statement expects an iterable object on the right hand side of the equal sign that produces the same number of values as the provided writable expressions when iterated through, and will iterate through it, assigning each of the produced values to the corresponding expression on the left.

Python has a "string format" operator %. This functions analogous to printf format strings in C, e.g. "spam=%s eggs=%d" % ("blah", 2) evaluates to "spam=blah eggs=2".

In Python 3 and 2.6+, this was supplemented by the format() method of the str class, e.g. "spam={0} eggs={1}".format("blah", 2). Python 3.6 added "f-strings": blah = "blah"; eggs = 2; f'spam={blah} eggs={eggs}'.

Python has various kinds of string literals:

Strings delimited by single or double quote marks. Unlike in Unix shells, Perl and Perl-influenced languages, single quote marks and double quote marks function identically. Both kinds of string use the backslash (\) as an escape character. String interpolation became available in Python 3.6 as "formatted string literals".

Triple-quoted strings, which begin and end with a series of three single or double quote marks. They may span multiple lines and function like here documents in shells, Perl and Ruby.

Raw string varieties, denoted by prefixing the string literal with an r. Escape sequences are not interpreted; hence raw strings are useful where literal backslashes are common, such as regular expressions and Windows-style paths. Compare "@-quoting" in C#.

Python has array index and array slicing expressions on lists, denoted as a[key], a[start:stop] or a[start:stop:step]. Indexes are zero-based, and negative indexes are relative to the end. Slices take elements from the start index up to, but not including, the stop index. The third slice parameter, called step or stride, allows elements to be skipped and reversed. Slice indexes may be omitted, for example a[:] returns a copy of the entire list. Each element of a slice is a shallow copy.

In Python, a distinction between expressions and statements is rigidly enforced, in contrast to languages such as Common Lisp, Scheme, or Ruby. This leads to duplicating some functionality. For example:

List comprehensions vs. for-loops

Conditional expressions vs. if blocks

The eval() vs. exec() built-in functions (in Python 2, exec is a statement); the former is for expressions, the latter is for statements.

Statements cannot be a part of an expression, so list and other comprehensions or lambda expressions, all being expressions, cannot contain statements. A particular case of this is that an assignment statement such as a = 1 cannot form part of the conditional

expression of a conditional statement. This has the advantage of avoiding a classic C error of mistaking an assignment operator = for an equality operator == in conditions: if (c = 1) { ... } is syntactically valid (but probably unintended) C code but if c = 1: ... causes a syntax error in Python.

METHODS

Methods on objects are functions attached to the object's class; the syntax instance.method(argument) is, for normal methods and functions, syntactic sugar for Class.method(instance, argument). Python methods have an explicit self parameter to access instance data, in contrast to the implicit self (or this) in some other object-oriented programming languages (e.g., C++, Java, Objective-C, or Ruby).

APPLICATIONS OF PYTHON

As mentioned before, Python is one of the most widely used language over the web. I'm going to list few of them here:

Easy-to-learn − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

Easy-to-read − Python code is more clearly defined and visible to the eyes.
Easy-to-maintain − Python's source code is fairly easy-to-maintain.

A broad standard library − Python's bulk of the library is very portable and crossplatform compatible on UNIX, Windows, and Macintosh.

Interactive Mode − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

Portable − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

Extendable − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

Databases − Python provides interfaces to all major commercial databases.

GUI Programming − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

Scalable − Python provides a better structure and support for large programs than shell scripting.

INSTALLATION STEPS OF PYTHON

Installing and using Python on Windows 10 is very simple. The installation procedure involves just three steps:

- Download the binaries
- Run the Executable installer
- Add Python to PATH environmental variables

To install Python, you need to download the official Python executable installer. Next, you need to run this installer and complete the installation steps. Finally, you can configure the PATH variable to use python from the command line.

Step 1: Download the Python Installer binaries

- Open the official Python website in your web browser. Navigate to the Downloads tab for Windows.
- Choose the latest Python 3 release. In our example, we choose the latest Python 3.7.3 version. Click on the link to download Windows x86 executable installer if you are using a 32-bit installer.
- In case your Windows installation is a 64-bit system, then download Windows x86-64 executable installer.

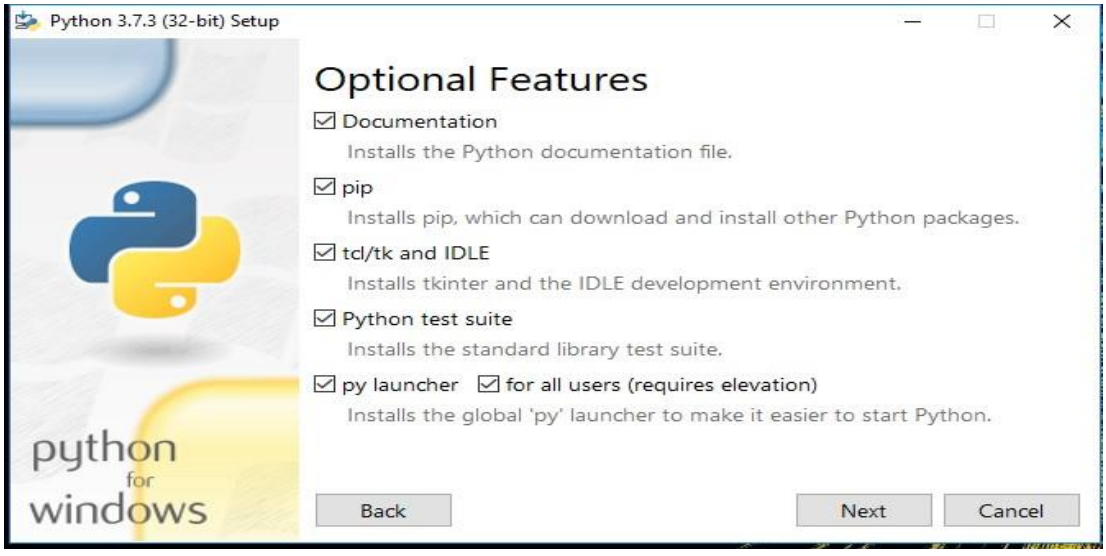
Step 2: Run the Executable Installer

1. Once the installer is downloaded, run the Python installer.
2. Check the Install launcher for all users check box. Further, you may check the Add Python 3.7 to path check box to include the interpreter in the execution path.

3. Select **Customize installation**.

Choose the optional features by checking the following check boxes:

1. Documentation
2. pip
3. tcl/tk and IDLE (to install tkinter and IDLE)
4. Python test suite (to install the standard library test suite of Python)
5. Install the global launcher for `.py` files. This makes it easier to start Python 6. Install for all users.

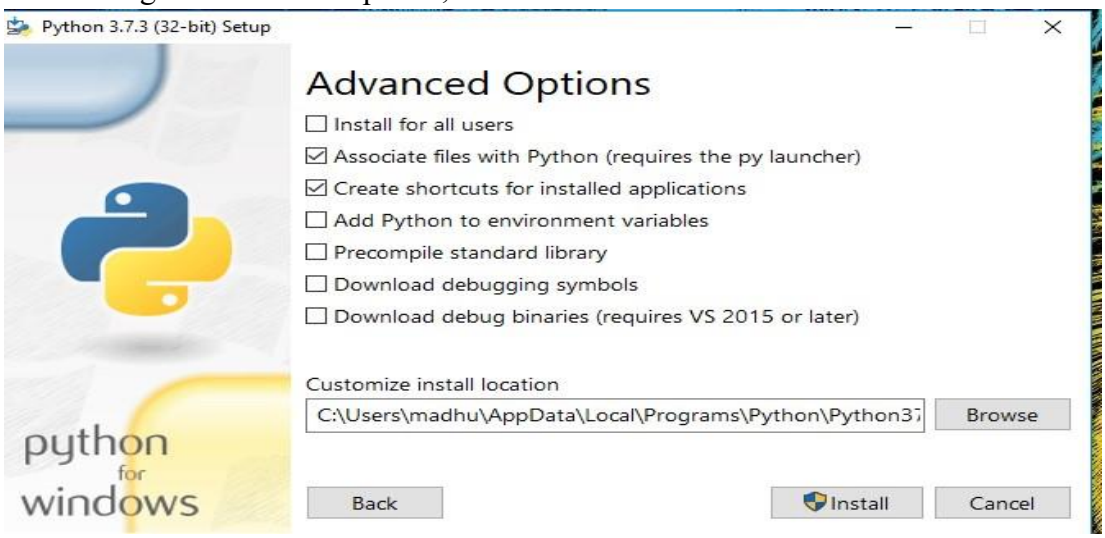


Click Next.

4. This takes you to Advanced Options available while installing Python. Here, select the Install for all users and Add Python to environment variables check boxes.

Optionally, you can select the Associate files with Python, Create shortcuts for installed applications and other advanced options. Make note of the python installation directory displayed in this step. You would need it for the next step.

After selecting the Advanced options, click Install to start installation.



5. Once the installation is over, you will see a Python Setup Successful window.

Step 3: Add Python to environmental variables

The last (optional) step in the installation process is to add Python Path to the System Environment variables. This step is done to access Python through the command line. In case you have added Python to environment variables while setting the Advanced options during the installation procedure, you can avoid this step. Else, this step is done manually as follows.

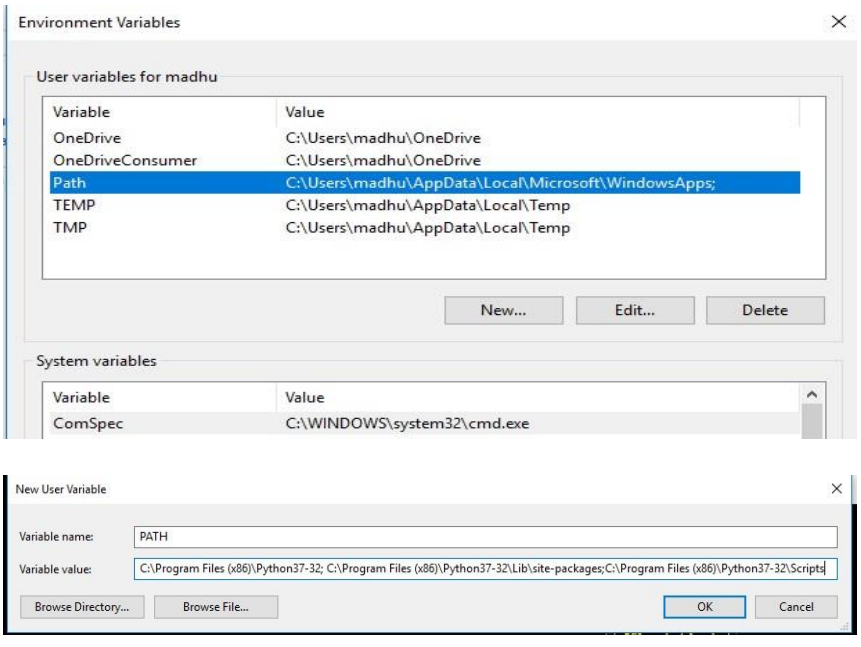
In the Start menu, search for “advanced system settings”. Select “View advanced system settings”. In the “System Properties” window, click on the “Advanced” tab and then click on the “Environment Variables” button.

Locate the Python installation directory on your system. If you followed the steps exactly as above, python will be installed in below locations:

- C:\Program Files (x86)\Python37-32: for 32-bit installation
- C:\Program Files\Python37-32: for 64-bit installation

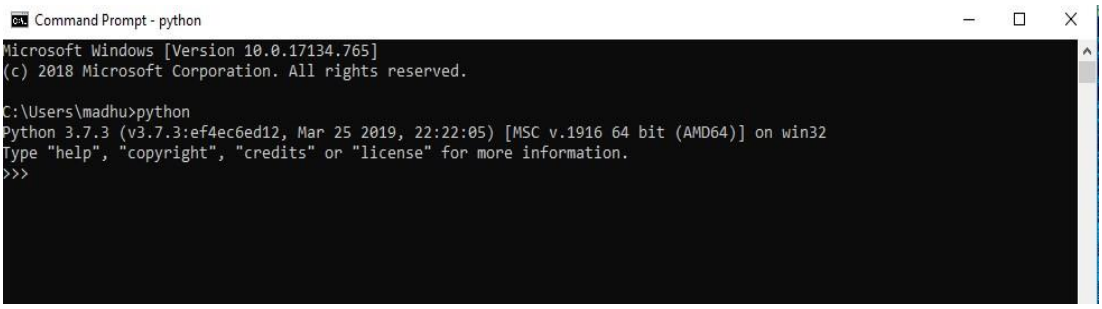
The folder name may be different from “Python37-32” if you installed a different version. Look for a folder whose name starts with Python.

Append the following entries to PATH variable as shown below:

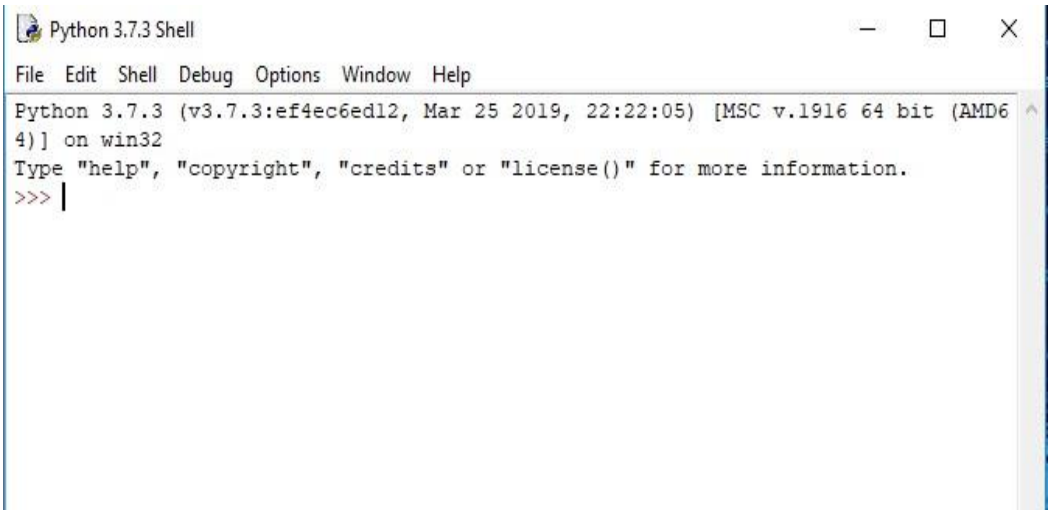


Step 4: Verify the Python Installation

You have now successfully installed Python 3.7.3 on Windows 10. You can verify if the Python installation is successful either through the command line or through the IDLE app that gets installed along with the installation. Search for the command prompt and type “python”. You can see that Python 3.7.3 is successfully installed.



An alternate way to reach python is to search for “Python” in the start menu and clicking on IDLE (Python 3.7 64-bit). You can start coding in Python using the Integrated Development Environment(IDLE).



Uses

Since 2003, Python has consistently ranked in the top ten most popular programming languages in the TIOBE Programming Community Index where, as of February 2020, it is the third most popular language (behind Java, and C). It was selected Programming Language of the Year in 2007, 2010, and 2018.

- An empirical study found that scripting languages, such as Python, are more productive than conventional languages, such as C and Java, for programming

problems involving string manipulation and search in a dictionary, and determined that memory consumption was often "better than Java and not much worse than C or C++".

- Large organizations that use Python include Wikipedia, Google, Yahoo!, CERN, NASA, Facebook, Amazon, Instagram, Spotify and some smaller entities like ILM and ITA. The social news networking site Reddit is written entirely in Python.
- Python can serve as a scripting language for web applications, e.g., via mod_wsgi for the Apache web server. With Web Server Gateway Interface, a standard API has evolved to facilitate these applications. Web frameworks like Django, Pylons, Pyramid, Turbo Gears, web2py, Tornado, Flask, Bottle and Zope support developers in the design and maintenance of complex applications. Pyjs and Iron Python can be used to develop the client-side of Ajax-based applications.
- SQL Alchemy can be used as data mapper to a relational database. Twisted is a framework to program communications between computers, and is used (for example) by Dropbox.
- Libraries such as NumPy, SciPy and Matplotlib allow the effective use of Python in scientific computing, with specialized libraries such as Biopython and Astropy providing domain-specific functionality. Sage Math is a mathematical software with a notebook interface programmable in Python: its library covers many aspects of mathematics, including algebra, combinatorics, numerical mathematics, number theory, and calculus.
- Python has been successfully embedded in many software products as a scripting language, including in finite element method software such as Abaqus, 3D parametric modeler like Free CAD, 3D animation packages such as 3ds Max, Blender, Cinema 4D, Lightwave, Houdini, Maya, modo, Motion Builder,

  Softimage, the visual effects compositor Nuke, 2D imaging programs like GIMP, Inkscape, Scribus and Paint Shop Pro, and musical notation programs like scorewriter and capella.
- GNU Debugger uses Python as a pretty printer to show complex structures such as C++ containers. Esri promotes Python as the best choice for writing scripts in

ArcGIS. It has also been used in several video games, and has been adopted as first of the three available programming languages in Google App Engine, the other two being Java and Go.

- Python is commonly used in artificial intelligence projects with the help of libraries like TensorFlow, Keras, Pytorch and Scikit-learn. As a scripting language with modular architecture, simple syntax and rich text processing tools, Python is often used for natural language processing.
- Many operating systems include Python as a standard component. It ships with most Linux distributions, AmigaOS 4, FreeBSD (as a package), NetBSD, OpenBSD (as a package) and macOS and can be used from the command line (terminal). Many Linux distributions use installers written in Python: Ubuntu uses the Ubiquity installer, while Red Hat Linux and Fedora use the Anaconda installer. Gentoo Linux uses Python in its package management system, Portage.
- Python is used extensively in the information security industry, including in exploit development.
- Most of the Sugar software for the One Laptop per Child XO, now developed at Sugar Labs, is written in Python. The Raspberry Pi single-board computer project has adopted Python as its main user-programming language.
- Due to Python's user-friendly conventions and easy-to-understand language, it is commonly used as an intro language into computing sciences with students. This allows students to easily learn computing theories and concepts and then apply them to other programming languages.
- LibreOffice includes Python, and intends to replace Java with Python. Its Python Scripting Provider is a core feature since Version 4.0 from 7 February 2013.

5.4 CONCLUSION

By observing the above methods of implementation, output screens, result analysis the implementation phase was completed successfully with quality, good performance and also satisfying all user requirements. It satisfies all java application development standards.

6. SYSTEM TESTING

6.1 INTRODUCTION

Testing is that the debugging program is one amongst the leading crucial aspects of the pc programming triggers, while not programming that works, the system would ne'er turn out relate in Nursing output of that it had been designed .Testing is best performed once user development is asked to help in characteristic all errors and bugs. The sample knowledge are used for testing. It is not amount however quality of the information used the matters of testing .Testing is aimed toward guaranteeing that the system was accurately relate in Nursing with efficiency before live operation commands.

TESTING OBJECTIVES

The most objective of testing is to uncover a bunch of errors, consistently and with minimum effort and time. Stating formally, Testing may be a method of corporal punishment a program with intent of finding miscalculation.

- ✓ A productive check is one that uncovers Associate in nursing hitherto undiscovered error.
- ✓ A decent legal action is one that has likelihood of finding miscalculation, if it exists.
- ✓ The check is insufficient to find probably gift errors.
- ✓ The code additional or less confirms to the standard and reliable standards.

6.1.1 TYPES OF TESTING

6.1.1.1 Unit Testing

Unit testing they have a tendency to test every module separately and integrate with the general system. Unit testing focuses verification efforts on the littlest unit of code style within the module. This is often conjointly called module testing. The module of the system is tested individually. As an example the validation check is completed for variable the user input given by the user that validity of the information entered. It's terribly straightforward to search out error rectify the system.

Every Module will be tested victimization the subsequent 2 Strategies: recording machine Testing and White Box Testing.

6.1.1.2 Black Box Testing

Recording machine checking may be a code testing techniques during which practicality of the code below test (SUT) is tested while not staring at the interior code structure, implementation details and data of internal ways of the code. This type of testing is predicated entirely on the code needs and specifications .In recording machine Testing they have a tendency to simply concentrate on inputs and output of the package while not bothering concerning internal data of the code program.

The on top of recording machine will be any package you wish to check. For example, Associate in Nursing software like Windows, a web site like Google ,a information like Oracle or maybe the own custom application. Under recording machine testing, you can check these applications by simply that specialize in the inputs and outputs while not knowing their internal code implementation.

Types of Black Box Testing

There are many varieties of recording machine testing however following the outstanding ones.

- ✓ **Functional testing:** This recording machine testing kind is said to purposeful needs of a system; it's done by code testers.
- ✓ **Non-Functional testing:** This sort of recording machine testing isn't associated with testing of a selected practicality, however non-functional needs like performance, measurability, usability.
- ✓ **Regression testing:** Regression testing is completed once code fixes, upgrades or the other system maintenance to visualize the new code has not affected the prevailing code.

6.1.1.3 White Box Testing

White Box Testing is that the testing of a code solution's internal committal to writing and infrastructure. It focuses totally on Traffic Redundancy Elimination security, the flow of inputs and outputs through the applying, and rising style and value. White box testing is additionally called clear, open, structural, and glass box testing. It is one amongst 2 elements of the "box testing" approach of code testing.

6.1.1.4. System Testing

Once the individual module testing is completed, modules are assembled and integrated to perform as a system. The top down testing that began from higher level to lower level module was allotted to visualize whether or not the whole system is playacting satisfactorily. There are 3 main types of System testing: Alpha Testing, Beta Testing, and Acceptance Testing.

- ✓ **Alpha Testing:** This refers to the system checking that allotted by the test team with the Organization.
- ✓ **Beta Testing:** This refers to the system testing that performed by a particular cluster of friendly customers.
- ✓ **Acceptance Testing:** This refers to the system testing that performed by the client to see whether or not or to not settle for the delivery of the system.

6.2 TEST STRATEGY AND APPROACH

Field Testing will be performed manually and functional tests will be written in detail.

- ➢ **Test objectives**
  - ✓ All field entries must work properly.
  - ✓ Pages must be activated from the identified link.
  - ✓ The entry screen, messages and responses must not be delayed.
- ➢ **Features to be tested**
  - ✓ Verify that the entries are of the correct format.
  - ✓ No duplicate entries should be allowed.
  - ✓ All links should take the user to the correct page.

6.2.1 TEST CASES

TC No	Test Case	Input	Expected Output	Observed Output	Result
TC1	Login	Enter mobile no and password	Login Successful	--do--	Pass
TC2	Login	Enter Wrong mobile no and password	Invalid login details	--do--	Pass
TC3	Registration	Enter all Fields Data	Registration Successful	--do--	Pass
TC4	Registration	Enter some fields data	All fields are mandatory	--do--	Pass

Table 6.1: Test Cases Table

6.3 VALIDATION

Testing process starts with a test plan. This plan identifies all the testing related activities that must be performed and specifies the schedules, allocates the resources, and specified guidelines for testing. During the testing of the unit the specified test cases are executed and the actual result compared with expected output. The final output of the testing phase is the test report and the error report.

Test Data:

Here all test cases that are used for the system testing are specified. The goal is to test the different functional requirements specified in Software Requirements Specifications (SRS) document.

Unit Testing:

Each individual module has been tested against the requirement with some test data.

Test Report:

The module is working properly provided the user has to enter information. All data entry forms have tested with specified test cases and all data entry forms are working properly.

Error Report:

If the user does not enter data in specified order then the user will be prompted with error messages. Error handling was done to handle the expected and unexpected errors.

6.4 CONCLUSION

By observing all sample test cases which were taken in my project was successfully done. The testing strategy like system testing, integration testing, unit tastings are used in my project to test the established system. Finally, I conclude that my project has been working very accurately with quality and good performance.

7. CONCLUSION

7.1 CONCLUSION

We can conclude that the chatbot is very easy to use for all people; they can use this chatbot in their language. This bot offers medical-related information like doctor's contact details, address of nearby hospitals, contact details for getting an oxygen cylinder, about the disease its symptoms, its prevalence, diagnosis, and its treatment procedures. We consider that our findings will help researchers to take advantage of the advanced information in the layout and other stuff of these innovative technologies, which may be required for continuous development in the functionality of medical chatbots and may help prevent COVID 19. This medical chatbot has wide future opportunities. People in remote areas can also receive benefits from this. Here we use `TensorFlow', which helps to build the NLP for chatbots and utilizes deep neural network architecture. After building the network for our chatbot, it will predict the correct answers to the user's queries. Even if it is not in the training model, the bot tries to predict it closer by checking the sentences and their words which will be closer to the training model's response.

8. BIBILOGRAPHY

References

[1] J. Cahn, ``CHATBOT: Architecture, design, & development," Dept. Comput. Inf. Sci., Univ. Pennsylvania School Eng. Appl. Sci., Philadelphia, PA, USA, Tech. Rep. EAS499, 2017.

[2] E. Tebenkov and I. Prokhorov, ``Machine learning algorithms for teaching AI chat bots," Proc. Comput. Sci., vol. 190, pp. 735744, Jan. 2021.

[3] A. S. Lokman and M. A. Ameedeen, ``Modern chatbot systems: A technical review," in Proc. Future Technol. Conf. Cham, Switzerland: Springer, Nov. 2018, pp. 10121023.

[4] A. Kumar, P. K. Meena, D. Panda, and M. Sangeetha, ``Chatbot in Python," Int. Res. J. Eng. Technol., vol. 6, no. 11, 2019.

[5] S. Raj and K. Raj, Building Chatbots With Python. New York, NY, USA: Apress, 2019.

[6] K. H. Koundinya, A. K. Palakurthi, V. Putnala, and K. A. Kumar, ``Smart college chatbot using ML and Python," in Proc. Int. Conf. Syst., Comput., Automat. Netw. (ICSCAN), Jul. 2020, pp. 15.

[7] S. A. Sheikh, ``Articial intelligence based chatbot for human resource using deep learning," Ph.D. dissertation, Dept. Comput. Sci. Eng., Manipal Univ., Manipal, India, 2019.

[8] M. M. Hossain, S. Krishna Pillai, S. E. Dansy, and A. A. Bilong, ``Mr. Dr. Healthassistant chatbot," Int. J. Artif. Intell., vol. 8, no. 2, pp. 5873, Dec. 2021.

[9] R. Dharwadkar and N. A. Deshpande, ``A medical chatbot," Int. J. Comput. Trends Technol., vol. 60, no. 1, pp. 4145, 2018.

[10] D. Madhu, C. J. N. Jain, E. Sebastain, S. Shaji, and A. Ajayakumar, ``A novel approach for medical assistance using trained chatbot," in Proc. Int. Conf. Inventive Commun. Comput. Technol. (ICICCT), Mar. 2017, pp. 243246.

[11] F. Mehfooz, S. Jha, S. Singh, S. Saini, and N. Sharma, ``Medical chatbot for novel COVID-19," in ICT Analysis and Applications. Singapore: Springer, 2021, pp. 423430.

[12] M. Herriman, E. Meer, R. Rosin, V. Lee, V. Washington, and K. G. Volpp, ``Asked and answered: Building a chatbot to address COVID-19-related concerns," NEJM Catalyst Innov. Care Del., vol. 1, pp. 113, Jun. 2020.

[13] S. Altay, A. S. Hacquin, C. Chevallier, and H. Mercier, ``Information delivered by a chatbot has a positive impact onCOVID-19 vaccines attitudes and intentions," J. Exp. Psychol., Appl., vol. 27, pp. 111, Oct. 2021.

[14] P. Amiri and E. Karahanna, ``Chatbot use cases in the COVID-19 public health response," J. Amer. Med. Inform. Assoc., vol. 29, no. 5, pp. 10001010, Apr. 2022.

[15] M. Almalki and F. Azeez, ``Health chatbots for fighting COVID-19: A scoping review," Acta Inf. Medica, vol. 28, no. 4, p. 241, 2020.

[16] P. Weber and T. Ludwig, ``(Non-) interacting with conversational agents: Perceptions and motivations of using chatbots and voice assistants," in Proc. Conf. Mensch Comput., vol. 1, Sep. 2020, pp. 321331.

[17] U. Bharti, D. Bajaj, H. Batra, S. Lalit, S. Lalit, and A. Gangwani, ``Medbot: Conversational articial intelligence powered chatbot for delivering tele-health after COVID-19," in Proc. 5th Int. Conf. Commun. Electron. Syst. (ICCES), Jun. 2020, pp. 870875.

[18] G. Battineni, N. Chintalapudi, and F. Amenta, ``AI chatbot design during an epidemic like the novel coronavirus," Healthcare, vol. 8, no. 2, pp. 18, Jun. 2020.

[19] B. A. Shawar and E. Atwell, ``Using dialogue corpora to train a chatbot," in Proc. Corpus Linguistics Conf., Mar. 2003, pp. 681690.

[20] M. Ciotti, M. Ciccozzi, A. Terrinoni, W. C. Jiang, C. B. Wang, and S. Bernardini, ``The COVID-19 pandemic," Crit. Rev. Clin. Lab. Sci., vol. 57, no. 6, pp. 365388, 2020.

[21] S. J. Daniel, ``Education and the COVID-19 pandemic," Prospects, vol. 49, no. 1, pp. 9196, 2020.

[22] N. Rosruen and T. Samanchuen, ``Chatbot utilization for medical consultant system," in Proc. 3rd Technol. Innov. Manag. Eng. Sci. Int. Conf. (TIMES-iCON), Dec. 2018, pp. 15.

[23] K. Rarhi, A. Bhattacharya, A. Mishra, and K. Mandal, ``Automated medical chatbot," Tech. Rep., 2017.

[24] S. Majumder and A. Mondal, ``Are chatbots really useful for human resource management?" Int. J. Speech Technol., vol. 24, no. 4, pp. 969977, Dec. 2021.

[25] A. S. Ashour, A. El-Attar, N. Dey, H. A. El-Kader, and M. M. A. El-Naby, ``Long short term memory based patient-dependent model for FOG detec-tion in Parkinson's disease," Pattern Recognit. Lett., vol. 131, pp. 2329, Mar. 2020, doi: 10.1016/j.patrec.2019.11.036.